

TEMARIO FINAL 2023

Computación



102 Años
del carisma de
San Jerónimo Emiliani
en América (1921-2023)
Misión, Memoria y Esperanza





Centenario Somascos en América

EMILIANI

1921-2021

**Devoción
Caridad
Trabajo**

TEMARIO TÉCNICO 2023

MECÁNICA • ELECTRÓNICA • ELECTRICIDAD

DIBUJO DE ARQUITECTURA • COMPUTACIÓN



Campus Central Mixco
Km 14.5, Calzada San Juan, Guatemala



www.emilianisomascos.edu.gt



Autoridades

Instituto Emiliani Somascos

Comunidad Somasca
Obras Somascas en Guatemala

Lic. Raúl Hernández Chacón
Director Técnico-Administrativo Instituto Emiliani Somascos

Lic. Henry Caal
Sub-director Instituto Emiliani Somascos

Lic. Juan Carlos Morales
Coordinador Académico

Prof. David Subuyuj
Coordinador Técnico

Armando Garcia
Coordinación de Pastoral

Bachillerato Industrial y Perito con especialidad en Computación

Silvio Ernesto Castro
Asesor de Práctica Supervisada

Christian Morales
Asesor de Práctica Supervisada

Promoción 2023

6to Computación

Nombres

Sección "A"

- Edwin Eduardo Alvarez Arriaga
- Diego Angel de Jesús Alvizures Orantes
- David Nehemias Bautista Patzán
- Brayan Ismael Chóm Pacajoj
- Cristian Alexander Coromac López
- Willian Estuardo Cuevas Rivas
- Nohelia Guadalupe CulajayOxcal
- Jenifer Celeste Cuyan Gomez
- Angel Daniel De León Alvarado
- Kevin Alexander De Leon Andrade
- Marta Alicia del Carmen Diaz Cifuentes
- Oscar Alexander Echeverría López
- Aroldo David Escobar López
- Thania Paola García Cos
- David Daniel Gerónimo Vaquiax
- Patrick Leonardo Gómez Paz
- Joshua Alessandro González Gonzáles
- Brandon David Grijalva Marroquín
- Braulio José Alexander Guarcax Lima
- Eddy Junior Guerrero Gomez
- Ricardo Alberto Hernández Bautista
- José Santiago Herrera Furlán
- Cristian Antonio Im Monterroso
- José Carlos Jordán Cardona
- Angel Enrique Juarez Asencio
- Gustavo Rafael Juárez Mendoza
- María Emilia Larios Contreras
- Glendy Sucely Lopez Cruz
- Didier Joshua Lopez Juarez
- Byron Aaron Mansilla Fuentes
- German Daniel Marquez Estrada
- Elder César Oswaldo Martínez Rustrian

Sección "B"

- Adolfo Isaác Morales Delgado
- Erick Estuardo Morán Pech
- María José Noj Sor
- José Antonio Osoy Crespo
- Diego Manuel Fernando Pacheco Racanac
- Pavel Alexander Palma Carrillo
- Erwin Gabriel Patzán Roca
- Wilmer Estuardo Patzán Vail
- Leonel De Jesús Erasmo Pérez Lúcas
- Krystel Adriany Popa Patzán
- Rodrigo Fernando Ramírez Guzmán
- José Gabriel Ramos Pérez
- Fernando José Raxón Pirir
- Hilda Aileen Raymundo Diaz
- Gabriel Eduardo Rivas Morales
- Ariel Ernesto Rodríguez Morales
- Iker Rodolfo Rojas Reyes
- David José Gabriel Romero Alvarado
- Ronaldo Hezler Adriel Ruano Valle
- Gabriela Alexandra Santos Carrera
- Dilan Elizandro Sarceño Jacinto
- Dulce María Shuyá Baten
- Maryori Omeydi Sipac Sesam
- Edgar Laureano Soc Rac
- Juan Alejandro Tiney García
- Jeremy Jhosef Tobar Lopez
- Jonathán Javier Tubac Arreaza
- Diego Gabriel Vásquez Rosales
- Eduardo Steve Velásquez Gómez
- Ricardo Mauricio Ventura Martínez
- Axel Daniel Yoc Garcia
- Mario Ricardo Yoc Hernández

Centenario Somasco en América

Cien años de la comunidad somasca en América, aquel lejano 5 de octubre de 1921 en el cual tuvimos el arribo en el puerto de la libertad, en el vecino país del Salvador, el que significa el primer apostolado somasco. Posteriormente en 1927 comenzaron las obras en Guatemala. Pensar que tenemos ya en este año 2023 más de 100 años de nosotros también ser partícipes de una obra que sigue hasta el día de hoy creciendo y expandiéndose, un mensaje de perseverancia que nos muestra que no todo llega rápidamente, pero que en algún momento todo da sus frutos.

Solamente Dios sabe qué es lo que hace y cómo lo hace, cuál es el rumbo de las cosas, de su corazón nacen las cosas que son buenas, la ayuda, la sabiduría, la inteligencia, solo vienen de él, sus planes para todos en los que en él confían, serán siempre de bendición. Él es el que da las herramientas para que podamos salir adelante, siempre interfiere en todo lo que nos pasa en el día a día, en cada hora, en cada minuto, cada instante de nuestras vidas está planeado desde mucho antes de que nosotros conociéramos el primer instante de vida.

Eso significa la comunidad, una comunidad que significó una luz, una muestra de que el amor divino, que la mano de Dios aún está con nosotros y que seguirá estando, de la mano de todos aquellos colaboradores y de todos aquellos corazones que están conectados con la espiritualidad, que son verdaderos devotos a la obra verdaderamente cristiana, que mucha falta hace en estos tiempos tan difíciles.

Una comunidad que ha hecho muchas obras de bien para la sociedad, fundando instituciones educativas, seminarios, con el fin siempre de acercar más a las personas a Jesús, que es la verdadera visión y vocación que siempre se ha querido plasmar y transmitir hacia cada persona que está involucrada en esta obra masiva de devoción, caridad y trabajo. Lo que representó San Jerónimo a partir de su conversión, y que demostró toda su vida a partir de aquella experiencia propia que tuvo con la Virgen María. Su virtud es llevar el mensaje de salvación a partir de la ayuda hacia aquellos que más lo necesitan.

La pandemia que vivimos en estos últimos dos años, sin duda significó muchas cosas perdidas, familias que perdieron trabajo, que están pasando angustias económicas, incluso familias en donde muchos miembros ya no están, suena como un mensaje pesimista, pero es la verdadera realidad en la cual nos encontramos viviendo, pero cuando hacemos algo, por ayudar a alguien más estamos contribuyendo al propósito de nuestras vidas.

Cuando empezamos a ver más allá de lo que nuestra perspectiva común ve, cuando dejamos de centrarnos en lo que pasamos, o a aún con problemas, luchas externas e internas, podemos ver aquellas realidades en las que nosotros, podemos intervenir, y quizá no cambiemos la realidad de la persona, hay situaciones realmente difíciles, pero, sí podemos dar una palabra incluso que venga del corazón, y dar ánimo nuevamente a una persona.

Porque la obra de caridad es algo más allá que ser llamado propiamente un santo, porque la caridad está en cada uno de aquellos corazones dispuestos a sacrificar algo, por servir, por ayudar a los demás. Porque la caridad requiere de intencionalidad también, el deseo y el sentido del deber, de dar sin esperar recibir nada a cambio, realmente eso es lo que completa una verdadera conversión, porque si recibimos la salvación, y la

misericordia de Cristo, cómo no tener misericordia de aquellos que alguna vez fuimos nosotros.

El Centenario Somasco en América, significa más que solo 100 años, significan años de trabajo, de sacrificio, de cambios, de una transformación que incluso todos buscamos a diario. Quizá lo más significativo más allá de toda la historia que engloba estos cien años de la Comunidad Somasca en América, tenemos un proyecto de vida que busca reunir a todas aquellas personas marginadas, necesitadas, dirigirse a los niños, como el mismo Jesús lo haría en su tiempo. Resulta más que grata su presencia en América y propiamente en Guatemala, y deseamos que, así como tenemos un centenario se sigan cumpliendo más y más años de nuevos retos, pero sobre todo de practicar el amor al prójimo, el mensaje de salvación por medio de las obras.



Introducción

6to Computación

A lo largo de este trabajo como alumnos del 6to grado en bachiller industrial y perito con especialidad en computación, daremos a conocer cada uno de los temas vistos e impartidos por nuestros docentes en el ciclo escolar 2021 que corresponde al grado de Quinto grado de nuestra especialidad ya antes mencionada.

Explicaremos y daremos a conocer en cada tema que en este documento se presenta, cada uno de los aspectos, características, aplicaciones y utilidades de dichos temas, tratando de dejar claro cada uno de ellos.

Abarcando de una manera clara y directa todo lo visto en ese ciclo escolar, principalmente orientado a la programación en C++, programación orientada a objetos en C#, conexiones a bases de datos, Visual Basic, entre muchos otros temas que serán de apoyo y guía para retroalimentar y fomentar en los estudiantes el correcto desarrollo de aplicaciones y proyectos que necesiten de estos tipos de software, lenguajes y conocimientos.

Justificación

6to Computación

El motivo por el cual todos los estudiantes realizamos la recopilación de toda la información que se presenta en este documento, es principalmente para que cada uno de nosotros tenga acceso a un medio de estudio y retroalimentación de todos los temas comprendidos, no solamente este año, sino también para un ámbito laboral y profesional que como futuros graduandos no estamos lejos a alcanzar, Cada uno podrá estudiar y repasar para mejorar y poner en práctica cada una de nuestras habilidades adquiridas a lo largo de todo este ciclo escolar. Ya que será información que no solo nos servirá este año, sino que, si nos esmeramos y ponemos empeño en cada una de las prácticas y competencias en cada tema, vamos a adquirir los conocimientos necesarios que nos servirán en un grado de educación superior universitaria y a lo largo de toda nuestra vida laboral.

Objetivos

6to Computación

En el siguiente documento tenemos una recopilación de todos los temas vistos en el primer año de nuestra carrera, todos estos temas son tanto prácticos como técnicos, este documento nos apoya a la retroalimentación sobre todo lo que hemos aprendido en el transcurso de los años.

El principal objetivo de este documento es poder estudiar todos los temas que contiene e introducirnos nuevamente a los temas, y así no dejarlos a un lado ni olvidarlos, ya que tanto estos como los que vemos este último año de carrera son de suma importancia para nuestra realización como profesionales.

Nuestro temario está dividido en clases y capítulos, cada clase tiene cuatro capítulos representando los bimestres transcurridos. En cada uno de los capítulos se pueden identificar los temas principales que vimos todo ese año. Leer el temario es una buena práctica para repasar los conocimientos adquiridos con anterioridad.

Los objetivos específicos de nuestro temario son:

- Retroalimentarnos respecto a los temas pasados.
- Explicar de forma fácil de retener, clara y concisa cada tema tocado en este documento.
- Brindar ejercicios prácticos que ayudan a la retención de información.

El temario tiene como fin el estudio del mismo para los graduandos que cursen el presente año 2023.

Temas

Sexto grado Programación

Estructurada

Capítulo I

1. PHP5
2. PHP Introducción
3. PHP Sintaxis
4. Variables PHP
5. PHP Echo i Imprimir
6. PHP Tipos de datos
7. Funciones PHP de Cuerda
8. PHP Constantes
9. PHP Operadores
10. PHP If ... Else ... Elseif
11. PHP conmutador
12. PHP do... \Vhile
13. PHP For
14. PHP Funciones
15. PHP Anays
16. PHP Ordenando Arrays
17. PHP Superglobales
18. PHP fecha y hora
19. PHP Incluir
20. PHP Administración de
21. archivos
22. PHP Open File / Read
23. PHP File Crear / escritura
24. PHP File Upload
25. PHP cookies
26. PHP Sesiones
27. PHP trlanejo de errores
28. excepción PHP

Capítulo II

1. Java Script I
2. JS Introduction
3. JS Where To
4. JS Output
5. JS Syntax
6. JS Statementsv
7. JS Comments
8. JS Variables
9. JS Data Types
10. JS Objects
11. JS Functions
12. JS Scope

13. JS Events
14. JS Strings
15. JS String Methods
16. JS Numbers
17. JS Number Methods
18. JS Operators
19. JS Math
20. JS Dates
21. JS Date Methods
22. JS Arrays
23. JS Array Methods
24. JS Booleans
25. JS RegExp
26. JS Type Conversion
27. JS Comparisons

Capítulo III

1. jQuery Hide/Show
2. jQuery Fade
3. jQuery Slide
4. jQuery Animate
5. jQuery stop()
6. jQuery Callback
7. jQuery Chaining
8. jQuery Get
9. jQuery Set
10. jQuery Add
11. jQuery Remove
12. jQuery CSS
13. Classes
14. jQuery css()
15. jQuery
16. Dimensions
17. Ajax
18. Filter Tables
19. Exposición Técnica
20. ° Briefing
21. ° Planning
22. ° Frontend
23. ° UI de software
24. ° Backend de programación,
25. lanzamiento
26. ° mantenimiento de software

Redes de la Información

Capítulo I

I. Introducción y clasificación de las redes

- 1.1 . Que es una red
- 1.2, Comunicación de datos
- 1.3, Canal de comunicación
- 1.4. Red de computadores
- 1.5. Telemática
- 1.6. Ventajas de las redes
- 1.7. Modelo de sistema de comunicación
- 1.8. Fuente
- 1.9. Transmisor
- 1.10. Clasificación de las redes según su tamaño y extensión:
 - 1.10.1. LAN
 - 1.10.2. PAN
 - 1.10.3. rLAN
 - 1.10.4. WAN
 - 1.10.5. WLAN
 - 1.10.6. VLAN
 - 1.10.7. CANI
 - 1.10.8. SAN
 - 1.10.9. Redes internet
 - 1.10.10. Redes inalámbricas
 - 1.10.11. Topologías

Capítulo II

2. Modelo Osi y protocolos TCP/IP

- 2.1 Capas del modelo OSI
- 2.2 Capa 1: La capa física
- 2.3 Capa 2: La capa de enlace de datos.
- 2.4 Capa 3: La capa de red.
- 2.5 Capa 4: La capa de transporte.
- 2.6 Capa 5: La capa de sesión.
- 2.7 Capa 6: La capa de presentación
- 2.8 Capa 7: La capa de aplicación
- 2.9 Funciones de cada capa
- 2.10 Protocolos utilizados en cada capa
- 2.11 Dispositivos de redes utilizados en cada capa
- 2.12 Redes IP
- 2.13 El protocolo IP versión 4
- 2.14 El protocolo ICMP
- 2.15 El protocolo UDP
- 2.16 El protocolo TCP

Capítulo III

1. Subneteo Clase A
 - No. De subredes
 - Dirección de red
 - Tipos de clase
 - Mascaras de subred
 - Redes Útiles Dirección
 - IP Asignación de subred
2. Subneteo Clase B
 - No. De subredes
 - Dirección de red
 - Tipos de clase
 - Mascaras de subred
 - Redes Útiles Dirección
 - IP Asignación de subred
3. Subneteo Clase C
 - No. De subredes
 - Dirección de red
 - Tipos de clase
 - Mascaras de subred
 - Redes Útiles Dirección
 - IP Asignación de subred
4. Exposición técnica:
 - Conexiones de red
 - Comunicación de cliente/servidor
 - Conexiones de red con gestor de base de datos.

Tecnología Vocacional III

Capítulo I

1. W3C

- 1.1 HTML & CSS
- 1.2 Gráficos
- 1.3 Audio y Video
- 1.4 internacionalización
- 1.5 Redes Móviles
- 1.6 Privacidad
- 1.7 Historia y evolución
- 1.8 Versiones
- 1.9 Ejemplos de navegadores Web

Capítulo II

2. Integración CMS+CRM+ERP

- 2.1 Conceptos
- 2.2 Historia
- 2.3 Implementación
- 2.4 ERP versus software de gestión
- 2.5 Ventajas y desventajas
- 2.6 Seguridad en la Web
- 2.7 Historia
- 2.8 Vulnerabilidad
- 2.9 Soluciones
- 2.10 Tipos de ataques

Capítulo III

3. Arquitectura y seguridad de las aplicaciones WEB

- 3.1 Qué es un software Web
- 3.2 Usos de un software Web
- 3.3 Por qué elegir una solución web
- 3.4 Elección de un software web

- 3.5 Tipos de Software web
- 3.6 Software Web Serial
- 3.7 Donde encontrarlos
- 3.8 Tipos de licencia
- 3.9 Ventajas y desventajas de un software serial
- 3.10 Software con servicios rentados
- 3.11 Software de software Web adapta
- 3.12 Como se implementó
- 3.13 Software Web desarrollado a medida
- 3.14 Ciclos de vida del software
- 3.15 Quienes lo hacen
- 3.16 Tiempos de desarrollo
- 3.17 Obsolescencia tecnológica o de desarrollo
- 3.18 Ejemplos de implementación

Estructura de datos

Capítulo I

- 1. MySQL (Gráfico consola)
- 2. Entorno de trabajo
- 3. Instalación de mysql
- 4. Crear base de datos
- 5. Llave primaria
- 6. Llave foránea
- 7. Mostrar base de datos y tablas
- 8. Tipos de datos básicos
- 9. Valores null
- 10. DML
 - 10.1 Create
 - 10.1 Alter
 - 10.1 Drop
 - 10.1 Create
 - 10.1 Truncate table
- 11. DDL
 - 11.1 Select
 - 11.2 Insert
 - 11.3 Update
 - 11.4 Delete

Capítulo II

- 2. Framework web y scripting
 - 2.1. ¿Qué es un framework web?
 - 2.2 Patrón MVC y Model 2.
 - 2.3 Tipos de framework Web
 - 2.4 Características
 - 2.5 Ejemplo
 - 2.6 Historia
 - 2.7 Aplicaciones
 - 2.8 Tecnologías

Capítulo III

- 3. Laravel
 - 3.1 Entorno de trabajo
 - 3.2 Comandos
 - 3.3 Node JS
 - 3.4 Composer
 - 3.5 Nuevas páginas
- 4. controladores
- 5. listas
- 6. Enlaces de páginas
- 7. Listas dinámicas
- 8. Diseño
- 9. Funciones
- 10. Aler

1.1 PHP

Por: Edwin Alvarez

1.1.1 PHP5

Fue una de las versiones de PHP que uso los manejadores de objetos.



Imagen: Logo de PHP 5

Fuente: <https://i.stack.imgur.com/BUtyg.png>

1.1.2 PHP Introducción

PHP es un lenguaje de programación interpretado usado generalmente para la creación de contenido dinámico en sitios web.



Imagen: Logo de PHP

Fuente: <https://upload.wikimedia.org/wikipedia/commons/2/27/PHP-logo.svg>

1.1.3 PHP Sintaxis

Etiqueta de apertura y cierre `<?php ?>`, comentarios `/* */`, variables `$`, asignación de valor `=`, sentencias y bloqueos de código `;`, operadores aritméticos `+ - * / %`, funciones función.

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Unidad 2 - Ejemplo 3</title>
5     <meta http-equiv="Content-Type"
6         content="text/html; charset=ISO-8859-1" />
7   </head>
8   <body>
9     <h3>Nombres: </h3>
10    <?php
11      echo "Luis Pablo <br />";
12    ?>
13    <h3>Apellidos:</h3>
14    <?php
15      echo "Perez Jimenez <br />";
16    ?>
17    <h3>Edad:</h3>
18    <?php
19      echo "26 años<br/>";
20    ?>
21    <h3>Teléfono:</h3>
22    <?php
23      echo "+57 (1) 323 12 00";
24    ?>
25  </body>
26 </html>

```

Imagen: Ejemplos de sintaxis

Fuente: https://cdn.goconqr.com/uploads/flash_card/image_question/7277208/desktop_ecdb4574-aebb-4e90-b814-1ba0c83b81e3.jpg

1.1.4 Variables PHP

Se declaran con el signo de \$, las variables pueden tener varios tipos de datos, los valores que se les asigna es con el signo =, luego están las variables superglobales que son \$_GET, \$_POST, \$_SESSION, \$_COOKIE, estas se encargan de acceder a datos enviados por formularios.

```

<?php

$websiteName = "Pi My Life Up";

$number = 5;

$decimalNumber = 5.10;

$myArray = ["Example", "Test"];

echo $websiteName . '<br>';
echo $number . '<br>';
echo $decimalNumber . '<br>';
var_dump($myArray);

```

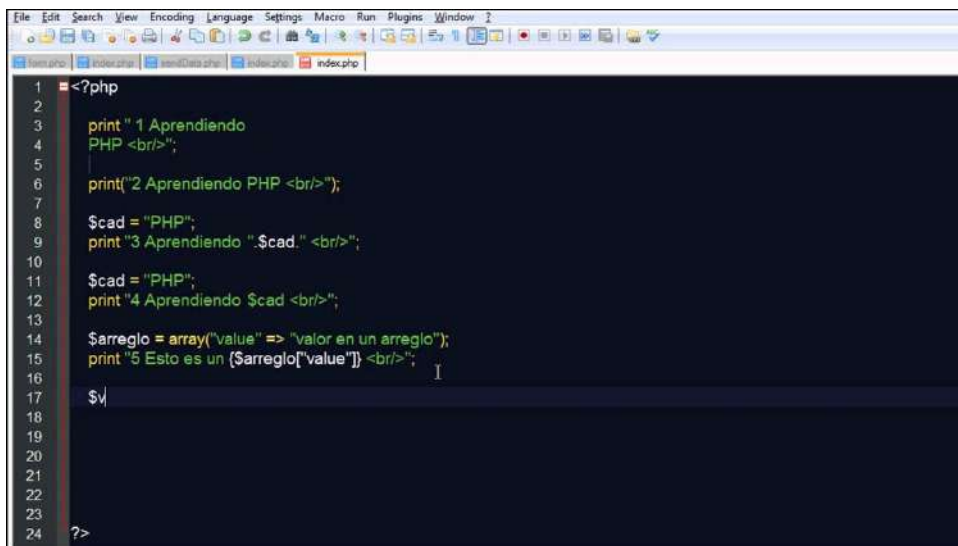


Imagen: Ejemplos de variables

Fuente: <https://cdn.pimylifeup.com/wp-content/uploads/2022/07/PHP-variables.jpg>

1.1.5 PHP Echo i Imprimir

Se utilizan para mostrar contenido en salida, ambas tienen la misma función que es imprimir texto, echo: puede usarse sin o con paréntesis, puede imprimir uno o varios elementos. Print: imprime un elemento a la vez, devuelve un valor entero y siempre usa paréntesis.



```
1 <?php
2
3 print " 1 Aprendiendo
4 PHP <br>";
5
6 print("2 Aprendiendo PHP <br>");
7
8 $cad = "PHP";
9 print "3 Aprendiendo ".$cad." <br>";
10
11 $cad = "PHP";
12 print "4 Aprendiendo $cad <br>";
13
14 $arreglo = array("value" => "valor en un arreglo");
15 print "5 Esto es un {$arreglo["value"]} <br>";
16
17 $v
18
19
20
21
22
23
24 ?>
```

Imagen: Ejemplificación del uso de Print

Fuente: https://i.ytimg.com/vi/ODQ18UBB_sl/maxresdefault.jpg

1.6 PHP Tipos de datos

Por: Diego Alvizures

El tipo de un dato se caracteriza por dos piezas de información: el valor en sí mismo y las operaciones con las que podemos manipularlo. Los tipos de datos que hay son:

- Integer: Los enteros, en esencia, son todos números ENTEROS positivos o negativos.
- Float: El formato de dato del tipo "coma flotante" o "float" se aplica a los números con decimales. Los números de coma flotante tienen una mayor resolución que los de 32 bits
- String: El tipo de datos de cadena es un conjunto de caracteres expresados entre comillas. "Esto es una cadena" o incluso "esto es UNA cadena" e incluso "1223429 12489 4058934" son tres ejemplos simples de cadenas, conjuntos de caracteres.
- Boolean: Es un tipo de dato que solo puede expresar dos valores: verdadero o falso.
- Null: El valor especial nulo se refiere a una variable sin valor. Nulo en PHP se considera un tipo de datos.
- Array y objetos: Un array en PHP es en realidad un mapa ordenado. Un mapa es un tipo de datos que asocia valores con claves.

```

<head>
<title> Trabajando con variables </title>
</head>
<body>
<?php
//Declaramos variables
$string = "Pepe";
$integer = 7;
$double = 6.7;
$boolean = true;
$array = array("Juana","Pepe");
echo "Hola $string <br/>";
echo "El numero $integer me trae suerte <br/>";
echo "$string sacó un $double en el examen <br/>";
echo "El resultado de la operación es $boolean <br/>";
echo "Hay dos alumnos $array[0] y $array[1] <br/>";
?>
</body>
</html>

```

Imagen: Ejemplo de tipos de datos de PHP

Fuente: <https://aulab.es/articulos-guias-avanzadas/30/los-tipos-de-datos-en-php>

1.7 PHP de Cuerda

Se utilizan para almacenar y procesar texto, después de crear una cadena, podemos manipularla.

PHP Source:	Result:
<pre><!DOCTYPE html> <html> <body> <?php echo strlen("Hello world!"); ?> </body> </html></pre>	12

Imagen: Ejemplo de una función de php de cuerda
 Fuente: https://www.w3bai.com/es/php/php_string.html#gsc.tab=0

1.7.1 Contar el número de palabras en una cadena

El PHP str_word_count() función cuenta el número de palabras de una cadena:

PHP Source:	Result:
<pre><!DOCTYPE html> <html> <body> <?php echo str_word_count("Hello world!"); ?> </body> </html></pre>	2

Imagen: Ejemplo de contar el número de palabras
 Fuente: https://www.w3bai.com/es/php/php_string.html#gsc.tab=0

1.7.2 Invertir una cadena

El PHP strrev() función invierte una cadena:

PHP Source:	Result:
<pre><!DOCTYPE html> <html> <body> <?php echo strrev("Hello world!"); ?> </body> </html></pre>	!dlrow olleH

Imagen: Ejemplo de invertir una palabra
 Fuente: https://www.w3bai.com/es/php/php_string.html#gsc.tab=0

1.7.3 Para buscar un texto específico dentro de una cadena

Los PHP strpos() función busca un texto específico dentro de una cadena.

Si se encuentra una coincidencia, la función devuelve la posición del primer partido. Si no se encuentra ninguna coincidencia, devolverá FALSO.

PHP Source:	Result:
<pre><!DOCTYPE html> <html> <body> <?php echo strpos("Hello world!", "world"); ?> </body> </html></pre>	6

Imagen: Ejemplo de buscar una palabra específica

Fuente: https://www.w3bai.com/es/php/php_string.html#gsc.tab=0

1.8 PHP Constantes

Una constante es un identificador (nombre) para un valor simple. Como el nombre sugiere, este valor no puede variar durante la ejecución del script (a excepción de las constantes mágicas, que en realidad no son constantes). Por defecto, una constante distingue mayúsculas y minúsculas. Por convención, los identificadores de constantes siempre se declaran en mayúsculas.

```
<?php
define("CONSTANTE", "Hola mundo.");
echo CONSTANTE; // muestra "Hola mundo."
echo Constante; // muestra "Constant" y se emite un aviso.
?>
```

Imagen: Ejemplo de sintaxis para definir una constante

Fuente: <https://www.php.net/manual/es/language.constants.syntax.php>

1.9 PHP Operadores

Los operadores son los símbolos que nos permiten expresar las operaciones entre los datos en PHP al crear una tienda o página web. Entre los tipos de operadores que hay están:

- Aritméticos: Son los más sencillos y proporcionan que podamos realizar operaciones numéricas con las variables.
- De asignación: Son los que traspasan valores en variables.
- De comparación: Las utilizaremos para comparar dos variables y demostrar si cumple o no la propiedad del operador.
- Lógicos: Son los que nos devuelven un sí o un no y pueden utilizarse combinados con los operadores de comparación cuando la expresión de la condición lo pide.

```
$c = true;
$d = false;

if($c && $d) {
    echo '$c y $d los dos son verdaderos';
} else {
    echo 'o uno de $a o $b son falsos, o los dos son falsos';
}
```

Imagen: Ejemplo de un operador lógico en PHP

Fuente: <https://www.arsys.es/blog/operadores-php#Operadores en PHP por tipos de operaciones>

1.10 PHP If ... Else ... Elseif

Es una combinación de if y else. Como else, extiende una if declaración para ejecutar una declaración diferente en caso de que la if expresión original se evalúe como false. Sin embargo, a diferencia de else, ejecutará esa expresión alternativa solo si la elseif expresión condicional se evalúa como true.

```
<?php
if ($a > $b) {
    echo "a is bigger than b";
} elseif ($a == $b) {
    echo "a is equal to b";
} else {
    echo "a is smaller than b";
}
?>
```

Imagen: Ejemplo de la sintaxis de un If ... Else ... Elseif en PHP
Fuente: <https://www.php.net/manual/en/control-structures.elseif.php>

1.11 PHP conmutador

Un conmutador o switch es similar a una serie de sentencias IF en la misma expresión. En muchas ocasiones, es posible que se quiera comparar la misma variable (o expresión) con muchos valores diferentes, y ejecutar una parte de código distinta dependiendo de a qué valor es igual.

```
<?php
if ($i == 0) {
    echo "i es igual a 0";
} elseif ($i == 1) {
    echo "i es igual a 1";
} elseif ($i == 2) {
    echo "i es igual a 2";
}

switch ($i) {
    case 0:
        echo "i es igual a 0";
        break;
    case 1:
        echo "i es igual a 1";
        break;
    case 2:
        echo "i es igual a 2";
        break;
}
?>
```

Imagen: Ejemplo de un conmutador en PHP

Fuente: <https://www.php.net/manual/en/control-structures.elseif.php>

1.1 PHP do\while

Por: David Bautista

En PHP, el bucle do-while es una estructura de control que permite ejecutar un bloque de código al menos una vez y luego repetirlo mientras se cumpla una condición. La sintaxis básica del bucle do-while en PHP es la siguiente:

```
do {
    // código a ejecutar
} while (condición);
```

Aquí está cómo funciona el bucle do-while:

1. El bloque de código dentro del do se ejecuta primero, sin importar la condición.
2. Después de que se ejecuta el bloque de código, la condición se evalúa.
3. Si la condición es verdadera, el bucle se repite y el bloque de código se ejecuta nuevamente.
4. Si la condición es falsa, el bucle se detiene y la ejecución continúa con la siguiente instrucción después del bucle do-while.

```

1  <?php
2
3  $numeros = array(5, 7, 11, 15, 20, 25);
4  $buscar = 15;
5  $posicion = -1;
6  $i = 0;
7
8  while ($i < count($numeros) && $posicion == -1) {
9      if ($numeros[$i] == $buscar) {
10         $posicion = $i;
11     }
12     $i++;
13 }
14
15 if ($posicion == -1) {
16     echo "El valor no se encuentra en el arreglo.\n";
17 } else {
18     echo "El valor se encuentra en la posición $posicion del arreglo.\n";
19 }
20

```

Imagen: Aquí hay un ejemplo para ilustrar como se usa do while en PHP

Fuente: <https://andresledo.es/wp-content/uploads/2023/01/php-while.png>

1.1.1 PHP For

El bucle **for** en PHP es una estructura de control que permite repetir un bloque de código un número específico de veces. Tiene la siguiente sintaxis:

```
for (inicialización; condición; incremento) {  
    // código a ejecutar en cada iteración  
}
```

La inicialización se utiliza para establecer un valor inicial antes de que comience el bucle. Por lo general, se utiliza para inicializar una variable de control que se utilizará en la condición del bucle.

La condición es evaluada antes de cada iteración. Si la condición se evalúa como verdadera, se ejecuta el bloque de código dentro del bucle. Si la condición se evalúa como falsa, el bucle se detiene y la ejecución continúa con el código después del bucle for.

El incremento se utiliza para actualizar la variable de control después de cada iteración. Por lo general, se incrementa o decrementa el valor de la variable en una cantidad fija.

```
<html>  
<head>  
<title> Trabajando con la sentencia for </title>  
</head>  
<body>  
  
<?php  
//Declaramos e incializamos variables  
$i = 1;  
  
for ($i = 1; ; $i++) {  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
}  
  
?>  
</body>  
</html>
```

TU
INSTITUTO
ONLINE

Imagen: Aquí hay un ejemplo que imprime los números del 1 al 10

Fuente: https://www.tuinstitutoonline.com/cursos/paginas_web_dinamicas/img/web127.png

1.1.2 PHP Funciones

- 1. Declaración de funciones:** Para declarar una función en PHP, se utiliza la palabra clave "function" seguida del nombre de la función y los paréntesis que pueden contener los argumentos de la función. Por ejemplo:

```
function nombreFuncion($arg1, $arg2) {
    // Cuerpo de la función
}
```

- 2. Argumentos:** Las funciones en PHP pueden aceptar argumentos que se pasan entre paréntesis cuando se llama a la función. Los argumentos pueden ser variables, valores o incluso otras funciones. Por ejemplo:

```
function suma($num1, $num2) {
    return $num1 + $num2;
}
```

`$resultado = suma(3, 5); // Llamada a la función con argumentos`



```
function.php
1 <?php
2
3 function holamundo(){
4     return "hola mundo";
5 }
6
7 function sumar($num1,$num2){
8
9     $suma=$num1+$num2;
10
11     return $suma;
12 }
13
14
15 echo holamundo();
16
17 echo "<br>";
18
19 echo sumar();
20
```

Imagen: Ejemplo de como funciona las funciones en PHP

Fuente: <https://i.ytimg.com/vi/kYYJm-2DmaU/maxresdefault.jpg>

1.1.3 PHP Ordenando Arrays

PHP proporciona varias funciones para ordenar arrays de diferentes maneras. A continuación, te presentaré algunas de las funciones más comunes:

1. `sort()`: Esta función ordena un array en orden ascendente. Modifica el array original reordenando sus elementos. Los índices asociativos se pierden y se reemplazan por índices numéricos. Aquí tienes un ejemplo:

```
$fruits = array("apple", "banana", "orange", "mango");
```

```
sort($fruits);
```

```
print_r($fruits);
```

Salida:

```
Array
```

```
(
```

```
  [0] => apple
```

```
  [1] => banana
```

```
  [2] => mango
```

```
  [3] => orange
```

```
)
```



```

1  <?php
2  // https://parzibyte.me/blog
3  function burbuja(&$arreglo)
4  {
5      $longitud = count($arreglo);
6      for
7
8
9
10
11
12
13
14  }
15 }
16
17 $miArreg
18 echo "An

```

```

C:\Users\parzibyte\Desktop\ejercicios-php
A php -f burbuja.php
Antes de ordenar: Array
(
    [0] => Marjio
    [1] => Luis
    [2] => Aloy
    [3] => Zelda
    [4] => Link
    [5] => Chris
    [6] => Claire
    [7] => Leon
)
Después de ordenar: Array
(
    [0] => Zelda
    [1] => Marjio
    [2] => Luis
    [3] => Link
    [4] => Leon
    [5] => Claire
    [6] => Chris
    [7] => Aloy
)

```

Imagen: Un ejemplo de como ordenarlo en metodo burbuja

Fuente: <https://parzibyte.me/blog/wp-content/uploads/2021/05/Arreglo-de-cadena-ordenado-con-PHP-y-metodo-de-burbuja.png>

1.17 PHP Superglobales

Por: Cristian Alexander Coromac López

Superglobals son variables internas que están disponibles siempre en todos los ámbitos.

Algunas variables predefinidas en PHP son "superglobales", lo que significa que están disponibles en todos los ámbitos a lo largo del script. No es necesario emplear global \$variable; para acceder a ellas dentro de las funciones o métodos. Las variables superglobales son:

- \$GLOBALS
- \$_SERVER
- \$_GET
- \$_POST
- \$_FILES
- \$_COOKIE
- \$_SESSION
- \$_REQUEST
- \$_ENV



Fuente:

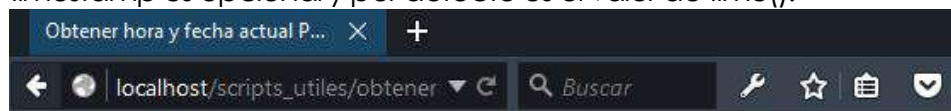
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.phpglossary.com%2Fwhat-are-php-super-global-variables-and-their-usage-in-php%2F&psig=AOvVaw2ksUWXeBIBKqsn21LEqxSe&ust=1685763383611000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCIjC5I7Uo_8CFQAAAAAdAAAAABAI

1.18 PHP Fecha y Hora

Date – Dar formato a la fecha/hora local

`date(string $format, int $timestamp = time()): string`

Devuelve una cadena formateada según el formato dado usando el parámetro de tipo integer timestamp dado o el momento actual si no se da una marca de tiempo. En otras palabras, timestamp es opcional y por defecto es el valor de time().



Fecha y Hora actual

08/24/15 1:33pm



Fuente:

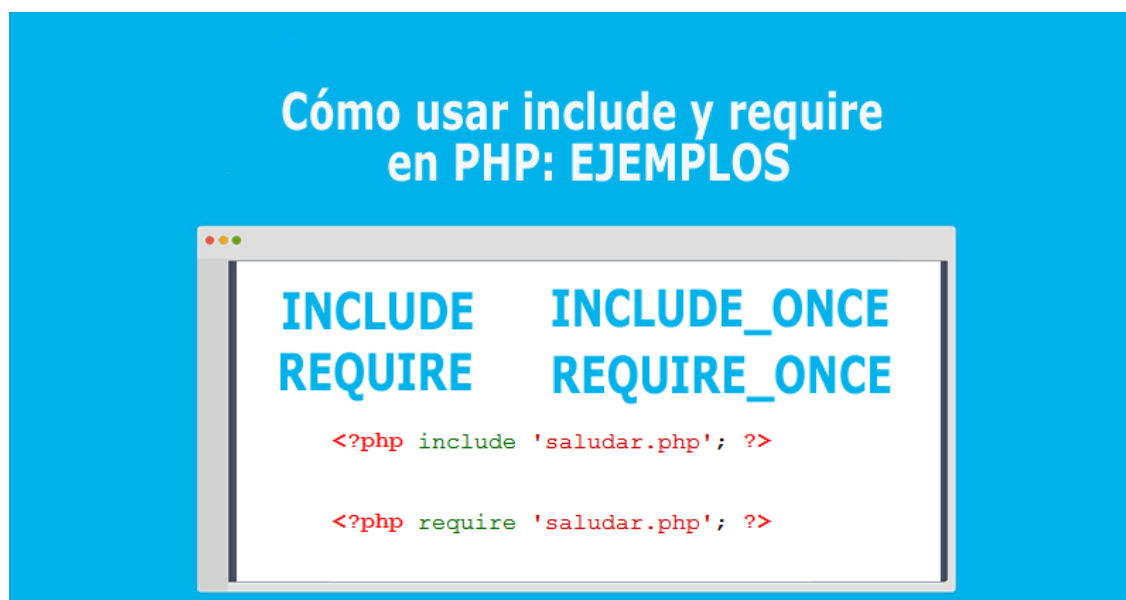
<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.solvetic.com%2Ftutoriales%2Farticle%2F1780-obtener-la-fecha-y-hora-actual-en->

[php%2F&psig=AOvVaw3t9Y3fQaHR8e8GhcMrfQQ2&ust=1685763869645000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCJCbT_bVo_8CFQAAAAAdAAAAABAS](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.baulphp.com%2Fcomo-usar-include-y-require-en-php-ejemplos%2F&psig=AOvVaw3t9Y3fQaHR8e8GhcMrfQQ2&ust=1685763869645000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCJCbT_bVo_8CFQAAAAAdAAAAABAS)

1.19 PHP Incluir

La sentencia include incluye y evalúa el archivo especificado.

Si una ruta es definida ya sea absoluta (comenzando con una letra de unidad o \ en Windows o / en sistemas Unix/Linux) o relativa al directorio actual (comenzando con . o ..) el include_path será ignorado por completo.



Fuente:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.baulphp.com%2Fcomo-usar-include-y-require-en-php-ejemplos%2F&psig=AOvVaw0LruCi-Gsktlm2gIPX8cpx&ust=1685764479807000&source=images&cd=vfe&ved=0CBEQjRxqFwoT_CPD7uJnYo_8CFQAAAAAdAAAAABAE

1.20 PHP Administración de archivos

Los archivos en PHP se abren con la función fopen(), que requiere dos parámetros: el archivo que se quiere abrir y el modo en el que se quiere abrir el archivo. La función devuelve un punteo en el archivo si es satisfactoria o cero si no lo es. Los archivos se abren para realizar operaciones de lectura o escritura.



Fuente:

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fdesarrolloweb.com%2Farticulos%2>

[F394.php&psig=AOvVaw0ymJ--HGvbM4ZODCOb1dRd&ust=1685764739161000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCKDDn5XZo_8CFQAAAAAdAAAAABBe](https://www.google.com/search?q=f394.php&psig=AOvVaw0ymJ--HGvbM4ZODCOb1dRd&ust=1685764739161000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCKDDn5XZo_8CFQAAAAAdAAAAABBe)

1.22 PHP Open File / Read

Un mejor método para abrir archivos es con la `fopen()` función. Esta función le ofrece más opciones que la `readfile()` función.

La `fread()` función lee de un archivo abierto.

El primer parámetro de `fread()` contiene el nombre del archivo desde el que leer y el segundo parámetro especifica el número máximo de bytes para leer.



Fuente:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.codingtag.com%2Fphp-file-openread&psig=AOvVaw2My_4P3ddrGCZsJ7FQY8NV&ust=1685765067385000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCMCs0bHao_8CFQAAAAAdAAAAABAR

1.1 PHP File Crear/Escritura

Por: Willian Estuardo Cuevas Rivas

PHP File Crear, nos ayuda a crear un proyecto de código orientado al lenguaje PHP dentro de nuestro editor de texto Visual Studio Code.

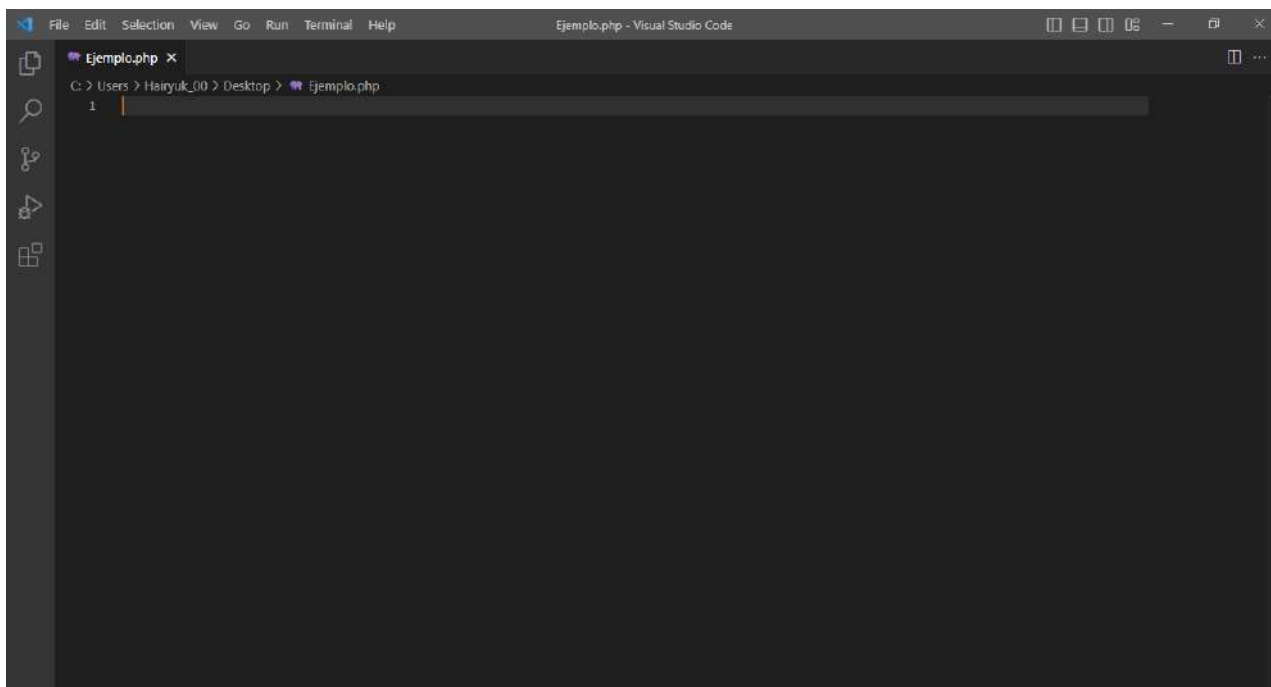


Imagen: Ejemplo de la creación de un proyecto PHP en Visual Code

Fuente: <https://code.visualstudio.com>

1.1.1 PHP File Upload

Por: Willian Estuardo Cuevas Rivas

PHP File Upload nos permite subir o abrir un archive de PHP dentro de nuestro editor de texto Visual Studio Code.

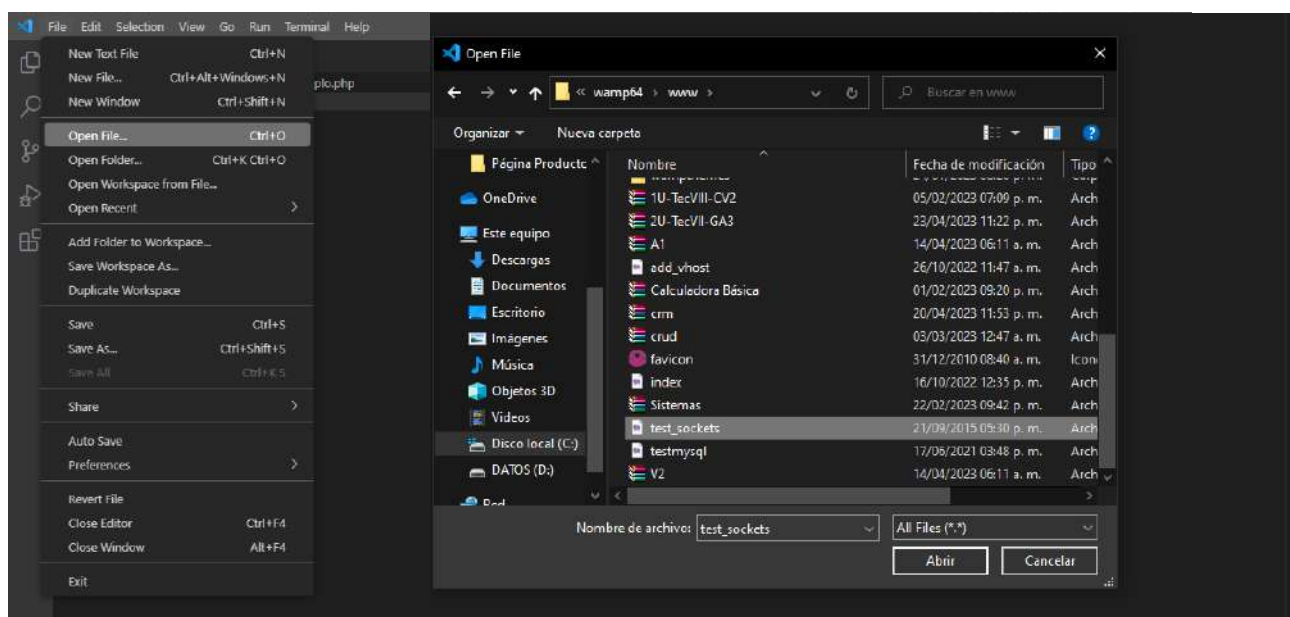


Imagen: Ejemplo de como subir un archivo a Visual Studio Code

Fuente: <https://code.visualstudio.com>

1.2 PHP Cookies

Por: Willian Estuardo Cuevas Rivas

PHP Cookies son un método que proporciona soporte para las Cookies de HTTP. Las Cookies son una forma de almacenar datos en el navegador de usuario para poder monitorear o identificar a los usuarios que regresan a un sitio web.

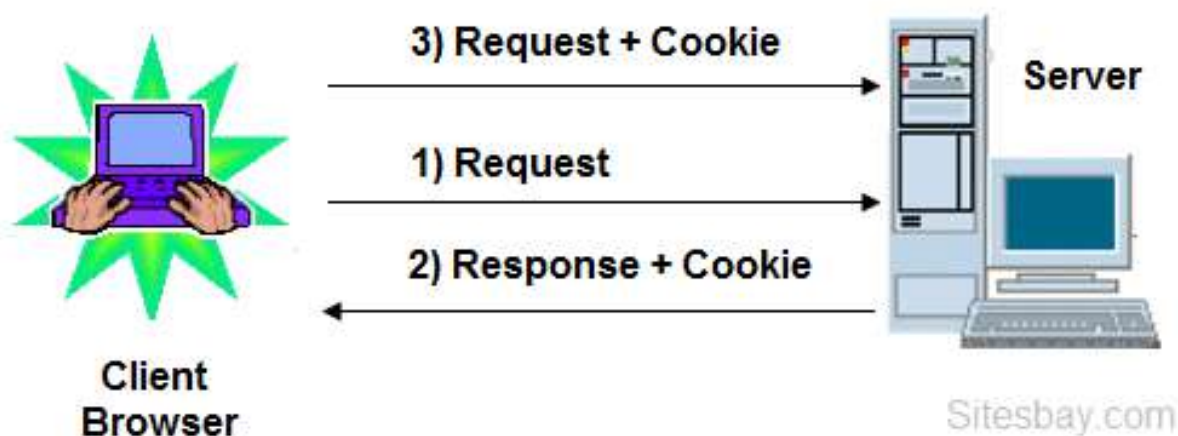


Imagen: Ejemplo de la función de las Cookies dentro de PHP
 Fuente: <https://www.php.net/manual/es/features.cookies.php>

1.2.1 PHP Sesiones

Por: Willian Estuardo Cuevas Rivas

PHP Sesiones son una herramienta que permite almacenar datos de manera individualizada para cada usuario, se utilizan ID de sesiones únicas para identificar y recuperar la información almacenada en sesiones anteriores.

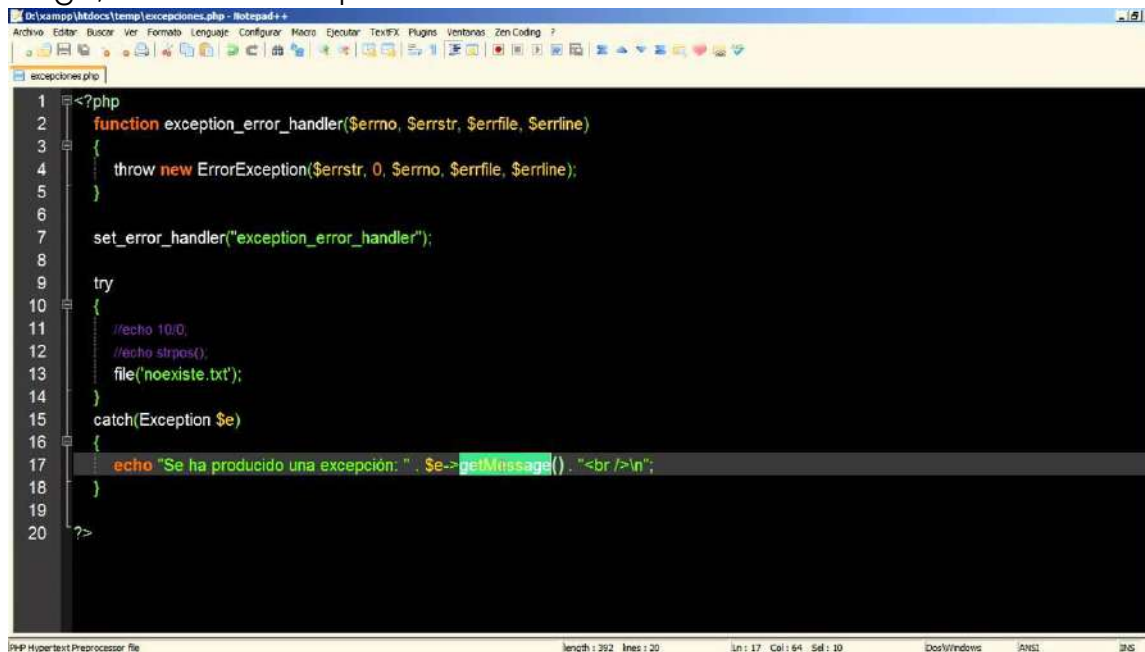


Imagen: Ejemplo de una Sesión dentro de PHP
 Fuente: <https://www.php.net/manual/es/session.examples.basic.php>

1.3 PHP Manejo de errores excepción PHP

Por: Willian Estuardo Cuevas Rivas

PHP tiene módulos de excepción similares a otros lenguajes de programación, estos ayudan principalmente a capturar errores que pueden facilitar el manejo de la ejecución de un código, haciendo así que se eviten los "Error Fatal" en PHP.



```
1 <?php
2 function exception_error_handler($errno, $errstr, $errfile, $errline)
3 {
4     throw new Exception($errstr, 0, $errno, $errfile, $errline);
5 }
6
7 set_error_handler("exception_error_handler");
8
9 try
10 {
11     //echo 10/0;
12     //echo strpos();
13     file('noexiste.txt');
14 }
15 catch(Exception $e)
16 {
17     echo "Se ha producido una excepción: " . $e->getMessage() . "<br />\n";
18 }
19
20 ?>
```

Imagen: Ejemplo de las excepciones dentro de PHP

Fuente: <https://www.php.net/manual/es/language.exceptions.php>

2.1 Java Script

Por: Nohelia Guadalupe Culajay Oxcal

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web.

1.2.2 JS Introduction

JavaScript es un lenguaje de programación compilado ligero, multiplataforma, de un solo subproceso e interpretado, que también se conoce como el lenguaje de secuencias de comandos para páginas web. Es bien conocido por el desarrollo de páginas web, y muchos entornos que no son de navegador también lo utilizan. JavaScript es un tipo de lenguaje tanto imperativo como declarativo. JavaScript contiene una biblioteca estándar de objetos, como Array, Date y Math y un conjunto central de elementos de lenguaje como operadores, estructuras de control y sentencias. JavaScript se puede agregar a su archivo HTML de dos maneras JS interno, JS externo. En HTML, el código JavaScript se inserta entre las etiquetas

```
<script>y </script>.
```

1.2.3 JS Where To

La función `where()` se usa para filtrar la colección por una clave o valor dado contenido dentro de la matriz dada. En JavaScript, la matriz primero se convierte en una colección y luego la función se aplica a la colección de datos. `where('clave')` Parámetros: esta función acepta un solo parámetro como se mencionó anteriormente y se describe a continuación. Clave: este parámetro contiene el nombre de la clave que define el valor de esa clave. Valor de retorno: Devuelve la colección con un valor clave que se mencionó.

1.2.4 JS Output

El comando OUTPUT FORM define el formulario de salida actual de formulario o form Usuario. El formulario debe pertenecer a tabla. El alcance de este comando es el proceso actual. Cada tabla tiene su propio formulario de salida en cada proceso. Los formularios de salida son utilizados por tres grupos de comandos. Un grupo muestra una lista de registros en pantalla, otro grupo genera informes, y el tercer grupo exporta datos. Los comandos DISPLAY SELECTION y MODIFY SELECTION muestran una lista de registros utilizando un formulario de salida. Utilice el formulario de salida durante la creación de informes con los comandos PRINT LABEL y PRINT SELECTION. Cada uno de los comandos de exportación (EXPORT DIF, EXPORT SYLK y EXPORT TEXT) utiliza también el formulario de salida.

```
<> testfile.html •
<> testfile.html > ...
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h1>linuxhint</h1>
6
7  <script>
8  document.write("This is linuxhint.com");
9  </script>
10
11 </body>
12 </html>
13
14
15
```

Imagen: Ejemplo de un Output

Fuente: <https://linuxhint.com/wp-content/uploads/2022/01/word-image-1813.png>

1.2.5 JS Syntax

El Syntax parser en general es quien lee carácter por carácter, formado palabras y traduciendo o entendiendo que orden queremos ejecutar, así como determinar si la sintaxis es correcta o errónea de acuerdo a los estándares que tenga dicho motor, cuando este termina de validar es entonces cuando la instrucción pasa a nuestro motor para procesarla. La sintaxis de JavaScript es el conjunto de reglas que rigen la programación de JavaScript. Este curso le enseñará los conceptos básicos de sintaxis, declaraciones, variables y asignaciones de variables. Descubrirá tipos de datos y cadenas de JavaScript y cómo tratarlos como objetos y funciones de llamada.

1.1 JS Statementsv

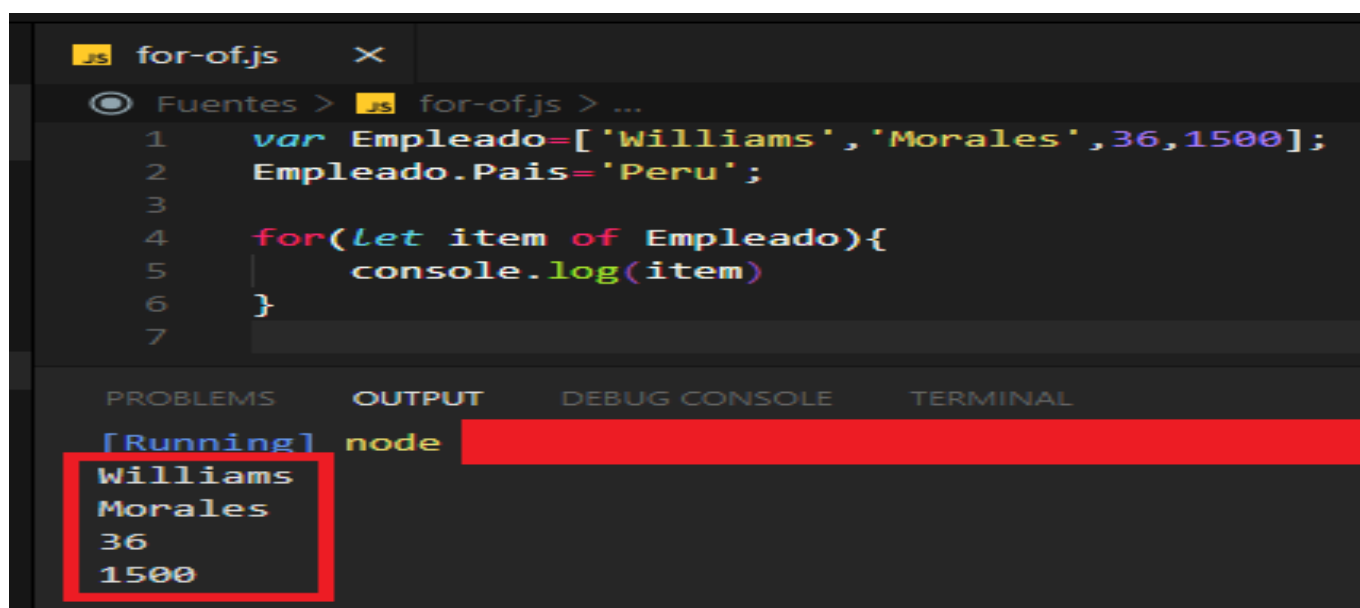
Por: Kevin Alexander De Leon Andrade

Las declaraciones en JavaScript son instrucciones que realizan acciones o ejecutan código. Se componen de palabras clave, operadores, valores y expresiones. Las declaraciones en JavaScript pueden clasificarse en varias categorías, incluyendo declaraciones condicionales, bucles, declaraciones de función, declaraciones de variable, entre otras.

Las declaraciones condicionales: se utilizan para ejecutar cierto bloque de código si se cumple una condición. Los ejemplos más comunes son las declaraciones if, if-else y switch.

Los bucles: permiten repetir un bloque de código varias veces. Los bucles más utilizados en JavaScript son for, while y do-while.

Las declaraciones de función: se utilizan para crear funciones reutilizables que pueden ser invocadas en diferentes partes del código. Se declaran con la palabra clave "function" seguida del nombre de la función y un par de paréntesis que pueden contener parámetros. Las declaraciones de variable se utilizan para crear variables en JavaScript. Se declaran utilizando las palabras clave "var", "let" o "const", seguidas del nombre de la variable y opcionalmente se les puede asignar un valor inicial.



```

for-of.js
Fuentes > JS for-of.js > ...
1  var Empleado=['Williams', 'Morales', 36, 1500];
2  Empleado.Pais='Peru';
3
4  for(let item of Empleado){
5      console.log(item)
6  }
7
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
[Running] node
Williams
Morales
36
1500
  
```

Imagen: Ejemplo del bucle

Fuente: <http://estilow3b.com/introduccion-bucles-javascript-for-while-do-in-of-break-continue/>

1.2 JS Comments

Los comentarios en JavaScript (JS) son fragmentos de texto que se agregan al código fuente para proporcionar información y aclaraciones sobre el funcionamiento del programa. Los comentarios no se ejecutan como parte del programa y no tienen ningún impacto en el comportamiento del código.

```

1  console.log('hello');
2
3  /** Pushes a message to the queue for sending to Discord. */
4  const pushMsgToQueue = (msgObj) => {
5      queue.push(msgObj)
6  }
7
8  /**
9   * Pushes a message to the queue for sending to Discord.
10   */
11  const pushMsgToQueue2 = (msgObj) => {
12      queue.push(msgObj)
13  }
14
15  /** Pushes a message
16   * to the queue for sending
17   * to Discord. */
18  const pushMsgToQueue3 = (msgObj) => {
19      queue.push(msgObj)
20  }
21
22  console.log(pushMsgToQueue);
23  console.log(pushMsgToQueue2);
24  console.log(pushMsgToQueue3);
25

```

Imagen: Ejemplo de comentario

Fuente: <https://github.com/microsoft/vscode/issues/64691>

1.3 JS Variables

Las variables en JavaScript son contenedores que se utilizan para almacenar valores.

Var: tienen un ámbito de función y pueden ser reasignadas y redeclaradas dentro de su ámbito.

Let: Las variables declaradas con "let" pueden ser reasignadas, pero no redeclaradas dentro del mismo ámbito.

const: Las variables declaradas con "const" tienen ámbito de bloque al igual que "let", pero no pueden ser reasignadas ni redeclaradas.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Introducción a las variables de texto JS</title>
5          <meta charset="utf-8">
6          <meta name="author" content="Francesc Ricart">
7      </head>
8      <body>
9          <script type="text/javascript">
10
11              /* Ejemplos de variables de texto */
12
13              var textoA = "comillas dobles";
14              var textoB = 'comillas simples'
15              var textoC = "comillas 'simples' dentro de comillas dobles"
16              //var textoD = "incorrecto"
17              //var textoE = 'incorrecto';
18
19              console.log(textoA);
20              console.log(textoB);
21              console.log(textoC);
22
23          </script>
24      </body>
25  </html>

```

Imagen: Ejemplo de variable var

Fuente: <https://francescricart.com/tipos-variables-javascript/>

1.4 JS Data Types

Números (Numbers): Los números en JavaScript se representan sin comillas y pueden ser enteros o decimales. Por ejemplo: 10, 3.14.

Cadenas de texto (Strings): Las cadenas de texto en JavaScript se representan entre comillas simples (') o comillas dobles ("). Por ejemplo: "Hola", 'Mundo'.

Booleanos: Los valores booleanos representan un estado lógico y pueden ser verdadero (true) o falso (false). Por ejemplo: true, false.

Arreglos (Arrays): Los arreglos son colecciones ordenadas de elementos. Pueden contener diferentes tipos de datos y se representan con corchetes []. Por ejemplo: [1, 2, 3], ['manzana', 'banana', 'naranja'].

Objetos (Objects): Los objetos en JavaScript son colecciones de pares clave-valor y se representan con llaves {}. Por ejemplo: {nombre: 'Juan', edad: 25}.

Nulos (Null): Null se utiliza para representar la ausencia intencional de cualquier objeto o valor. Por ejemplo: null.

Indefinidos (Undefined): Undefined se utiliza cuando una variable ha sido declarada pero no tiene un valor asignado. Por ejemplo: undefined.

Símbolos (Symbols): Los símbolos son valores únicos e inmutables que se utilizan como identificadores de propiedades de objetos. Se crean mediante la función Symbol(). Por ejemplo: let sym = Symbol('descripción').

```

eof("Hola") // string
eof(12) // integer
eof(true) // boolean
eof(undefined) // undefined
eof(null) // object
eof({}) // object
eof(Symbol()) // symbol
eof(1n) // bigint
eof(function(){} ) // function

```

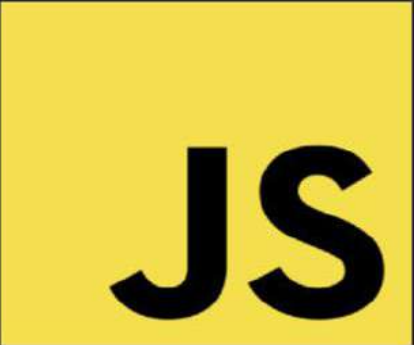


Imagen: Ejemplo de Js data types

Fuente: <https://dev.to/juliansci/javascript-data-types-112g>

1.5 JS Objects

Un objeto en JavaScript es una entidad que agrupa datos y funciones relacionadas en una estructura. Puede considerarse como una colección de propiedades, donde cada propiedad tiene un nombre y un valor asociado.

```
let person = {  
  name: 'John',  
  age: 20  
};
```

Keys --- { } --- Values

Imagen: Ejemplo de Js objects

Fuente: <https://www.programiz.com/javascript/object>

11. JS Functions

Por: Marta Díaz

Una función en JavaScript es similar a un procedimiento un conjunto de instrucciones que realiza una tarea o calcula un valor, pero para que un procedimiento califique como función, debe tomar alguna entrada y devolver una salida donde hay alguna relación obvia entre la entrada y la salida.

Los parámetros en la llamada a una función son los argumentos de la función. Los argumentos se pasan a las funciones por valor. Si la función cambia el valor de un argumento, este cambio no se refleja globalmente ni en la llamada de la función. Sin embargo, las referencias a objetos también son valores, y son especiales: si la función cambia las propiedades del objeto referenciado, ese cambio es visible fuera de la función, tal y como se muestra en el siguiente ejemplo:

```

/* Declarando la función 'myFunc' */
function myFunc(elobjeto)
{
    elobjeto.marca= "Toyota";
}

/*
 * Declarando la variable 'mycar';
 * Se crea e inicializa el nuevo objeto;
 * para hacer referencia a él mediante 'mycar'
 */
var mycar = {
    marca: "Honda",
    modelo: "Accord",
    año: 1998
};

```

Imagen: Function

Fuente: <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Functions>

12. JS Scope

El scope puede definirse como el alcance que una variable tendrá en tu código. En otras palabras, el scope decide a qué variables tienes acceso en cada parte del código. Existen dos tipos de scope, el scope global y el scope local.

Cuando puedes acceder a una variable únicamente en cierta parte del código, se dice que esa variable está declarada en el scope local. Estas también son conocidas como variables locales. Un ejemplo de esto es cuando una variable está declarada dentro de un bloque o una función. Si tratas de acceder a esta variable fuera de esta función o bloque, tendrás un error que dirá que la variable no está definida.

```
function platzi() {
  const soyEstudiante = true;
  console.log(soyEstudiante);
}

platzi(); // true
console.log(soyEstudiante); // soyEstudiante is not defined
```

Imagen: scope local

Fuente: <https://platzi.com/blog/como-funciona-el-scope-en-javascript/#:~:text=Comparte%20en%3A-%C2%BFQu%C3%A9%20es%20el%20scope%20en%20JavaScript%3F,global%20y%20el%20scope%20local>.

Se dice que una variable está en el scope global cuando está declarada fuera de una función o de un bloque. También son conocidas como variables globales. Puedes acceder a este tipo de variables desde cualquier parte de tu código, ya sea dentro o fuera de una función.

```
const soyEstudiante = true;

function platzi() {
  console.log(soyEstudiante);
}

platzi(); //true
console.log(soyEstudiante); //true
```

Imagen: scope global

Fuente: <https://platzi.com/blog/como-funciona-el-scope-en-javascript/#:~:text=Comparte%20en%3A-%C2%BFQu%C3%A9%20es%20el%20scope%20en%20JavaScript%3F,global%20y%20el%20scope%20local>.

13. JS Events

Los eventos son la manera que tenemos en Javascript de controlar las acciones de los visitantes y definir un comportamiento de la página cuando se producen. Cuando un usuario visita una página web e interactúa con ella se producen los eventos y con Javascript podemos definir qué queremos que ocurra cuando se produzcan los eventos.

Para entender los eventos necesitamos conocer algunos conceptos básicos:

Evento: Es algo que ocurre. Generalmente los eventos ocurren cuando el usuario interactúa con el documento, pero podrían producirse por situaciones ajenas al usuario, como una imagen que no se puede cargar porque esté indisponible.

Tipo de evento: Es el tipo del suceso que ha ocurrido, por ejemplo, un clic sobre un botón o el envío de un formulario. Cada tipo de elemento de la página ofrece diversos tipos de eventos de Javascript. En esta página puedes saber cuáles son los tipos de eventos en Javascript.

Manejador de evento: es el comportamiento que nosotros podemos asignar como respuesta a un evento. Se especifica mediante una función Javascript, que se asocia a un tipo de evento en concreto. Una vez asociado el manejador a un tipo de evento sobre un elemento de la página, la vez que ocurre ese tipo de evento sobre ese elemento en concreto, se ejecutará el manejador de evento asociado.

14. JS Strings

El tipo String de JavaScript se utiliza para representar datos textuales. Es un conjunto de "elementos" de valores enteros sin signo de 16 bits (unidades de código UTF-16). Cada elemento de la cadena de caracteres ocupa una posición en la cadena. El primer elemento está en el índice 0, el siguiente en el índice 1, y así sucesivamente. La longitud de una cadena es el número de elementos que contiene. Puedes crear cadenas utilizando cadenas literales u objetos string.

15. JS String Methods

Cadenas

literales:

Puedes crear cadenas simples utilizando comillas simples o dobles



```
'foo'
"bar"
```

Imagen: string methods

Fuente: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Text_formatting

Se pueden crear cadenas más avanzadas usando secuencias de escape: Secuencias de escape hexadecimales

El número después de \x se interpreta como un número del Sistema Hexadecimal



```
'\xA9' // "©"
```

Imagen: string methods

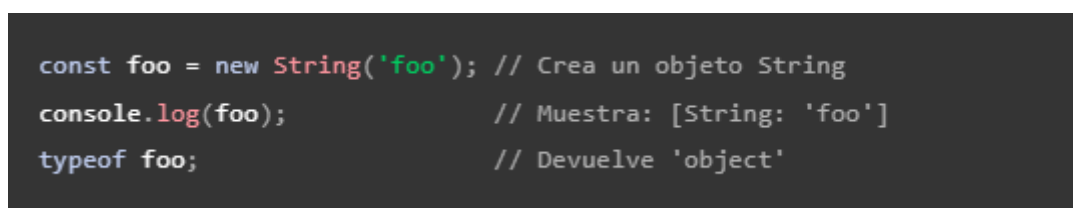
Fuente: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Text_formatting

Secuencias de escape Unicode:

Las secuencias de escape Unicode requieren al menos cuatro dígitos hexadecimales después de \u

Objetos String:

El objeto String es una envoltura alrededor del tipo de dato primitivo string



```
const foo = new String('foo'); // Crea un objeto String
console.log(foo);             // Muestra: [String: 'foo']
typeof foo;                   // Devuelve 'object'
```

Imagen: string methods

Fuente: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Text_formatting

Puedes llamar a cualquiera de los métodos del objeto String en un valor de cadena literal: JavaScript automáticamente convierte la cadena literal en un objeto String temporal, llama al método y luego descarta el objeto String temporal. También puedes usar la propiedad String.length con una cadena literal:

Debes usar cadenas literales a menos que necesites específicamente usar un objeto String, porque los objetos String pueden tener un comportamiento contrario a la intuición. Por ejemplo:

```
const firstString = '2 + 2';           // Crea un valor de cadena literal
const secondString = new String('2 + 2'); // Crea un objeto String
eval(firstString);                     // Devuelve el número 4
eval(secondString);                   // Devuelve la cadena "2 + 2"
```

Imagen: string methods

Fuente: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Text_formatting

Un objeto String tiene una propiedad, `length`, que indica el número de unidades de código UTF-16 en la cadena. Por ejemplo, el siguiente código asigna a `helloLength` el valor 13, porque "¡Hola, mundo!" tiene 13 caracteres, cada uno representado por una unidad de código UTF-16. Puedes acceder a cada unidad de código utilizando la notación de corchete de los arreglos. No puedes cambiar caracteres individuales porque las cadenas son objetos inmutables similares a los arreglos:

```
const hello = '¡Hola, mundo!';
const helloLength = hello.length;
hello[0] = 'L'; // Esto no tiene ningún efecto, porque las cadenas son inmutables
hello[1];      // Esto devuelve "H"
```

Imagen: string methods

Fuente: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Text_formatting

1.1 Numbers en JavaScript

Por: Oscar Echeverría

"Numbers" se refiere al tipo de dato numérico utilizado para representar valores numéricos. Los números en JavaScript pueden ser enteros o decimales y se utilizan para realizar cálculos matemáticos y operaciones aritméticas.

JavaScript proporciona una serie de operadores y funciones integradas para trabajar con números, como sumar (+), restar (-), multiplicar (*), dividir (/), entre otros. También hay funciones como `parseInt()` y `parseFloat()` que permiten convertir cadenas de texto en números enteros y decimales, respectivamente.

Ejemplos:

```
let edad = 25;
// Declaración de un número entero
```

```
let precio = 99.99;
// Declaración de un número decimal
```

```
// También se pueden asignar números directamente a variables sin
// especificar el tipo de dato
```

```
let cantidad = 10;
```

```
// Los números también se pueden utilizar en operaciones aritméticas
```

```
let total = precio * cantidad;
```

1.1.1 JS Number Methods

Son métodos incorporados en el objeto global "Number" que proporcionan funcionalidades específicas para trabajar con números. Estos métodos permiten realizar diversas operaciones y manipulaciones en los valores numéricos.

Ejemplos:

- `Number.parseFloat()`: Convierte una cadena en un número de punto flotante (decimal).
- `Number.parseInt()`: Convierte una cadena en un número entero.
- `Number.isNaN()`: Comprueba si un valor es NaN (No es un número).
- `Number.isFinite()`: Comprueba si un valor es un número finito.
- `Number.isInteger()`: Comprueba si un valor es un número entero.
- `Number.MAX_VALUE`: Proporciona el valor máximo representable en JavaScript.
- `Number.MIN_VALUE`: Proporciona el valor mínimo representable en JavaScript.
- `Number.toFixed()`: Formatea un número con una cantidad fija de decimales.
- `Number.toPrecision()`: Formatea un número con una longitud total fija.
- `Number.toString()`: Convierte un número en una cadena de texto en una base específica.

1.1.2 JS Operators

Son símbolos o palabras clave utilizados para realizar operaciones y manipulaciones en los valores y variables. Los operadores permiten combinar, comparar y manipular datos, y son fundamentales para la escritura de expresiones y la ejecución de instrucciones en JavaScript.

Ejemplos:

Operadores aritméticos:

```
let a = 5;
```

```
let b = 3;
```

```
let suma = a + b;
```

```
let multiplicacion = a * b;
```

Operadores de asignación:

```
let x = 10;
```

```
x += 5;
```

Operadores de comparación:

```
let a = 5;
```

```
let b = 3;
```

```
console.log(a > b);
```

Operadores lógicos:

```
let x = 5;
```

```
let y = 10;
```

```
console.log(x > 0 && y < 15);
```

1.1.3 JS Math

Es un objeto incorporado que proporciona métodos y constantes para realizar operaciones matemáticas más complejas. El objeto "Math" en JavaScript no se instancia, es decir, no se crea con la palabra clave "new", sino que se accede directamente a sus métodos y propiedades a través del nombre del objeto.

Ejemplos:

Métodos trigonométricos: `Math.sin()`, `Math.cos()`, `Math.tan()`, etc., para calcular las funciones trigonométricas seno, coseno y tangente de un ángulo.

Métodos exponenciales y logarítmicos: `Math.exp()`, `Math.log()`, `Math.log10()`, etc., para calcular funciones exponenciales y logarítmicas.

Métodos de redondeo: `Math.round()`, `Math.floor()`, `Math.ceil()`, para redondear números a su entero más cercano, al entero inferior o al entero superior, respectivamente.

Métodos de valor absoluto: `Math.abs()`, para obtener el valor absoluto de un número.

Métodos de generación de números aleatorios: `Math.random()`, para generar un número aleatorio entre 0 (inclusive) y 1 (exclusivo).

Métodos para obtener el máximo y el mínimo: `Math.max()`, `Math.min()`, para obtener el valor máximo y mínimo de un conjunto de números.

Constantes matemáticas: `Math.PI`, `Math.E`, para acceder a los valores de pi y el número de Euler (e), respectivamente.

1.1.4 JS Dates

Se refiere al manejo de fechas y horas utilizando el objeto Date incorporado. El objeto Date proporciona métodos y propiedades para trabajar con fechas, permitiendo la creación, manipulación y formato de fechas y horas en JavaScript.

Ejemplos:

```
let currentDate = new Date(); // Fecha y hora actual
let customDate = new Date(2023, 5, 1); // 1 de junio de 2023
let year = customDate.getFullYear(); // Obtener el año (2023)
let month = customDate.getMonth(); // Obtener el mes (5, ya que los meses se indexan desde 0)
let day = customDate.getDate(); // Obtener el día (1)
customDate.setHours(10); // Establecer las horas a 10
customDate.setMinutes(30); // Establecer los minutos a 30
let formattedDate = customDate.toLocaleString(); // Formatear la fecha a una cadena legible
let tomorrow = new Date();
tomorrow.setDate(tomorrow.getDate() + 1); // Sumar un día a la fecha actual
let isAfter = tomorrow > currentDate; // Comparar fechas
```

1.1.5 JS Date Methods

Son métodos incorporados en el objeto Date que permiten realizar diversas operaciones y manipulaciones en las fechas y horas. Estos métodos proporcionan funcionalidades para obtener y establecer componentes de fecha y hora, realizar cálculos, formatear fechas y más.

Ejemplos:

- `getFullYear()`: Devuelve el año de una fecha como un número de cuatro dígitos.
- `getMonth()`: Devuelve el mes de una fecha como un número, donde enero es 0 y diciembre es 11.
- `getDate()`: Devuelve el día del mes de una fecha como un número.
- `getDay()`: Devuelve el día de la semana de una fecha como un número, donde domingo es 0 y sábado es 6.
- `getHours()`: Devuelve las horas de una fecha como un número.
- `getMinutes()`: Devuelve los minutos de una fecha como un número.
- `getSeconds()`: Devuelve los segundos de una fecha como un número.

22. JS Arrays

Por: Aroldo López

JavaScript Arrays (arreglos en JavaScript) son estructuras de datos que permiten almacenar múltiples valores en una única variable. Los arreglos son utilizados para almacenar colecciones de datos relacionados y son una parte fundamental de la programación en JavaScript.

```

1  const array = ['banana', 'apple', 'orange', 'pear']
2  const resultArray = array.forEach((element, i) => {
3      console.log(`${element} se encuentra en el indice ${i}`)
4  });
5  // banana se encuentra en el indice 0
6  // apple se encuentra en el indice 1
7  // orange se encuentra en el indice 2
8  // pear se encuentra en el indice 3

```

Imagen: En este ejemplo, el método `forEach` se llama en el arreglo `miArray`. Recibe una función de devolución de llamada como argumento, que se ejecutará para cada elemento del arreglo. La función de devolución de llamada recibe un parámetro, que es el elemento actual que se está iterando en cada iteración. Dentro de la función de devolución de llamada, utilizamos `console.log` para mostrar cada elemento en la consola.

Fuente: https://res.cloudinary.com/practicaldev/image/fetch/s--LwaMt-IT--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https://dev-to-uploads.s3.amazonaws.com/uploads/articles/kxv5bwmtsptuohcd2dqr.png

23. JS Array Methods

Los métodos de los arreglos en JavaScript son funciones incorporadas que se pueden utilizar para manipular y trabajar con arreglos de manera más conveniente y eficiente. Estos métodos proporcionan funcionalidades específicas para agregar, eliminar, buscar, filtrar, ordenar y transformar elementos dentro de un arreglo.

```

/**
 * reduce
 */

const arr = [1, 2, 3, 4, 5, 6];

const sum = arr.reduce((total, value) => total + value, 0);
console.log(sum); // 21

// @khriztianmoreno

```

Imagen: realiza una suma de todos los elementos en el arreglo 'arr' utilizando el método 'reduce' y luego muestra el resultado de la suma en la consola.

Fuente: https://cdn-images-1.medium.com/max/1600/1*iftqS3X2yhltJdiqGg-k_A.png

24. JS Booleans

Los valores booleanos son un tipo de dato que puede tener uno de dos valores: **true** (verdadero) o **false** (falso). Los valores booleanos son utilizados para representar la lógica binaria en programación, donde **true** representa algo verdadero o válido, y **false** representa algo falso o inválido.

```
public class Example
{
    public static void main(String[] args)
    {
        boolean blue = false;
        boolean green = true;
    }
}
```

Imagen: Ejemplo de valores booleanos (true y false)

Fuente: <https://i.ytimg.com/vi/K4jKFe1rN8/maxresdefault.jpg>

25. JS RegExp

Regular Expression es un objeto integrado en JavaScript que permite trabajar con patrones de búsqueda y manipulación de texto. Las expresiones regulares son secuencias de caracteres que forman un patrón de búsqueda y se utilizan para realizar operaciones de búsqueda y manipulación en cadenas de texto.

Con JavaScript RegExp, puedes realizar diversas operaciones en cadenas de texto, como buscar coincidencias, extraer partes de un texto, reemplazar texto y validar formatos. Las expresiones regulares son especialmente útiles cuando necesitas realizar búsquedas avanzadas o realizar transformaciones en el texto de manera flexible.

Example: `var myPattern = /@gmail.com/i;`
`var myString = 'mark.christ@gmail.com';`
`console.log(myString.search(myPattern));`
`console.log(myString.replace(myPattern, '@facebook.com'));`

Output:

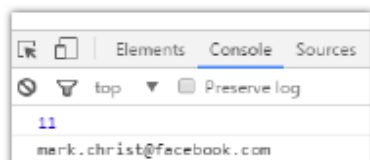


Imagen: Ejemplo de un RegExp

Fuente: https://www.bookofnetwork.com/images/javascript-images/JS_RegExp---Example-01_04Oct16_1546.png

26. JS Type Conversion

JavaScript Type Conversion (conversión de tipos en JavaScript) se refiere al proceso de convertir un valor de un tipo de dato a otro tipo de dato. JavaScript es un lenguaje de programación dinámicamente tipado, lo que significa que las conversiones de tipo se realizan de manera automática en ciertos casos, y también se pueden realizar explícitamente utilizando funciones y operadores específicos.

```
public class Main {
    public static void main(String[] args) {
        float floatVal = 1.0f;
        double doubleVal = 4.0d;
        byte byteVal = 7;
        short shortVal = 7;
        long longVal = 5;

        short result1 = (short)byteVal;
        short result2 = (short)(byteVal - longVal);

        System.out.println("Success");
    }
}
```

Imagen: Ejemplo de conversiones de las distintos tipos de variables

Fuente: <https://cdn.educba.com/academy/wp-content/uploads/2019/11/Type-Conversion-in-Java-eg10.png>

27. JS Comparisons

En JavaScript, las comparaciones se utilizan para evaluar y comparar dos valores o expresiones y determinar su relación lógica. Las comparaciones en JavaScript devuelven un valor booleano (**true** o **false**) que indica si la comparación es verdadera o falsa.

```
// Recomendaciones:
// * Comparación (==), mismo valor
// * Compara (===), mismo valor y mismo tipo de dato
// * Usar mejor (===)
// * Comparación entre objetos (misma referencia), igual objeto en memoria

var x = 4, y = '4'

var jeferson = {
  name: 'Jeferson'
}

var otherPerson = jeferson
```

Imagen: Ejemplo de comparación de dos valores

Fuente: https://static.platzi.com/media/user_upload/class9Comparison-177e7657-8037-42ca-9221-cf0fad4319bb.jpg

1.1 jQuery Hide/Show

Por: Thania Paola García Cos

Las funciones de jQuery show () y hide (), se usan para mostrar y ocultar elementos del DOM de la web, mayoritariamente suelen ser divs, ids, clases, imágenes, elementos de formularios, etiquetas.

1.1.1 Método show

El método show () muestra los elementos seleccionados ocultos. Nota: show () funciona en elementos ocultos con métodos jQuery y display: none en CSS (pero no visibilidad: hidden).

Consejo: para ocultar elementos, mire el método hide ()

1.1.2 Método hide

El método hide () oculta los elementos seleccionados. Sugerencia: Esto es similar a la visualización de propiedades CSS: ninguna.

Nota: Los elementos ocultos no se mostrarán en absoluto (ya no afecta el diseño de la página).

Consejo: para mostrar elementos ocultos, consulte el método show () .

1.1.3 jQuery Fade

En el post de hoy voy a hablar del efecto de jQuery 'fading', este efecto de transiciones consta de 4 funciones:

- fadeIn()
- fadeOut()
- fadeTo()
- fadeToggle()

Son efectos básicos como lo son jQuery show () & hide () y jQuery slideDown(), slideUp() y slideToggle. Estas 4 funciones no solo hacen ocultar o mostrar divs, si no que realizan variaciones de opacidad al ocultar o mostrar los objetos.



Imagen: jQuery Fade

Fuente: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.anerbarrena.com%2Fjquery-fadein-y-fadeout-3114%2F&psig=AOvVaw2ez-EJAtFHROw8_5kKna6c&ust=1685755857729000&source=images&cd=vfe&ved=0CA4QjRxqFwoTCNDv0o24o_8CFQA AAAAdAAAAABAD

1.1.4 jQuery Slide

El efecto slide en jQuery consiste en deslizar un elemento hasta que desaparezca o aparezca en nuestra vista. Este post explicará cómo crear el efecto slide con las diferentes funciones que nos proporciona jQuery. Como sucede con las otras animaciones, podremos asignar una velocidad en milisegundos para conseguir que la animación tenga la fluidez que necesitamos.



Imagen: jQuery Slide

Fuente: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.educba.com%2Fslide-in-jquery%2F&psig=AOvVaw02URbRgcQcQ4yuGVsQp1z4&ust=1685756737014000&source=images&cd=vfe&ved=0CA4QjRxqFwoTCJjBrK-7o_8CFQAAAAAdAAAAABAD

1.1.5 jQuery Animate

jQuery permite animar varias propiedades, con valores numéricos, de CSS en un solo paso. Hemos avanzado bastante en el Manual de jQuery y con los conocimientos adquiridos hasta el momento ya estamos en disposición de aprender cualquier cosa más avanzada en este framework Javascript.

1.1.6 jQuery stop()

la funcionalidad es la de detener las funciones o efectos que se estén ejecutando antes de que finalicen. Otras funciones para trabajar con las ejecuciones de las funciones en jQuery son: delay ()

1.1.7 jQuery Callback

Qué es un callback de funciones jQuery y cómo podemos usar este recurso para asegurarnos que unas funciones se ejecutan después de otras, algo que será muchas veces necesario para funciones de efectos suavizados.



Imagen: jQuery Callback

Fuente: https://www.google.com/url?sa=i&url=https%3A%2F%2Fdesarrolloweb.com%2Farticulos%2Fcallback-funciones-jquery.html&psig=AOvVaw2JibR_Co1BY-O_M-N1anGZ&ust=1685757096858000&source=images&cd=vfe&ved=0CA4QjRxqFwoTCLCu-dm8o_8CFQAAAAAdAAAAABAD

1.1 JQUERY

Por: David Daniel de Jesús Gerónimo Vaquias

jQuery es una biblioteca de JavaScript rápida, compacta y fácil de usar, diseñada para simplificar la manipulación y navegación de documentos HTML, el manejo de eventos, la animación y la comunicación con servidores web utilizando técnicas de Ajax.

1.1.1 jQuery Chaining

jQuery Chaining, también conocido como encadenamiento de métodos en jQuery, es una característica que permite realizar múltiples acciones o manipulaciones en un elemento o conjunto de elementos seleccionados en una sola línea de código.

En lugar de escribir múltiples instrucciones separadas para cada manipulación, se puede encadenar una serie de métodos de jQuery uno tras otro, utilizando el operador de punto (.) para aplicar acciones secuenciales a los elementos seleccionados. Esto permite un código más conciso y legible.

Por ejemplo, en lugar de escribir:

```
$("#miElemento").addClass("rojo");  
$("#miElemento").slideUp();  
$("#miElemento").fadeOut();
```

Se puede encadenar los métodos de la siguiente manera:

```
$("#miElemento").addClass("rojo").slideUp().fadeOut();
```

1.1.2 jQuery Get

jQuery Get es un método de jQuery utilizado para realizar solicitudes HTTP GET asincrónicas en JavaScript. Proporciona una forma sencilla de interactuar con un servidor y obtener datos o recursos sin recargar la página web completa.

El método `$.get()` en jQuery se utiliza para enviar una solicitud GET a un URL específico y recuperar los datos devueltos por el servidor. La respuesta del servidor puede ser en cualquier formato, como texto, JSON o XML.

La sintaxis básica de `$.get()` es la siguiente:

```
$.get(url, data, success, dataType);
```

- url: La URL a la cual se enviará la solicitud GET.
- data (opcional): Los datos que se enviarán al servidor. Puede ser una cadena de consulta o un objeto JavaScript.
- success (opcional): Una función de devolución de llamada que se ejecutará si la solicitud es exitosa. Recibe los datos de respuesta como argumento.
- dataType (opcional): El tipo de datos esperados desde el servidor. Puede ser "xml", "json", "script" o "html". Por defecto, se realiza una suposición inteligente basada en el tipo MIME de la respuesta

```
$.get("https://api.ejemplo.com/datos", { id: 1 }, function(response) {
    // Código para manejar los datos de respuesta
    console.log(response);
}, "json");
```

1.1.3 jQuery Set

No existe un método llamado "jQuery Set" en la biblioteca jQuery. Sin embargo, podría haber una confusión o malentendido en la terminología utilizada.

En jQuery, hay métodos para establecer valores o propiedades de elementos HTML. Algunos de los métodos más comunes para establecer valores son:

- `text()`: Establece el contenido de texto de un elemento. Por ejemplo, `$(selector).text("Hola, mundo!")` establecerá el texto "Hola, mundo!" dentro del elemento seleccionado.
- `html()`: Establece el contenido HTML de un elemento. Puedes utilizar este método para insertar o reemplazar el contenido HTML de un elemento. Por ejemplo, `$(selector).html("<p>Este es un párrafo</p>")` establecerá el contenido HTML del elemento seleccionado como un párrafo.
- `val()`: Establece el valor de un elemento de formulario, como un campo de texto, una casilla de verificación o un botón de opción. Por ejemplo, `$(selector).val("Nuevo valor")` establecerá el valor del elemento seleccionado.



Imagen: Get and Set HTML attributes and content with jQuery

Fuente: <https://learncodeweb.com/jquery/get-and-set-html-attributes-and-content-with-jquery/>

1.1.4 jQuery Add

En jQuery, el método `$.add()` se utiliza para agregar elementos adicionales a una colección existente de elementos seleccionados. Permite combinar elementos adicionales con una selección existente para formar una nueva colección de elementos.

La sintaxis básica de `$.add()` es la siguiente:

```
$(selector).add(elements);
```

selector: El selector que especifica los elementos existentes que se deben incluir en la nueva colección.

elements: Los elementos adicionales que se deben agregar a la colección. Pueden ser elementos HTML, otros selectores o incluso objetos jQuery.

Aquí hay un ejemplo que muestra cómo usar `$.add()`:

```
var selectedElements = $(".clase1"); // Selección inicial de elementos con la clase "clase1"

var additionalElements = $(".clase2"); // Selección adicional de elementos con la clase "clase2"

var combinedElements = selectedElements.add(additionalElements);
```

1.1.5 jQuery Remove

En jQuery, el método `$.remove()` se utiliza para eliminar elementos del DOM (Document Object Model). Permite eliminar un elemento y todos sus elementos secundarios de la página.

La sintaxis básica de `$.remove()` es la siguiente:

```
$(selector).remove();
```

selector: El selector que especifica los elementos que se deben eliminar del DOM.

Aquí hay un ejemplo que muestra cómo usar `$.remove()`:

```
var elementoEliminado = $("#miElemento").remove();
```

1.1.6 jQuery CSS

En jQuery, el método `$.css()` se utiliza para obtener o establecer los estilos CSS de uno o varios elementos seleccionados. Proporciona una forma conveniente de manipular los estilos CSS de los elementos en el DOM (Document Object Model).

La sintaxis básica de `$.css()` es la siguiente:

```
$(selector).css(propertyName);
```

Para establecer el valor de un estilo CSS:

```
$(selector).css(propertyName, value);
```

- **selector:** El selector que especifica los elementos a los que se les aplicarán los estilos CSS.
- **propertyName:** El nombre de la propiedad CSS que se desea obtener o establecer.

- **value:** El valor que se desea establecer para la propiedad CSS.
- Aquí hay algunos ejemplos que muestran cómo usar `$.css()`:

Para obtener el valor de un estilo CSS:

```
var color = $("p").css("color");
```

1.1.7 Classes

En jQuery, hay varios métodos y funciones que se utilizan para trabajar con clases CSS en los elementos seleccionados. Aquí hay algunos de los métodos más comunes:

addClass(): Agrega una o varias clases CSS a los elementos seleccionados. Puedes pasar una o varias clases como argumento. Por ejemplo: `$(selector).addClass("clase1 clase2")`.

removeClass(): Elimina una o varias clases CSS de los elementos seleccionados. Puedes pasar una o varias clases como argumento. Por ejemplo: `$(selector).removeClass("clase1 clase2")`.

toggleClass(): Agrega una clase si no está presente o la elimina si está presente en los elementos seleccionados. Puedes pasar una o varias clases como argumento. Por ejemplo: `$(selector).toggleClass("clase1 clase2")`.

hasClass(): Verifica si al menos uno de los elementos seleccionados tiene una clase específica. Devuelve true o false según corresponda. Por ejemplo: `$(selector).hasClass("miClase")`.

Estos métodos te permiten agregar, eliminar o cambiar las clases CSS de los elementos seleccionados de manera fácil y eficiente. También puedes combinar estos métodos con otros métodos de selección para aplicar cambios a subconjuntos específicos de elementos en tu página web.

```
$(document).ready(function() {  
  $("button").click(function() {  
    $("p").toggleClass("resaltado");  
  });  
});
```



Imagen: Añadir y quitar clases CSS sobre elementos con jQuery

Fuente: <https://desarrolloweb.com/articulos/anadir-quitar-class-jquery.html>

1.1 jQuery css()

Por: Patrick Leonardo Gomez Paz

En jQuery, la función `css()` es un método utilizado para obtener o establecer el valor de una o varias propiedades CSS de los elementos seleccionados.

La sintaxis básica del método `css()` es la siguiente:

```
$(selector).css(propertyName)
```

Donde:

`selector` es la expresión para seleccionar los elementos en el DOM.

`propertyName` es el nombre de la propiedad CSS que se desea obtener o establecer.

Si se utiliza solo el parámetro `propertyName`, el método `css()` devuelve el valor actual de la propiedad CSS para el primer elemento seleccionado. Por ejemplo:

```
var valor = $('.miClase').css('color');
```

En este caso, se obtendría el valor del color de texto establecido para el primer elemento con la clase `miClase`.

Además, se puede utilizar el método `css()` para establecer una o varias propiedades CSS para los elementos seleccionados. Para esto, se pasa un objeto de pares clave-valor como argumento, donde la clave es el nombre de la propiedad CSS y el valor es el nuevo valor que se desea asignar. Por ejemplo:

```
$('.miClase').css({
  'color': 'red',
  'font-size': '16px',
  'background-color': '#f0f0f0'
});
```

En este caso, se establecerían las propiedades `color`, `font-size` y `background-color` con los respectivos valores para todos los elementos con la clase `miClase`.

```
<style>
  div {
    display: block !important;
    color: red !important;
  }
</style>

<script>
  $(document).ready(function () {
    $('div').attr('style', 'color: blue !important');
  });
</script>
```

Color red was defined.

<https://www.encodedna.com>

Now, color blue is important

Imagen: Ejemplo jquery css()

Fuente:

[https://www.google.com/search?q=jquery+css\(\)&rlz=1C1CHZN_esGT938GT938&sxsrf=APwXEdcrE58yzXIUD5dM0Jnzp1JfIkHWzA:1685673144376&source=lnms&tbn=isch&sa=X&ved=2ahUKewiikunnxaP_AhXpRzABHV_YCjCQ_AUoAXoECAEQAw&biw=1366&bih=657&dpr=1#imgsrc=c_msghCKmBZW5M](https://www.google.com/search?q=jquery+css()&rlz=1C1CHZN_esGT938GT938&sxsrf=APwXEdcrE58yzXIUD5dM0Jnzp1JfIkHWzA:1685673144376&source=lnms&tbn=isch&sa=X&ved=2ahUKewiikunnxaP_AhXpRzABHV_YCjCQ_AUoAXoECAEQAw&biw=1366&bih=657&dpr=1#imgsrc=c_msghCKmBZW5M)

1.1.1 jQuery

jQuery es una biblioteca de JavaScript rápida, pequeña y versátil que simplifica la manipulación y el manejo de eventos en el HTML, el manejo de animaciones, el manejo de interacciones con el usuario y la realización de peticiones AJAX.

Fue desarrollada con el objetivo de simplificar y abstraer las tareas comunes en la programación web, proporcionando una sintaxis más concisa y fácil de usar en comparación con JavaScript puro. jQuery se basa en el modelo de selección de elementos similar al de CSS, lo que facilita la manipulación y el acceso a los elementos del DOM.

La principal ventaja de jQuery es su capacidad para interactuar con el DOM de manera sencilla y eficiente. Permite seleccionar elementos del DOM utilizando selectores CSS y luego realizar diversas operaciones, como modificar el contenido, cambiar los estilos, agregar o eliminar elementos, gestionar eventos, etc.

```

<!DOCTYPE html>
<html>
<head>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
  <h1 id="miElemento">Texto original</h1>

  <script>
    $(document).ready(function() {
      $('#miElemento').text('Nuevo texto');
    });
  </script>
</body>
</html>

```

Imagen: Ejemplo jquery

Fuente: https://www.google.com/search?q=jquery&tbn=isch&ved=2ahUKewiHgKjtxaP_AhU3YjABHW_7CIYQ2-cCegQIABAA&oq=jquery&gs_lcp=CgNpbWcQAZIECCMQJzIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDoGCAAQBxAgOgQIABAeOgYIABAIEB5QraIKWK2iCmDcpApoAXAAeACAAC0DiAGVB5IBBzItMi4wLjGYAQcGAAQgAQQnd3Mtd2l6LWltZ8ABAQ&scIent=img&ei=w1R5ZMeSNrfEwbkP7_arsAU&bih=657&biw=1366&rlz=1C1CHZN_esGT938GT938#imgsrc=zF2gNzbls-voSM

1.1.2 Dimensions

En jQuery, puedes utilizar varios métodos para obtener y establecer las dimensiones de los elementos del DOM. Algunos de los métodos más comunes son los siguientes:

1.1.3 .width()

Este método devuelve el ancho calculado del primer elemento seleccionado o establece el ancho de todos los elementos seleccionados. Puedes utilizarlo de las siguientes formas:

```
// Obtener el ancho del primer elemento seleccionado
```

```
var ancho = $(selector).width();
```

```
// Establecer el ancho para todos los elementos seleccionados
```

```
$(selector).width(ancho);
```

1.1.4 height()

Este método funciona de manera similar a width(), pero se aplica a la altura de los elementos. Puedes utilizarlo de la siguiente manera:

```
// Obtener la altura del primer elemento seleccionado
```

```
var altura = $(selector).height();
```

```
// Establecer la altura para todos los elementos seleccionados
```

```
$(selector).height(altura);
```

1.1.5 innerwidth()

Este método devuelve el ancho del contenido de un elemento, incluyendo el padding, pero sin incluir el margen. Puedes utilizarlo de la siguiente forma:

```
// Obtener el ancho interior del primer elemento seleccionado
```

```
var anchoInterior = $(selector).innerWidth();
```

1.2 AJAX

Es una técnica de desarrollo web que permite realizar solicitudes asíncronas al servidor y actualizar parte de una página web sin necesidad de recargarla por completo. En lugar de realizar una solicitud tradicional que resultaría en una carga completa de la página, AJAX permite cargar o enviar datos en segundo plano, lo que mejora la experiencia del usuario al hacer que las interacciones sean más rápidas y fluidas.



Imagen: Ejemplo AJAX

Fuente:

https://www.google.com/search?q=AJAX+jquery&tbm=isch&ved=2ahUKEwiHjtG_yqP_AhWEszEKHefIA28Q2-cCegQIABAA&oeq=AJAX+jquery&gs_lcp=CgNpbWcQAzoHCAAQigUQQzoFCAAQgAQ6BAgAEB46BggAEAgQHjoHCAAQGBCABFCqC1ifJWCVJ2gAcAB4AIABpQOIAawOkgEJMC41LjluMC4xmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&scIent=img&ei=oll5Zle1EITnxgHny4_4Bg#imgrc=MoCcwzy1pUM2tM

1.3 Filter Tables

Filtrar tablas es una tarea común en el desarrollo web que implica filtrar dinámicamente los datos que se muestran en una tabla según la entrada del usuario o criterios específicos. Aquí hay un enfoque general para filtrar tablas usando JavaScript/jQuery:

1.3.1 Marcado HTML

Configure la estructura de su tabla con las etiquetas HTML adecuadas y agregue clases o atributos de datos a los elementos que se utilizarán para el filtrado. Por ejemplo:

```
<input type="text" id="filterInput" placeholder="Search...">
<table id="myTable">
  <thead>
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>Country</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John Doe</td>
      <td>25</td>
      <td>USA</td>
    </tr>
    <!-- More table rows... -->
  </tbody>
</table>
```

Imagen: Ejemplo HTML

Fuente: [https://www.google.com/search?q=marcado+html&tbm=isch&ved=2ahUKEwi77JrEYqP_AhW-joQIHxDiD5kQ2-](https://www.google.com/search?q=marcado+html&tbm=isch&ved=2ahUKEwi77JrEYqP_AhW-joQIHxDiD5kQ2-cCegQIABAA&oq=marcado+html&gs_lcp=CgNpbWcQAzIGCAAQBRAeMgclABAYEIAEOgQIlxAnOggIABCABBCxAzoLCAAQgAQQsQMogwE6BQgAEIAEOgclABCKBRBDUKcGWPOUYIIWaABwAHgAgAGHAogBgxGSAQYwLjExLjGYAQcGAAQgqAQtd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=q1l5ZPvqNL6dkvQP8MS_yAk#imgrc=vUzrKWLwObYMIM)

[cCegQIABAA&oq=marcado+html&gs_lcp=CgNpbWcQAzIGCAAQBRAeMgclABAYEIAEOgQIlxAnOggIABCABBCxAzoLCAAQgAQQsQMogwE6BQgAEIAEOgclABCKBRBDUKcGWPOUYIIWaABwAHgAgAGHAogBgxGSAQYwLjExLjGYAQcGAAQgqAQtd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=q1l5ZPvqNL6dkvQP8MS_yAk#imgrc=vUzrKWLwObYMIM](https://www.google.com/search?q=marcado+html&tbm=isch&ved=2ahUKEwi77JrEYqP_AhW-joQIHxDiD5kQ2-cCegQIABAA&oq=marcado+html&gs_lcp=CgNpbWcQAzIGCAAQBRAeMgclABAYEIAEOgQIlxAnOggIABCABBCxAzoLCAAQgAQQsQMogwE6BQgAEIAEOgclABCKBRBDUKcGWPOUYIIWaABwAHgAgAGHAogBgxGSAQYwLjExLjGYAQcGAAQgqAQtd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=q1l5ZPvqNL6dkvQP8MS_yAk#imgrc=vUzrKWLwObYMIM)

1.3.2 Lógica JavaScript/jQuery

Escribe código JavaScript/jQuery para manejar la funcionalidad de filtrado. Aquí hay un ejemplo usando jQuery:

```
$(document).ready(function() {
  $('#filterInput').on('keyup', function() {
    var value = $(this).val().toLowerCase(); // Get the user input and
    $('#myTable tbody tr').filter(function() {
      $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1);
    });
  });
});
```

Imagen: Ejemplo HTML

Fuente: [https://www.google.com/search?q=marcado+html&tbm=isch&ved=2ahUKEwi77JrEYqP_AhW-joQIHxDiD5kQ2-](https://www.google.com/search?q=marcado+html&tbm=isch&ved=2ahUKEwi77JrEYqP_AhW-joQIHxDiD5kQ2-cCegQIABAA&oq=marcado+html&gs_lcp=CgNpbWcQAzIGCAAQBRAeMgclABAYEIAEOgQIlxAnOggIABCABBCxAzoLCAAQgAQQsQMogwE6BQgAEIAEOgclABCKBRBDUKcGWPOUYIIWaABwAHgAgAGHAogBgxGSAQYwLjExLjGYAQcGAAQgqAQtd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=q1l5ZPvqNL6dkvQP8MS_yAk#imgrc=vUzrKWLwObYMIM)

[cCegQIABAA&oq=marcado+html&gs_lcp=CgNpbWcQAzIGCAAQBRAeMgclABAYEIAEOgQIlxAnOggIABCABBCxAzoLCAAQgAQQsQMogwE6BQgAEIAEOgclABCKBRBDUKcGWPOUYIIWaABwAHgAgAGHAogBgxGSAQYwLjExLjGYAQcGAAQgqAQtd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=q1l5ZPvqNL6dkvQP8MS_yAk#imgrc=vUzrKWLwObYMIM](https://www.google.com/search?q=marcado+html&tbm=isch&ved=2ahUKEwi77JrEYqP_AhW-joQIHxDiD5kQ2-cCegQIABAA&oq=marcado+html&gs_lcp=CgNpbWcQAzIGCAAQBRAeMgclABAYEIAEOgQIlxAnOggIABCABBCxAzoLCAAQgAQQsQMogwE6BQgAEIAEOgclABCKBRBDUKcGWPOUYIIWaABwAHgAgAGHAogBgxGSAQYwLjExLjGYAQcGAAQgqAQtd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=q1l5ZPvqNL6dkvQP8MS_yAk#imgrc=vUzrKWLwObYMIM)

1.3.2 Exposición Técnica

Una exposición técnica es un evento o actividad en la que se presentan y exhiben productos, servicios o proyectos relacionados con el ámbito técnico, científico o tecnológico. Estas exposiciones suelen ser organizadas por empresas, instituciones educativas, organizaciones sin fines de lucro o entidades gubernamentales con el objetivo de mostrar avances, innovaciones y desarrollos en un determinado campo técnico.

Estas exposiciones suelen ser oportunidades importantes para establecer contactos, intercambiar conocimientos, generar colaboraciones y promover el intercambio de ideas entre los participantes. Además, pueden incluir conferencias, charlas o mesas redondas en las que expertos del sector comparten sus experiencias y conocimientos con la audiencia.



Imagen: Ejemplo Exposicion

Fuente:

[https://www.google.com/search?q=exposicion+tecnica&tbm=isch&ved=2ahUKEwj69NPezaP_AhXmkYQIHUd7BvQQ2-](https://www.google.com/search?q=exposicion+tecnica&tbm=isch&ved=2ahUKEwj69NPezaP_AhXmkYQIHUd7BvQQ2-cCegQIABAA&oeq=exposicion+tecn&gs_lcp=CgNpbWcQARgBMgUIABCABDIFCAAQgAQyBQgAEIAEMgYIABAIEB4yBggAEAgQHjIGCAAQCBAeMgYIABAIEB4yBggAEAgQHjIGCAAQCBAeMgYIABAIEB5QAFgAYI4PaABwAHgAgAH9A4gBmwaSAQcyLTEuNS0xmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=CF15ZPrQJuaJkVQPx_aZoA8#imgrc=Z52ZurjNSQUO7M)

[cCegQIABAA&oeq=exposicion+tecn&gs_lcp=CgNpbWcQARgBMgUIABCABDIFCAAQgAQyBQgAEIAEMgYIABAIEB4yBggAEAgQHjIGCAAQCBAeMgYIABAIEB4yBggAEAgQHjIGCAAQCBAeMgYIABAIEB5QAFgAYI4PaABwAHgAgAH9A4gBmwaSAQcyLTEuNS0xmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=CF15ZPrQJuaJkVQPx_aZoA8#imgrc=Z52ZurjNSQUO7M](https://www.google.com/search?q=exposicion+tecn&gs_lcp=CgNpbWcQARgBMgUIABCABDIFCAAQgAQyBQgAEIAEMgYIABAIEB4yBggAEAgQHjIGCAAQCBAeMgYIABAIEB5QAFgAYI4PaABwAHgAgAH9A4gBmwaSAQcyLTEuNS0xmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=CF15ZPrQJuaJkVQPx_aZoA8#imgrc=Z52ZurjNSQUO7M)

1.1 Exposición Técnica

Por: Joshua González

1.1.1 Briefing

Un el equipo de desarrollo de software. Durante este briefing, se recopilan los requisitos, objetivos y especificaciones del proyecto de software. Briefing se refiere a la reunión o proceso de comunicación entre el cliente o el solicitante y

1.1.2 Planning

Un planning se refiere a un proceso o documento que establece la secuencia de tareas, los plazos y los recursos necesarios para desarrollar un proyecto de software de manera eficiente. También se conoce como planificación de proyectos de desarrollo de software.

1.1.3 FrontEnd

El frontend se refiere a la parte de un sistema de software o una aplicación que los usuarios pueden ver y con la que interactúan directamente. También se le conoce como la capa de presentación.



Imagen: Ejemplo del FrontEnd

Fuente: <https://wellaggio.com/wp-content/uploads/2017/12/Wellaggio-Que%CC%81-reglas-debo-seguir-para-tener-un-buen-disen%CC%83o-web.jpg>

1.1.4 UI de software



UI (User Interface) se refiere a la interfaz de usuario de un software o una aplicación. Es el medio por el cual los usuarios interactúan con el sistema, permitiéndoles realizar acciones, visualizar información y recibir retroalimentación.

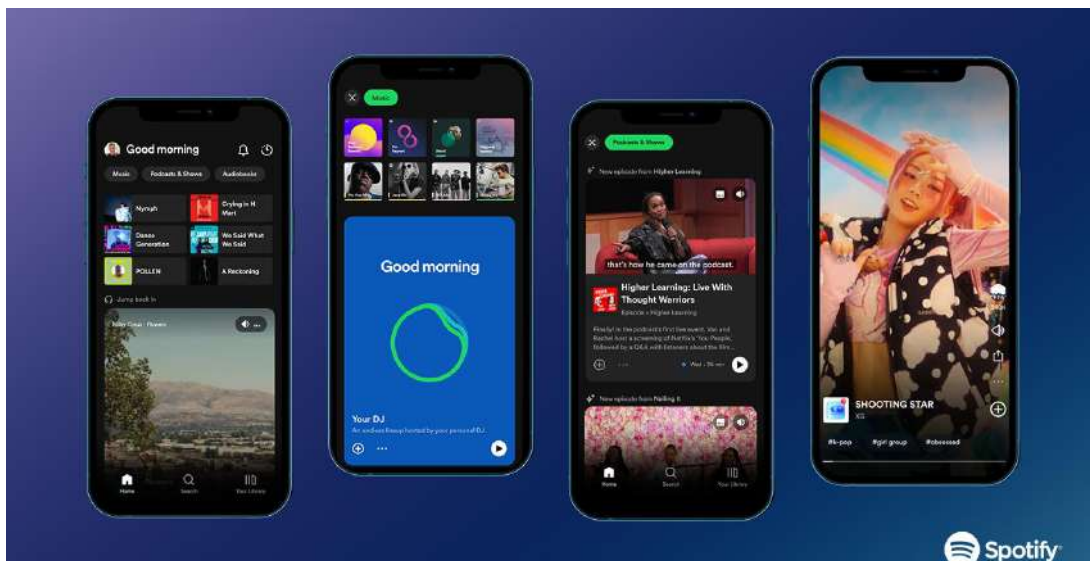


Imagen: Ejemplo de UI (Spotify)

Fuente: https://storage.googleapis.com/pr-newsroom-wp/1/2023/03/New-Home-Feed_FTR-Header-1440x733.png

1.1.5 BackEnd de Programación-Lanzamiento

El backend, también conocido como el lado del servidor, se refiere a la parte de un sistema de software o una aplicación que se encarga del procesamiento y la gestión de los datos, así como de la lógica empresarial subyacente.



Imagen: Ejemplo del BackEnd

Fuente: <https://kinsta.com/es/wp-content/uploads/sites/8/2021/12/back-end-developer-1024x512.png>

1.1.6 Mantenimiento del Software

El mantenimiento de software es el proceso de realizar modificaciones, correcciones y mejoras en un software después de su implementación inicial. Consiste en mantener y garantizar el correcto funcionamiento del software a lo largo del tiempo, así como adaptarlo a los cambios y requisitos del entorno en el que se utiliza.

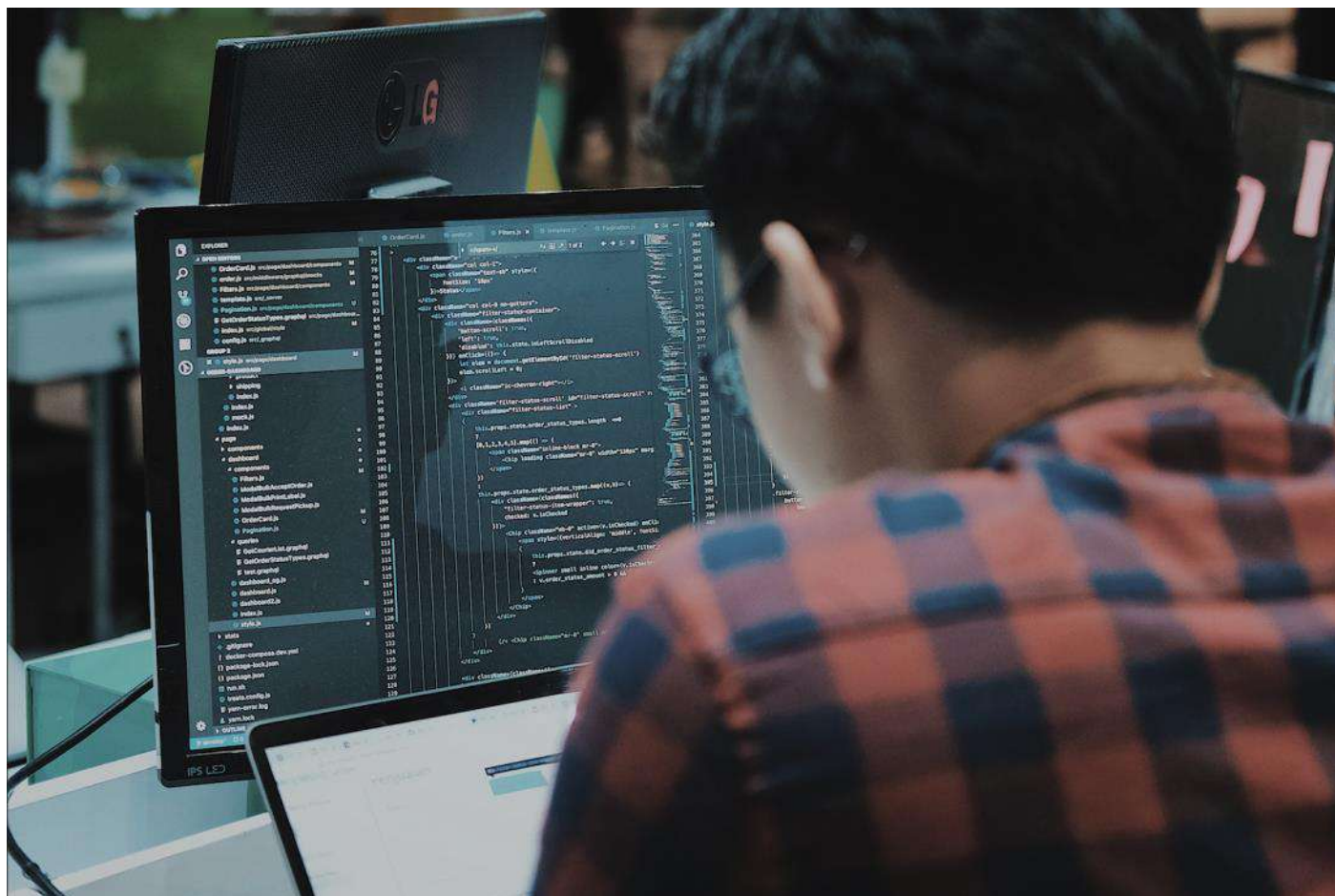


Imagen: Ejemplo de Mantenimiento

Fuente: <https://evolbit.net/blog/wp-content/uploads/2022/02/Captura-de-Pantalla-2022-02-23-a-las-14.52.03-1089x730.png>

1 introducción y clasificación de las redes

Por: **Brandon Grijalva**

Las redes son estructuras interconectadas que permiten la comunicación y el intercambio de información entre diferentes dispositivos o nodos. Pueden clasificarse en diferentes categorías según sus características y aplicaciones.

Clasificación de las redes:

1. Según su tamaño y alcance geográfico:

- Redes de área personal (PAN): Son redes pequeñas y de corto alcance, diseñadas para la comunicación entre dispositivos personales cercanos, como teléfonos móviles y computadoras portátiles.
- Redes de área local (LAN): Se utilizan para interconectar dispositivos en una ubicación geográfica limitada, como una oficina, un edificio o un campus.
- Redes de área metropolitana (MAN): Cubren una zona geográfica más amplia, como una ciudad o un área metropolitana, y suelen ser propiedad de una organización o proveedor de servicios.
- Redes de área amplia (WAN): Son redes que abarcan grandes distancias y conectan dispositivos en diferentes ubicaciones geográficas, como sucursales de una empresa ubicadas en diferentes ciudades o incluso países.

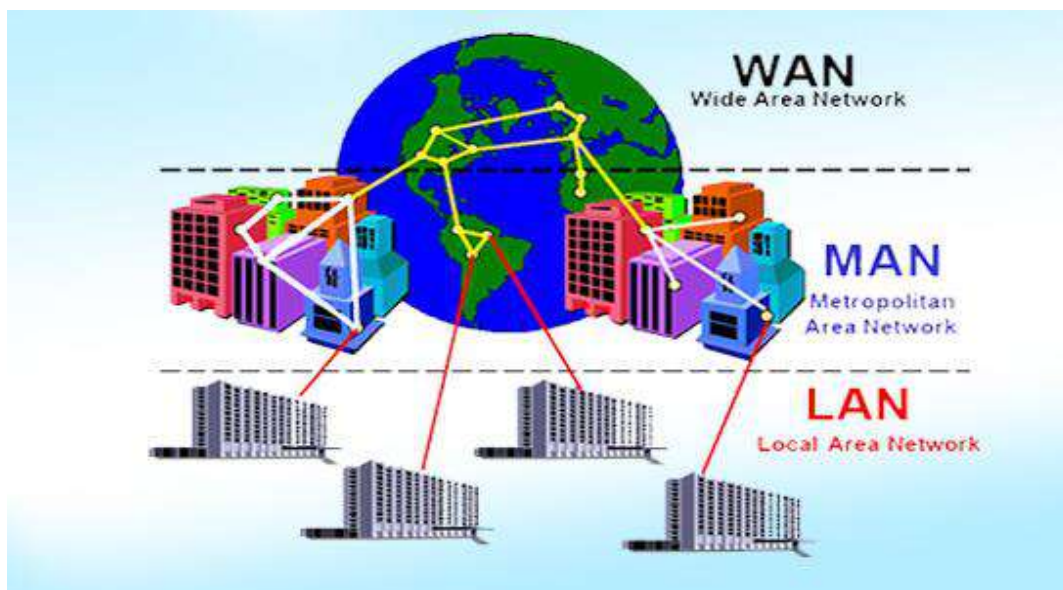


Imagen: Redes(WAN,MAN Y LAN)

Fuente: <https://userscontent2.emaze.com/images/ae240908-f850-4733-bb7c-0494391c1842/5b42c324-2b56-4ad1-8590-96283838ef17.jpg>

1.1 Que es una red

Una red es una estructura interconectada de dispositivos que permite la comunicación y el intercambio de información entre ellos. Puede ser tanto física como virtual, y se utiliza para compartir recursos, como archivos, impresoras, y servicios, como el acceso a internet.



Imagen: Interconexión de la red en el mundo

Fuente: <https://www.tecnologia-informatica.com/wp-content/uploads/2020/02/tarjetas-de-red-1.jpg>

1.2 Comunicación de datos

La comunicación de datos se refiere al proceso de transmitir información entre dispositivos o sistemas a través de un medio de transmisión, como cables, ondas de radio o redes inalámbricas. Esta comunicación puede involucrar el intercambio de texto, imágenes, audio, video u otros tipos de datos.



Imagen: Red conectada por conexión física

Fuente: <https://th.bing.com/th/id/OIP.JcWdNaeHIRFTTnthGRgjRgHaEo?pid=ImgDet&rs=1>

1.3 Canal de comunicación

Un canal de comunicación es el medio físico o virtual utilizado para transmitir información entre un emisor y un receptor. Puede ser un cable, una conexión inalámbrica, una fibra óptica, una red de telefonía, entre otros.



Imagen: Distintos tipos de canales de comunicación

Fuente:

<https://th.bing.com/th/id/R.095307460c4dd4e314058bca642160ce?rik=x3H1RvN7Nv4%2btA&pid=ImgRaw&rs=0>

1.1 introducción y clasificación de las redes

Por: Braulio José Alexander Guarca Lima

1.1.1 Red de Computadoras

Es un conjunto de dispositivos interconectados que se utilizan para compartir recursos, como información, datos, archivos y servicios, entre diferentes computadoras. Estas computadoras, también llamadas nodos de la red, están conectadas mediante cables físicos, como cables Ethernet, o de forma inalámbrica a través de señales de radio o infrarrojos.

Las redes de computadoras permiten la comunicación y el intercambio de información entre los dispositivos conectados, lo que facilita la colaboración, el acceso compartido a recursos y la transferencia de datos. Además, las redes de computadoras pueden tener diferentes topologías, como en forma de estrella, bus, anillo o malla, que determinan la forma en que los nodos están conectados entre sí.



Imagen: Ejemplo de una de red computadoras
Fuente: [Red de Computadoras - Bing images](#)

1.1.2 Telemática

Es una disciplina que combina las tecnologías de las telecomunicaciones y la informática para permitir la transmisión, recepción y procesamiento de información a distancia. El término "telemática" proviene de la combinación de las palabras "telecomunicaciones" e "informática".

La telemática se ocupa del estudio, diseño, desarrollo y aplicación de sistemas y servicios que integran la transmisión de datos, voz y video, junto con el procesamiento y almacenamiento de información.



Imagen: Ejemplo de una de red tematica
Fuente: [telematicas - Bing images](#)

1.1.3 Ventajas de las redes

- Compartir recursos: Permite compartir archivos, impresoras, escáneres y conexiones a internet, lo que mejora la eficiencia y la colaboración.
- Comunicación eficiente: Facilita la comunicación rápida y efectiva entre usuarios, mediante herramientas como correo electrónico, mensajería instantánea y videoconferencias.
- Centralización y administración simplificada: Permite gestionar usuarios, permisos, copias de seguridad y actualizaciones de manera centralizada desde un servidor o punto de control.
- Compartir información: Facilita el intercambio de archivos, documentos y bases de datos de forma segura y rápida, fomentando la colaboración y el intercambio de conocimientos.
- Ahorro de costos: Compartir recursos físicos reduce costos al evitar la duplicación de equipos o servicios.

1.1.4 Modelo de sistema de comunicación

Son representaciones conceptuales que describen los componentes y procesos involucrados en la comunicación de datos en una red. Algunos de los modelos comunes incluyen el modelo OSI, el modelo TCP/IP y el modelo de capas de red de Cisco.

Estos modelos proporcionan una estructura lógica para comprender y gestionar las comunicaciones en una red de computadoras, facilitando el diseño de redes eficientes y la resolución de problemas.

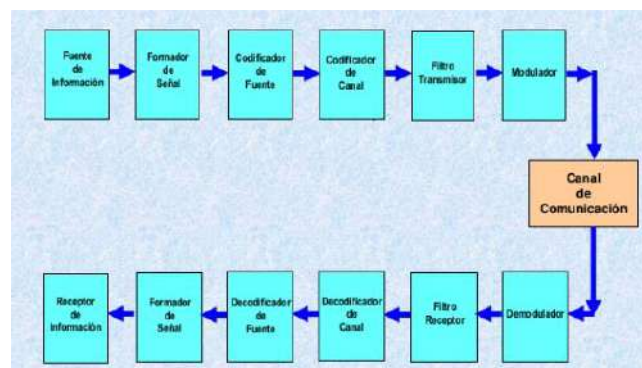


Imagen: Ejemplo de un Modelo de sistema de comunicación
Fuente: [Modelo de sistema de comunicación en programación - Bing images](#)

1.1 REDES

Por: Eddy Guerrero

1.8 Fuentes

la fuente se refiere a la entidad o dispositivo que genera y proporciona los datos o la información para su distribución a través de la red.

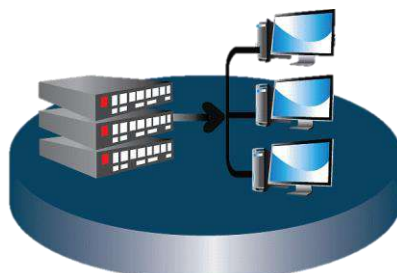


Imagen: Ejemplo del fuente de una red

Fuente: <https://w7.pngwing.com/pngs/110/927/png-transparent-computer-network-service-font-distance-routing-computer-network-service-computer.png>

1.9 Transmisor

Se refiere al dispositivo o componente que se encarga de enviar los datos o la información a través de la red. El transmisor convierte los datos en señales o paquetes de datos adecuados para su transmisión

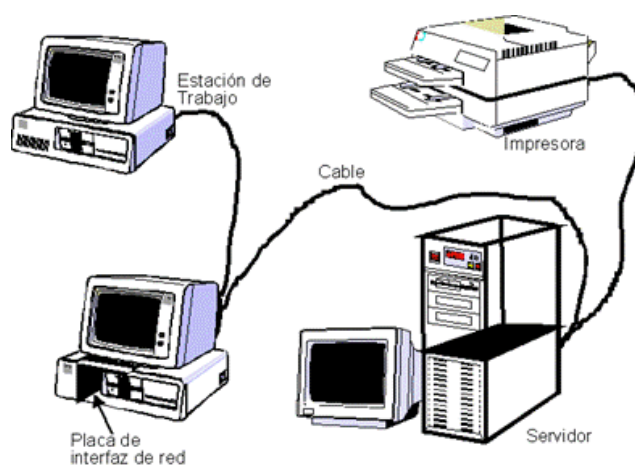


Imagen: Ejemplo transmisor

Fuente: <https://e97817d9aa.cboul-cdnwnd.com/3f8dadb8554eece00e2ee2119d295a03/20000024-88728896e1/red.gif>

1.10 Clasificación de las redes según su tamaño y extensión:

Las redes informáticas pueden clasificarse según su tamaño y extensión en tres categorías principales:

- Red de área personal (PAN)
- Red de área local (LAN)
- Red de área local inalámbrica (WLAN)
- Red de área del campus (CAN)
- Red de área metropolitana (MAN)
- Red de área amplia (WAN)
- Red de área de almacenamiento (SAN)
- Red de área local óptica pasiva (POLAN)

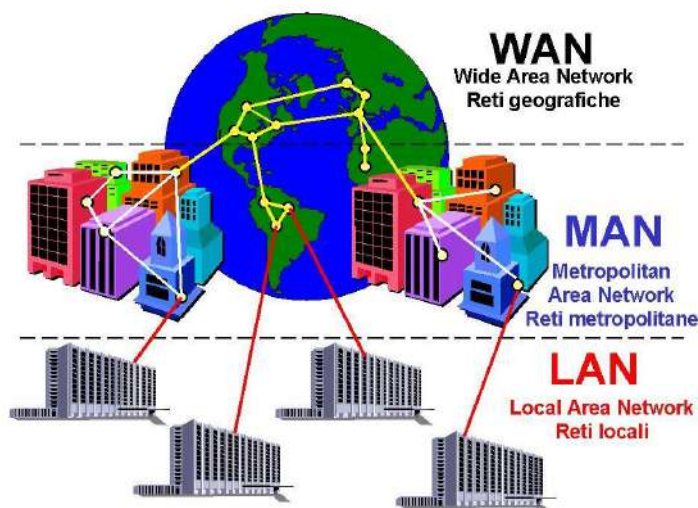
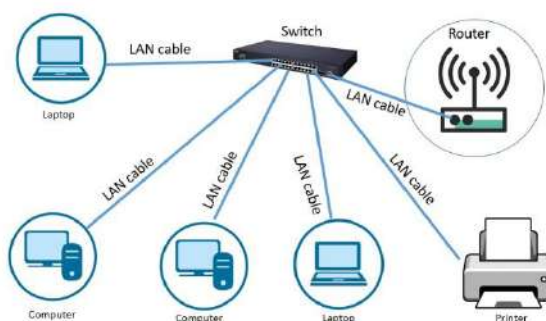


Imagen: Ejemplo de clasificación de las redes

Fuente: <https://www.google.com/search?q=imagen+ejemplo>

1.10.1 LAN

- LAN (Local Area Network) es una clasificación de redes informáticas que se refiere a una red de área local y son utilizadas para interconectar dispositivos cercanos entre sí, como computadoras, impresoras, servidores con el objetivo de compartir recursos,



Local Area Network

Imagen: Ejemplo de Red LAN

Fuente: [https://assets-global.website-](https://assets-global.website-files.com/620d42e86cb8ec4d0839e59d/6230ee03762e4e5318c33297_61ca0988bf85585f5ccc26b5_Local-Area-Network-Diagram.jpeg)

[files.com/620d42e86cb8ec4d0839e59d/6230ee03762e4e5318c33297_61ca0988bf85585f5ccc26b5_Local-Area-Network-Diagram.jpeg](https://assets-global.website-files.com/620d42e86cb8ec4d0839e59d/6230ee03762e4e5318c33297_61ca0988bf85585f5ccc26b5_Local-Area-Network-Diagram.jpeg)

1.10.2. PAN

Por: Ricardo Alberto

Una red de área personal (PAN) conecta dispositivos electrónicos dentro del área inmediata de un usuario. El tamaño de una PAN oscila entre unos centímetros y unos metros.



Imagen: Ejemplo de una red de pan
 Fuente: [que es pan en redes inalambricas - Bing images](#)

1.10.4. WAN

La red inalámbrica WAN, o redes de área amplia, enlazan dispositivos sobre grandes distancias de tierra. Las compañías de teléfonos celulares utilizan estas redes para permitir el acceso a Internet a sus clientes, a menudo bajo el término "banda ancha móvil".

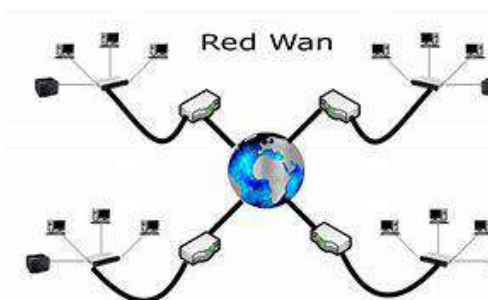


Imagen: Ejemplo de una red de Wan
 Fuente: [que es wan en redes inalambricas - Bing images](#)

1.10.5. WLAN

es una red de computadoras que utiliza tecnología inalámbrica para conectarse y comunicarse entre sí. En una red WLAN, los dispositivos se conectan a un punto de acceso inalámbrico



Imagen: Ejemplo de una red de Wlan
 Fuente: [que es wlan en redes inalambricas - Bing images](#)

1.10.6 VLAN

Por: José Santiago Herrera Furlán

Una VLAN (**Virtual Local Area Network**) es una red lógica que se crea dentro de una red física, permitiendo agrupar dispositivos en segmentos lógicos independientes, aunque estén conectados a la misma infraestructura física. En otras palabras, una VLAN divide una red en múltiples subredes virtuales, lo que permite que los dispositivos se comuniquen como si estuvieran conectados a switches físicos separados.

Las VLAN se crean y administran a nivel de switch. Cada puerto del switch puede ser asignado a una VLAN específica, y los dispositivos conectados a esos puertos formarán parte de esa VLAN. Aunque estos dispositivos pueden estar físicamente conectados al mismo switch, se comportarán como si estuvieran en una red independiente.

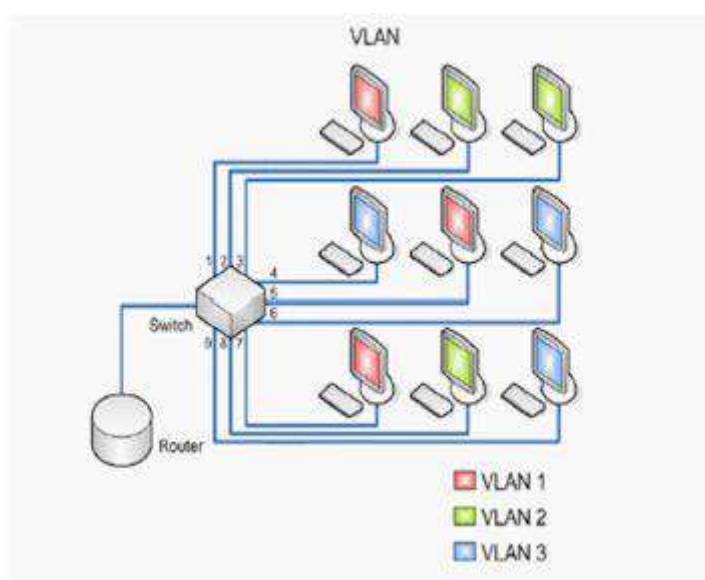


Imagen: Estructura de varias redes VLAN creadas en una misma red física
Fuente: https://www.oas.org/en/citel/infocitel/2008/junio/inictel_e.asp

La principal ventaja de utilizar VLAN es la capacidad de segmentar una red en diferentes grupos lógicos. Esto ofrece varios beneficios:

1. **Mejora de rendimiento:** Al segmentar la red, se reduce el tráfico de difusión y colisión, lo que puede mejorar el rendimiento general y la velocidad de la red.
2. **Aislamiento de tráfico:** Cada VLAN puede tener políticas de seguridad y priorización de tráfico independientes. Esto permite aislar y controlar el flujo de datos entre grupos de dispositivos, mejorando la seguridad y evitando interferencias entre aplicaciones o usuarios.
3. **Flexibilidad y escalabilidad:** Las VLAN permiten cambios y reorganizaciones en la red sin necesidad de modificar la infraestructura física. Se pueden crear, modificar o eliminar VLAN fácilmente según las necesidades de la organización.
4. **Control de acceso:** Las VLAN facilitan la aplicación de políticas de seguridad y control de acceso a nivel de red. Se pueden configurar reglas para permitir o denegar la comunicación entre VLAN, lo que ayuda a proteger los recursos de red.

1.10.7 CAN

Por: José Santiago Herrera Furlán

Una red CAN (**Campus Area Network**) se refiere a una red de área local extendida que abarca un campus universitario o corporativo. Esta red conecta múltiples edificios o ubicaciones dentro de un área geográfica limitada, como un campus universitario, un complejo empresarial o una instalación industrial.

Una red CAN se utiliza para permitir la comunicación y el intercambio de información entre diferentes departamentos, oficinas o unidades dentro de un campus. Permite la conexión de computadoras, servidores, impresoras, dispositivos de almacenamiento y otros recursos compartidos, lo que facilita la colaboración y el acceso a los recursos en toda la organización.

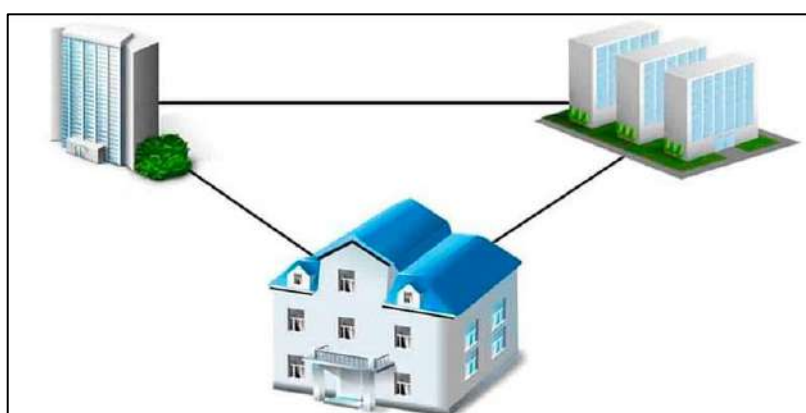


Imagen: Estructura de una red CAN

Fuente: <https://internetpasoapaso.com/red-area-campus/>

Algunas características y componentes comunes de una red CAN pueden incluir:

1. **Cableado estructurado:** Se utiliza un sistema de cableado bien definido para conectar los diferentes edificios o ubicaciones dentro del campus. Esto puede implicar el uso de cables de cobre, fibra óptica u otros medios de transmisión de datos.
2. **Conmutadores y enrutadores:** Estos dispositivos se utilizan para gestionar el tráfico de datos dentro de la red y permitir la conectividad entre diferentes subredes o VLAN (Virtual LAN).
3. **Servidores y servicios de red:** Se pueden implementar servidores en la red CAN para proporcionar servicios compartidos, como servidores de archivos, servidores de correo electrónico, servicios de directorio, sistemas de seguridad, etc.
4. **Puntos de acceso inalámbricos:** En muchos casos, las redes CAN también ofrecen conectividad inalámbrica para permitir la movilidad y el acceso a la red desde diferentes áreas del campus.
5. **Seguridad de red:** La seguridad es una consideración importante en una red CAN. Se utilizan medidas como firewalls, sistemas de detección de intrusiones, autenticación de usuarios y encriptación para proteger los datos y los recursos de la red.

1.10.8 SAN

Por: José Santiago Herrera Furlán

Una red SAN (**Storage Area Network**) es una arquitectura de red especializada diseñada para conectar y compartir dispositivos de almacenamiento de datos, como discos duros, matrices de almacenamiento y bibliotecas de cintas, con servidores y sistemas informáticos.

La principal función de una red SAN es permitir un acceso rápido y confiable a los datos almacenados, independientemente de la ubicación física de los dispositivos de almacenamiento. A diferencia de las redes de área local (LAN) tradicionales, que se centran en la comunicación entre dispositivos de cómputo, una red SAN se enfoca en la transferencia eficiente de datos entre los servidores y los sistemas de almacenamiento.

Las redes SAN utilizan tecnologías de conectividad de alto rendimiento, como *Fibre Channel* (FC) y más recientemente *iSCSI* (*Internet Small Computer System Interface*), para establecer enlaces dedicados y de alta velocidad entre los servidores y los dispositivos de almacenamiento. Estos enlaces pueden atravesar cables de fibra óptica o utilizar redes IP existentes para la transmisión de datos.

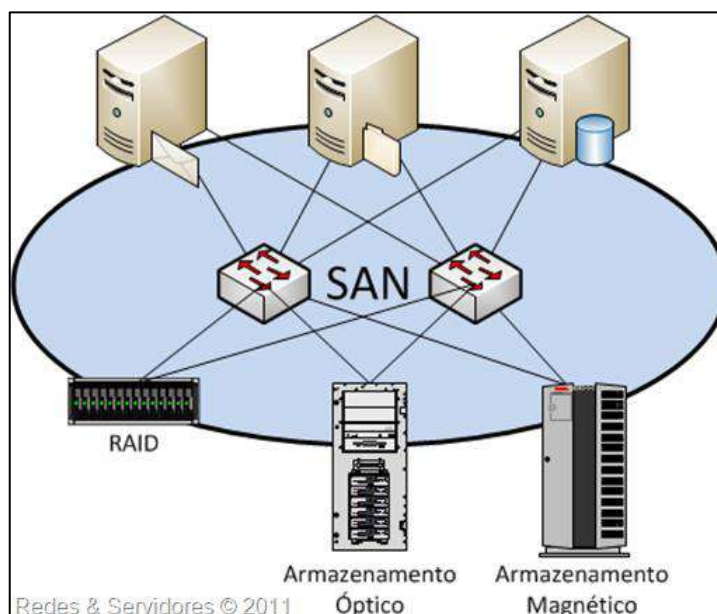


Imagen: Red SAN con comunicación entre sistemas de almacenamiento y computadoras

Fuente: <https://www.pinterest.com.mx/pin/720716746599061500/>

Algunas características y ventajas clave de las redes SAN son:

1. **Rendimiento:** Las redes SAN están diseñadas para ofrecer un alto rendimiento y baja latencia, lo que permite una transferencia rápida de grandes volúmenes de datos.
2. **Escalabilidad:** Las redes SAN son altamente escalables, lo que significa que pueden crecer y adaptarse fácilmente a medida que se añaden más servidores y dispositivos de almacenamiento a la red.
3. **Consolidación:** Las redes SAN permiten la consolidación del almacenamiento, lo que significa que múltiples servidores pueden acceder y compartir los mismos dispositivos de almacenamiento. Esto simplifica la administración y reduce los costos al eliminar la necesidad de almacenamiento local en cada servidor.

4. **Alta disponibilidad:** Las redes SAN ofrecen capacidades de alta disponibilidad, lo que significa que se pueden implementar configuraciones redundantes para garantizar la continuidad del acceso a los datos en caso de fallas en los componentes de la red.
5. **Administración centralizada:** Las redes SAN facilitan la administración centralizada del almacenamiento, lo que simplifica las tareas de administración, como la asignación de recursos, la realización de copias de seguridad y la recuperación de datos.

1.10.9 Redes Internet

Por: José Santiago Herrera Furlán

Una "red de Internet" se refiere a la infraestructura global que interconecta millones de dispositivos informáticos y redes en todo el mundo, permitiendo la comunicación y el intercambio de información a través de protocolos de Internet estándar. Es la red de redes más grande y es ampliamente utilizada en el mundo.

Las redes de Internet se basan en una arquitectura descentralizada y utilizan el protocolo de Internet (IP) como estándar para el enrutamiento y la transmisión de datos. Está compuesta por una amplia variedad de tecnologías de red, incluyendo cables de fibra óptica, líneas telefónicas, enlaces de satélite, redes inalámbricas y otros medios de comunicación.

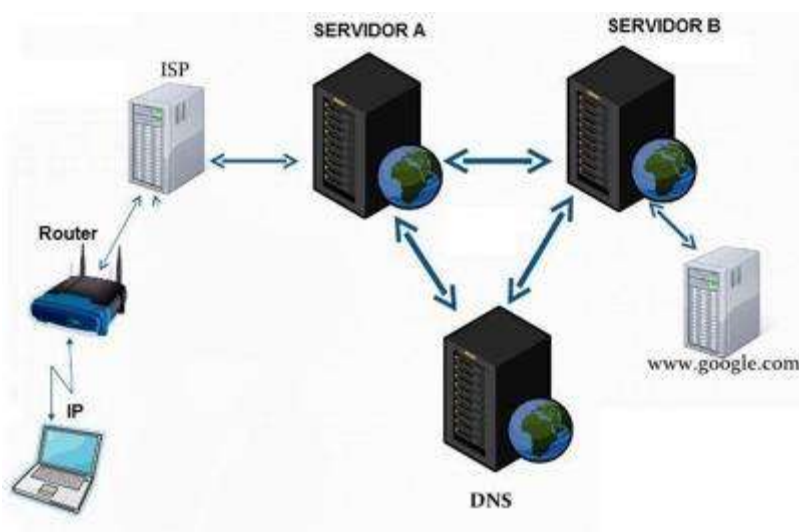


Imagen: Estructura de enrutamiento en Internet

Fuente: <https://pc-solucion.es/unidad/3-introduccion-a-internet/>

El funcionamiento de las redes de Internet se basa en el intercambio de datos en paquetes, donde la información se divide en pequeños fragmentos para ser transmitida de forma eficiente a través de la red. Los routers y conmutadores de red se encargan de dirigir los paquetes de datos hacia su destino final a través de rutas óptimas.

La red de Internet permite una amplia gama de servicios y aplicaciones, como el correo electrónico, la transferencia de archivos, la navegación web, las redes sociales, el streaming de video, las llamadas telefónicas por Internet (VoIP) y muchas otras formas de comunicación y colaboración en línea.

1 . 1 0. 10. Redes inalámbricas

Por: Jose Jordan

Las redes inalámbricas son sistemas de comunicación que permiten la transmisión de datos sin la necesidad de cables físicos. Utilizan tecnologías como Wi-Fi, Bluetooth, infrarrojo o satélite para establecer conexiones entre dispositivos electrónicos. Estas redes son ampliamente utilizadas en entornos domésticos, empresariales y públicos, ya que ofrecen movilidad y flexibilidad al permitir la conexión de dispositivos sin necesidad de cables. Además, proporcionan acceso a internet, permiten la transferencia de datos y la comunicación entre dispositivos de manera inalámbrica.



Imagen: son aquellas que la información viaja por el aire, en lugar de por cable.

Fuente: https://sp-ao.shortpixel.ai/client/to_webp,q_glossy,ret_img,w_560/https://www.redesinalambricas.es/wp-content/uploads/2019/03/redes-inalambricas.jpg

1.10.11. Topologías

Las topologías de red son los diferentes diseños físicos y lógicos que se utilizan para interconectar dispositivos en una red de computadoras.

Topología de bus: Todos los dispositivos están conectados a un único cable principal (bus).

Topología de estrella: Todos los dispositivos están conectados a un nodo central, como un switch o un concentrador.

Topología de anillo: Los dispositivos se conectan en un círculo cerrado formando un anillo.

Topología de malla: Cada dispositivo está conectado directamente a todos los demás dispositivos en la red.

Topología de árbol: Los dispositivos están organizados en una estructura jerárquica similar a un árbol.

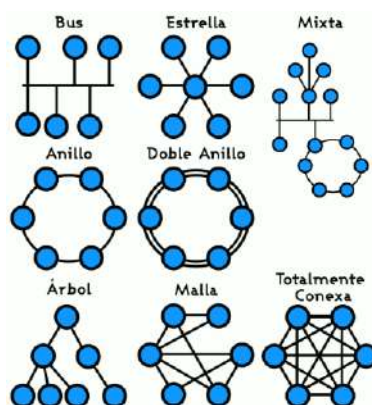


Imagen: forma en la que se organizan los elementos que conforman una red de comunicaciones

Fuente https://upload.wikimedia.org/wikipedia/commons/4/4a/Topolog%C3%ADa_de_red.png

2. Modelo Osi y protocolos TCP/IP

El modelo OSI (Open Systems Interconnection) es un marco de referencia que define una serie de siete capas que describen cómo los sistemas de comunicación deben interactuar entre sí. Estas capas son: física, enlace de datos, red, transporte, sesión, presentación y aplicación. Cada capa se encarga de tareas específicas y se comunica con las capas adyacentes para proporcionar servicios de red.

Los protocolos TCP/IP (Transmission Control Protocol/Internet Protocol) son el conjunto de reglas que permiten la comunicación en Internet. TCP es responsable de la entrega confiable de datos, dividiéndolos en paquetes y asegurando que lleguen en orden y sin errores. IP, por su parte, se encarga del enrutamiento y direccionamiento de los paquetes en la red.

2.1 Capas del modelo OSI

El modelo OSI (Open Systems Interconnection) es un marco de referencia que describe las funciones de comunicación de un sistema de red en capas. Estas capas proporcionan un enfoque estructurado para el diseño, implementación y gestión de redes de computadoras.

Capa física: Esta capa se encarga de la transmisión de bits a través del medio físico, como cables o señales inalámbricas.

Capa de enlace de datos: Se encarga de la detección y corrección de errores en los bits transmitidos.

Capa de red: Aquí se establecen las rutas y se controla el enrutamiento de los datos a través de la red.

Capa de transporte: Proporciona un transporte confiable de extremo a extremo de los datos.

Capa de sesión: Permite establecer, mantener y terminar sesiones de comunicación entre aplicaciones en diferentes dispositivos.

Capa de presentación: Se ocupa de la representación y el formato de los datos para que sean comprensibles por las aplicaciones.

Capa de aplicación: Es la capa más alta del modelo OSI y proporciona servicios de red a las aplicaciones.

1.1 Capa física

Por: Angel Enrique Juarez Asencio

La capa física es la capa más baja del modelo OSI y se encarga de la transmisión física de los datos a través del medio de comunicación, como cables o señales inalámbricas.

Medios de transmisión:

Los diferentes tipos de medios de transmisión utilizados en redes, como cables de cobre, fibra óptica y transmisión inalámbrica. Puedes explorar las características, ventajas y desventajas de cada tipo de medio.

Técnicas de modulación:

La modulación es el proceso de codificar la información en una señal para transmitirla a través del medio de transmisión. Puedes aprender sobre técnicas de modulación como AM (Amplitude Modulation), FM (Frequency Modulation) y QAM (Quadrature Amplitude Modulation).

Estándares de cableado:

Los estándares como el cableado estructurado y los diferentes tipos de cables, como UTP (Unshielded Twisted Pair) y STP (Shielded Twisted Pair), utilizados para conectar dispositivos en redes de área local (LAN).

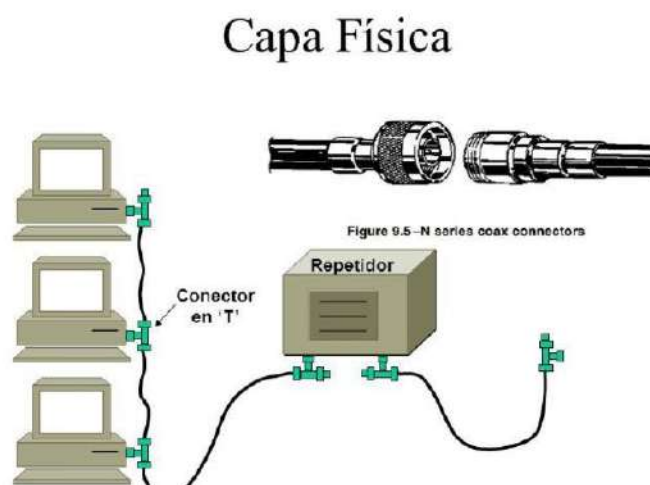


Imagen: computadoras conectadas

Fuente: <https://es.scribd.com/presentation/352897073/Capa-Fisica-10-A>

1.1.1 Capa de enlace de datos

La capa de enlace de datos es responsable de la transferencia confiable de datos entre nodos adyacentes en una red y proporciona control de flujo y detección y corrección de errores.

Ethernet:

Un protocolo de capa de enlace de datos muy común utilizado en redes LAN. Puedes explorar sus diferentes estándares, como Ethernet 802.3, y aprender sobre la estructura del marco Ethernet y los mecanismos de control de acceso al medio.

Switching:

Los switches son dispositivos utilizados en la capa de enlace de datos para conectar múltiples dispositivos en una red y permitir la comunicación eficiente. Puedes aprender sobre los diferentes métodos de switching, como switching de circuitos, de paquetes y de etiquetas.



Imagen: conexión a servidores

Fuente: https://www.google.com/url?sa=i&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FProgramaci%25C3%25B3n_por_capas&psig=AOvVaw1yFo6Q2VJ4rmlTumX7w5pV&ust=1685768293162000&source=images&cd=vfe&ved=OCBEQjRxqFwoTCNiIr7Tmo_8CFQAAAAAdAAAAABAI

1.1.2 Capa de red

La capa de red se encarga del enrutamiento de los paquetes de datos a través de una red, seleccionando las mejores rutas y proporcionando la conectividad lógica entre diferentes redes.

Enrutamiento:

El proceso de determinar la mejor ruta para enviar paquetes de datos desde una fuente hasta un destino a través de una red. Puedes explorar los protocolos de enrutamiento, como RIP, OSPF y BGP, y aprender sobre algoritmos de enrutamiento como Dijkstra y Bellman-Ford.

Protocolos de Internet:

Los protocolos utilizados en la capa de red, como IPv4 e IPv6. Puedes estudiar el formato de los encabezados IP, la estructura de las direcciones IP y los conceptos de enrutamiento en internet.

1.1.3 Capa de transporte

La capa de transporte es responsable de la entrega confiable y ordenada de los datos entre aplicaciones en dispositivos finales. También proporciona mecanismos de control de

flujo y de manejo de errores.

Protocolo de Control de Transmisión (TCP):

Un protocolo de transporte confiable y orientado a la conexión que se utiliza ampliamente en aplicaciones que requieren una entrega precisa y ordenada de datos.

Protocolo de Datagrama de Usuario (UDP):

Un protocolo de transporte no confiable y sin conexión que se utiliza para aplicaciones que requieren una transmisión rápida y eficiente, como streaming de video y juegos en línea.

Control de flujo y control de congestión:

Mecanismos utilizados en la capa de transporte para regular el flujo de datos y evitar la congestión en la red. Puedes explorar técnicas como Ventana Deslizante y Algoritmos de Control de Congestión, como TCP Reno.

Protocolos de Internet: Los protocolos utilizados en la capa de red, como IPv4 e IPv6. Puedes estudiar el formato de los encabezados IP, la estructura de las direcciones IP y los conceptos de enrutamiento en internet.

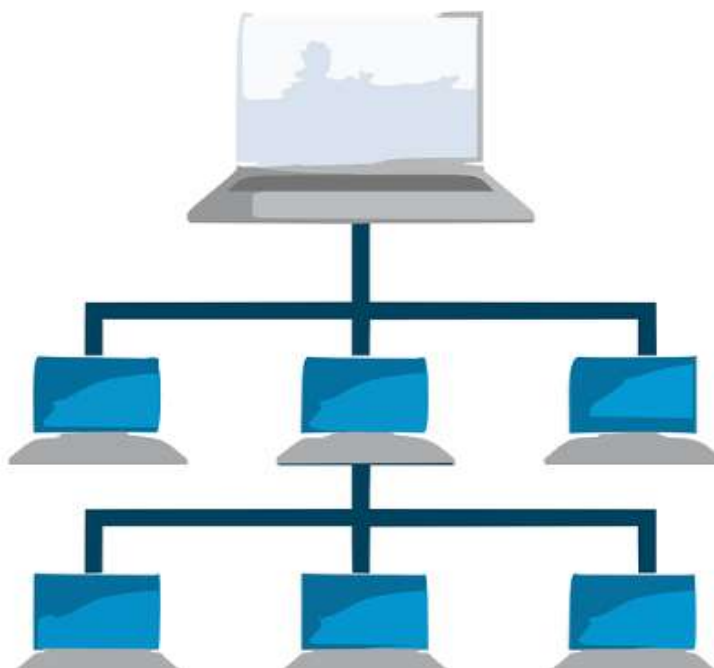


Imagen: varias computadoras conectadas

Fuente: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.welivesecurity.com%2Fes%2F2015%2F06%2F01%2Fcomo-fortalecer-capas-redes-informaticas%2F&psig=A0vVaw1ZAGljRBIhOOgs4dDFPD0z&ust=1685768350199000&source=images&cd=vfe&ved=OCBEQjRxqFwoTCJDzyc_mo_8CFQAAAAAdAAAAABAE

1.1 Capa 1: La capa física

Por: Gustavo Juarez

La capa física se encarga de establecer y mantener la conexión física entre los dispositivos de la red, definiendo las características eléctricas, mecánicas y funcionales de los cables, conectores y otros elementos físicos utilizados para la transmisión de datos. La codificación de datos es el proceso de convertir la información en un formato específico para su almacenamiento, procesamiento o transmisión. En el contexto de la programación, la codificación de datos se refiere a la representación de datos en un formato adecuado para su manipulación en un programa

1.1.1 Capa 2: Codificación de datos

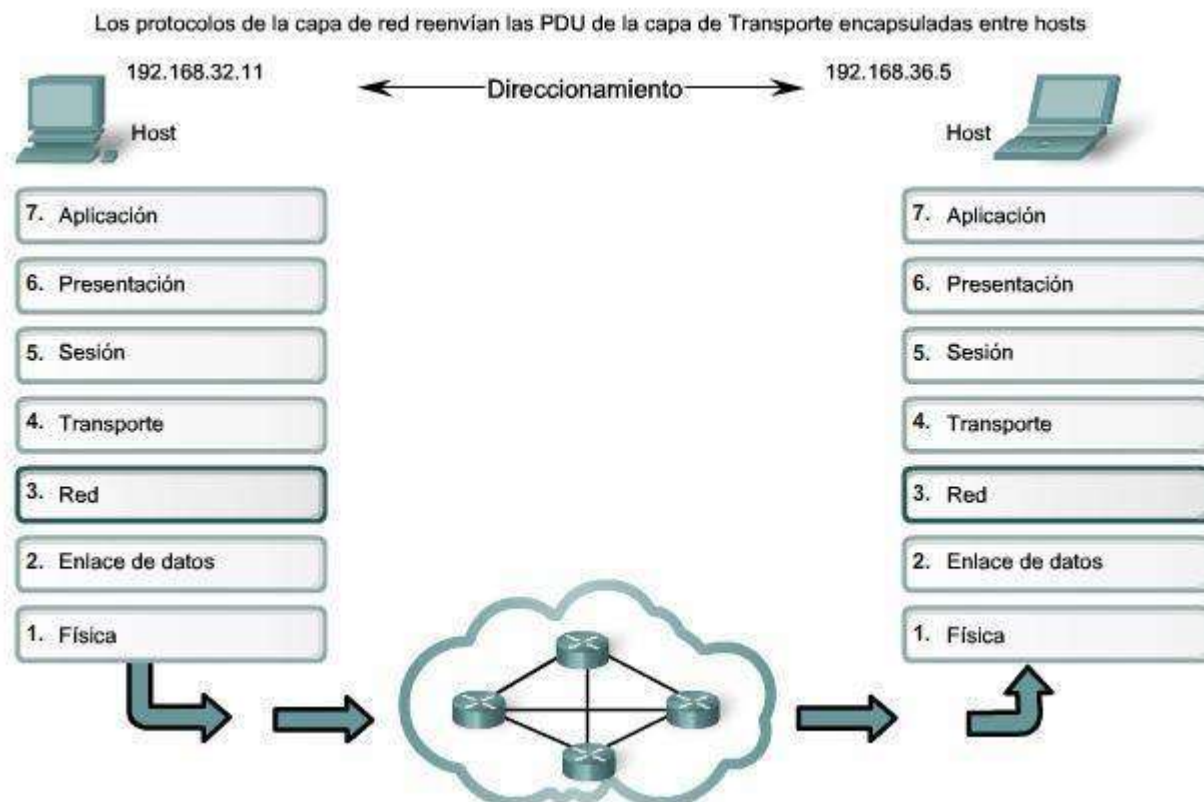
La capa física se encarga de convertir los datos digitales en señales eléctricas, ópticas o electromagnéticas adecuadas para su transmisión a través del medio físico. Esto implica la conversión de bits en señales físicas, como la modulación y demodulación de la señal.

1.1.2 Transmisión de bits

La capa física se ocupa del envío de bits individuales a través del medio de transmisión. Esto puede implicar la división de los datos en tramas más pequeñas para su transmisión y la sincronización de los dispositivos para asegurar una transmisión correcta.

1.1.3 Capa 3: La capa de red.

La capa de red, también conocida como Capa 3, es una de las capas del modelo OSI (Interconexión de Sistemas Abiertos) y del modelo TCP/IP, que son dos modelos de referencia utilizados en redes de computadoras para describir cómo se realiza la comunicación entre sistemas. La capa de red se encarga de proporcionar servicios de enrutamiento y conmutación de datos para permitir que los paquetes de datos lleguen desde el origen hasta el destino a través de una red.



Fuente:

<https://sites.google.com/site/informaticaredesdecomputadoras/unidad-3-capas-inferiores-del-modelo-osi-y-tcp-ip/3-1-capa-de-red>

1.1.4 Capa 4: La capa de transporte.

La capa de transporte, también conocida como Capa 4 del modelo OSI (Open Systems Interconnection), es una capa fundamental en el ámbito de las redes de computadoras. Su principal función es garantizar una entrega confiable y eficiente de los datos entre hosts o dispositivos finales.

1.1.5 Capa 5: La capa de transporte.

La Capa 5 del modelo OSI (Interconexión de Sistemas Abiertos) es la capa de sesión. Esta capa se encarga de establecer, mantener y terminar las sesiones de comunicación entre dos nodos de una red. Su principal objetivo es garantizar que las aplicaciones en ambos extremos de la comunicación tengan una interacción coherente y confiable.

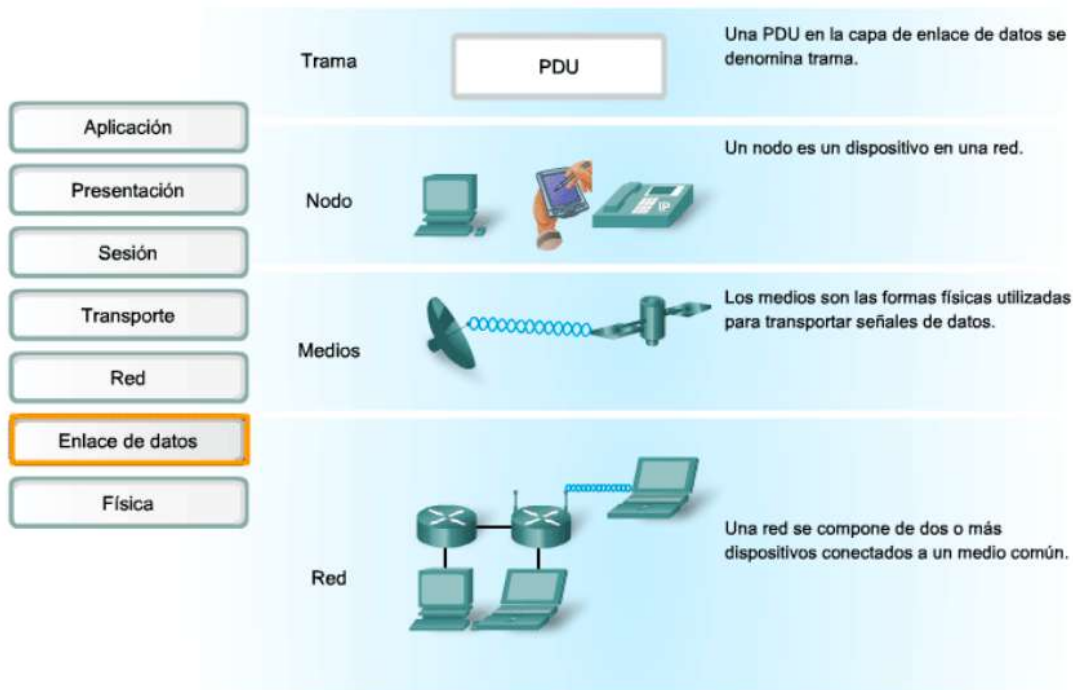


Imagen: Ejemplo del tamaño solicitado (información)

Fuente: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro35/221_acceso_al_medio.html

1.1 Modelo OSI

Por: **María Emilia Larios Contreras**

El modelo OSI, que significa Open Systems Interconnection (Interconexión de Sistemas Abiertos), es un marco conceptual desarrollado por la Organización Internacional de Normalización (ISO) en la década de 1980. Este modelo se utiliza para describir cómo los sistemas de comunicación de computadoras interactúan entre sí. Consiste en siete capas distintas que desempeñan funciones específicas en el proceso de comunicación.

1.1.1 Capa 6: La capa de presentación

La capa de presentación es una de las capas del modelo OSI y es responsable de la representación y el formato de los datos para que puedan ser entendidos por las aplicaciones receptoras. Su objetivo principal es proporcionar una presentación coherente e independiente de la forma en que los datos se almacenan, transmiten y representan internamente en los sistemas.

Un ejemplo común de la función de la capa de presentación es la compresión de datos. La capa de presentación puede comprimir los datos antes de enviarlos a través de la red y descomprimirlos en el extremo receptor. Esto puede ahorrar ancho de banda y acelerar la transferencia de datos.

Además, la capa de presentación puede convertir los datos en un formato comúnmente aceptado para su intercambio entre diferentes sistemas. Por ejemplo, puede convertir datos en un formato de texto o en un formato de imagen estándar para que puedan ser interpretados y mostrados correctamente por las aplicaciones receptoras.



Imagen: Ejemplo de capa de aplicación

Fuente: <https://acis.org.co/portal/sites/default/files/Captura%20de%20pantalla%202018-07-11%20a%20la%28s%29%202.53.57%20p.%20m..png>

1.1.2 Capa 7: La capa de aplicación

La capa de aplicación es la capa más alta del modelo OSI y es la capa más cercana al usuario final. Esta capa proporciona interfaces y servicios para que las aplicaciones se comuniquen a través de la red. Es la capa en la que los usuarios interactúan directamente con las aplicaciones y los servicios de red.

La capa de aplicación incluye una variedad de protocolos y servicios que permiten a los usuarios acceder a recursos de red y realizar tareas específicas.

Esta capa también se encarga de funciones como el manejo de autenticación y seguridad, el manejo de sesiones de usuario, la gestión de solicitudes y respuestas, y la traducción de datos a formatos comprensibles para las aplicaciones.



Imagen: Aplicación móvil

Fuente: <https://blogs.unsw.edu.au/nowideas/files/2017/06/aplicacion-movil.jpg>

1.1.3 Funciones de cada capa

Capa física: Esta capa se encarga de la transmisión física de bits a través de medios de comunicación como cables, fibra óptica o señales inalámbricas. Define las características eléctricas, mecánicas y funcionales de los medios físicos.

Capa de enlace de datos: Su función principal es proporcionar una comunicación confiable y libre de errores entre dos nodos de red adyacentes. Se encarga de la detección y corrección de errores, el control de flujo y el acceso al medio compartido.

Capa de red: La capa de red se ocupa del enrutamiento de datos y de la entrega de paquetes desde el origen hasta el destino a través de una o varias redes intermedias. Su función es establecer rutas óptimas, gestionar direcciones IP y determinar el mejor camino para el envío de datos.

Capa de transporte: Esta capa proporciona servicios de transporte confiables y transparentes a través de la red. Se encarga de dividir los datos en segmentos, establecer conexiones y realizar el control de flujo y la corrección de errores.

Capa de sesión: Su función principal es establecer, mantener y finalizar las sesiones de comunicación entre aplicaciones. Proporciona mecanismos para el control de diálogo y la sincronización de la comunicación.

Capa de presentación: La capa de presentación se encarga de la representación y el formato de los datos para garantizar su comprensión entre las aplicaciones. Realiza tareas como la compresión, el cifrado, la conversión de formatos y la sintaxis de los datos.

Capa de aplicación: Es la capa más cercana al usuario final y proporciona interfaces para que las aplicaciones se comuniquen a través de la red. Incluye protocolos como HTTP, FTP, SMTP, DNS, entre otros, que permiten el acceso a servicios y recursos de red.

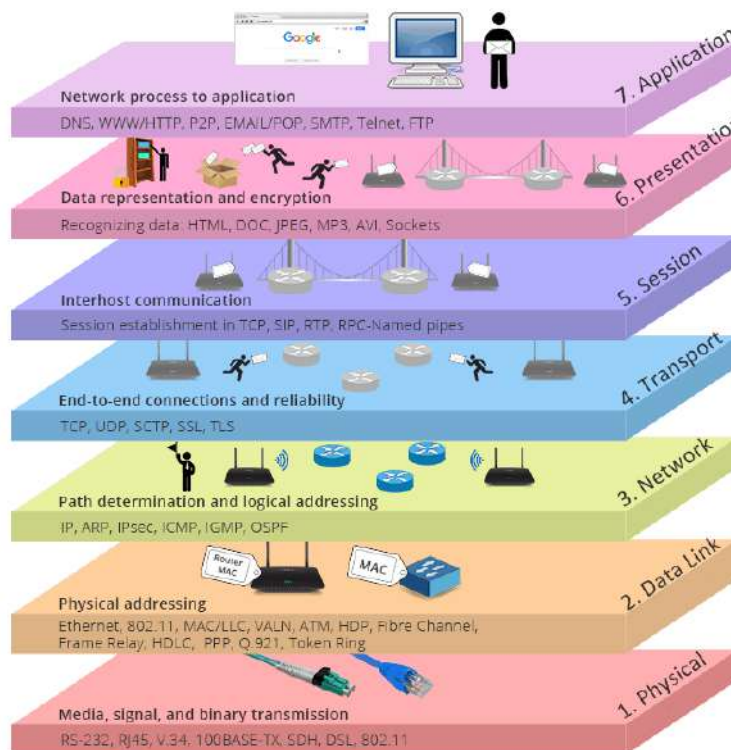


Imagen: Modelo OSI, capas

Fuente: <https://media.fs.com/images/community/upload/kindEditor/202106/08/siete-capas-de-modelo-osi-1623136242-cywGEI9RkM.png>

1.1.4 Protocolos de cada capa

Capa física: Ethernet, RS-232, Wi-Fi, Bluetooth.

Capa de enlace de datos: Ethernet, IEEE 802.11 (Wi-Fi), HDLC (High-Level Data Link Control), PPP (Point-to-Point Protocol).

Capa de red: IP (Internet Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol), OSPF (Open Shortest Path First).

Capa de transporte: TCP (Transmission Control Protocol), UDP (User Datagram Protocol), SCTP (Stream Control Transmission Protocol).

Capa de sesión: NetBIOS (Network Basic Input/Output System), SIP (Session Initiation Protocol), NFS (Network File System).

Capa de presentación: MIME (Multipurpose Internet Mail Extensions), SSL/TLS (Secure Sockets Layer/Transport Layer Security), ASCII (American Standard Code for Information Interchange).

Capa de aplicación: HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), DNS (Domain Name System), SNMP (Simple Network Management Protocol).



Imagen: Modelo OSI, representación de capas

Fuente: https://www.prored.es/wp-content/uploads/2018/11/prored-capas-modelo-osi_v2.png

1.1.5 Dispositivos de redes utilizados en cada capa

Capa física: Los dispositivos de esta capa se encargan de la transmisión física de bits a través de los medios de comunicación. Ejemplos de dispositivos de esta capa son cables Ethernet, repetidores, hubs, convertidores de medios y tarjetas de red.

Capa de enlace de datos: Los dispositivos de esta capa se ocupan de la comunicación confiable entre nodos adyacentes. Ejemplos de dispositivos son switches (conmutadores), bridges (puentes), adaptadores de red y controladores de interfaz de red (NIC).

Capa de red: Los dispositivos de esta capa se encargan del enrutamiento y la entrega de paquetes a través de las redes. Ejemplos de dispositivos son routers (enrutadores), gateways, firewalls y switches de capa 3.

Capa de transporte: Los dispositivos de esta capa proporcionan servicios de transporte confiables y transparentes. Ejemplos de dispositivos son gateways de transporte, firewalls de capa 4 y controladores de flujo.

Capa de sesión: Los dispositivos de esta capa ayudan a establecer, mantener y finalizar sesiones de comunicación. Ejemplos de dispositivos son servidores de autenticación, servidores de sesión y controladores de sesión.

Capa de presentación: Los dispositivos de esta capa se encargan de la representación y el formato de los datos. Ejemplos de dispositivos son gateways de protocolo, servidores de compresión y servidores de conversión de formatos.

Capa de aplicación: Los dispositivos de esta capa brindan servicios y aplicaciones directamente a los usuarios finales. Ejemplos de dispositivos son servidores web, servidores de correo electrónico, servidores DNS y servidores de archivos.

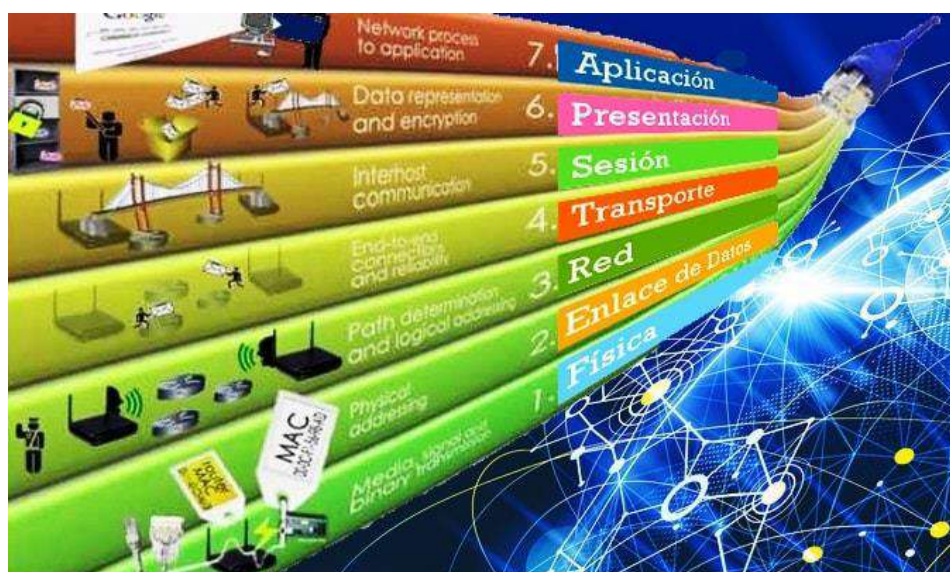


Imagen: Capas y dispositivos del modelo OSI

Fuente: <https://comofriki.com/wp-content/uploads/2019/06/modelo-osi-6-min.jpg>

Por: Glendy Lopez

2.12 Redes IP

Cada dispositivo en una red IP necesita 3 piezas diferentes de información para comunicarse correctamente con otros dispositivos en la red: dirección IP, máscara de subred y dirección de transmisión. Normalmente, cada uno de estos números se escribirá como cuatro "octetos" (por ejemplo, 198.41.12.151, 255.255.255.0 y 198.41.12.255). Cada dirección IP en realidad consta de dos partes: una parte de "red" que le dice al enrutador a qué conjunto de dispositivos debe ir el paquete (por ejemplo, cualquiera, campus, etc.), y una parte de "host" que le dice al enrutador. El paquete debe ser enviado a una unidad específica de este grupo.

Cómo funciona una red IP

Una red permite que una computadora use Internet para comunicarse con otra computadora, recolectando información y enviándola directamente a otra. Cada computadora conectada a la red tiene su propia dirección IP, lo que permite a sus empleados enviar información directamente a la persona que la necesita.

Hay tres tipos de direcciones IP utilizadas en la red:

- Dirección del host: esta dirección es única para esta persona y les permite comunicarse uno a uno y con la red.
- Dirección de red: esta es la primera dirección IP en la red. Esta dirección brinda a los hosts que brindan servicios de red especiales la capacidad de comunicarse con otros.
- Dirección de difusión: este tipo de dirección permite que el anfitrión transmita un anuncio a través de la red.

Cualquier enrutador puede determinar la mejor ruta para reenviar un paquete de un clúster a otro al examinar la dirección de destino en el paquete IP que reenvía y usar la configuración estática o la información recopilada dinámicamente de otros enrutadores. Cada grupo de dispositivos IP de Internet debe tener un recurso compartido de red único, y cada dispositivo de ese grupo también necesita un recurso compartido de host único. En el caso de Internet, esta singularidad es posible gracias a la obtención indirecta de todas las asignaciones de segmentos de red a través de una cámara de compensación central conocida como Network Information Center o "NIC". Las NIC asignan bloques de direcciones a los proveedores de servicios de Internet (ISP), quienes luego asignan estas direcciones a sus clientes.

Clases de direcciones IP

Clase	Porción de red	Anfitriones permitidos
A	de 1,0 a 127,0	aprox. 16 millones
B	de 128.0 a 191.255	65,536
C	de 192.0 a 223.255.255	255

Máscaras de subred

La máscara de subred le dice al enrutador qué parte de la dirección debe considerarse parte de la red.

Clase	Máscara de subred
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

Así como las máscaras anteriores especifican qué porción del rango global de direcciones IP está usando una red, una máscara de subred también se puede usar para subdividir un rango de red Clase A, B o C en múltiples grupos de hosts o "subredes".

Máscara de subred	Rangos de host
255.255.255.0	1 a 254 (C tradicional)
255.255.255.128	1-126, 129-254
255.255.255.192	1-62, 65-126, 129-190, 193-254
255.255.255.224	1-30, 33-62, 65-94, 97-126, 129-158, 161-190, 193-222, 225-254
255.255.255.240	1-14, 17-30, 33-46, 49-62, 65-78, 81-94, 97-110, 113-126, 129-142, 145-158, 161-174, 177-190, 193-206, 209-222, 225-238, 241-254
255.255.255.248	1-6, etc

La dirección calculada más baja en cada rango (0 en el rango C tradicional) no se muestra, no se puede usar y no se muestra en el gráfico. La dirección más alta en cada rango (255 en el rango C tradicional) tampoco se muestra y es la dirección de transmisión para la subred.

En cada máscara anterior, 1 representa la parte de la red en binario y 0 representa la parte del host (128 es 10000000, 192 es 11000000, etc.). Cuantos más bits use para representar la parte de la red, menos bits quedarán para usar como direcciones de host.

Direcciones de difusión

La dirección de difusión es la dirección a la que los dispositivos envían paquetes destinados a todos los demás dispositivos. Todos los dispositivos "escuchan" transmisiones además de su propia dirección.

En la mayoría de los casos, la dirección de transmisión es la última dirección en la red (o subred) y la parte del host consta de direcciones binarias (sin embargo, en algunas redes se usa 0.0.0.0 o 255.255.255.255).

Clase	Red	Máscara de subred	Transmisión
A	45.0.0.0	255.0.0.0	45.255.255.255
B	128.138.0.0	255.255.0.0	128.138.255.255
C	198.41.9.0	255.255.255.0	198.41.9.255
A*	45.21.16.0	255.255.252.0	42.21.19.255
C*	198.41.9.64	255.255.255.224	198.41.9.95

Asignación de una dirección IP

Utilice la parte de red proporcionada por su administrador o ISP. Asigne una parte de host única (es decir, no utilizada) a la interfaz del enrutador. Por ejemplo, si su ISP le dice que su parte de red es 198.41.9, puede asignarle la interfaz 198.41.9.1. Si tiene un enrutador con más de una interfaz, las partes de red (+ subred) de la dirección IP de cada puerto deben ser diferentes.

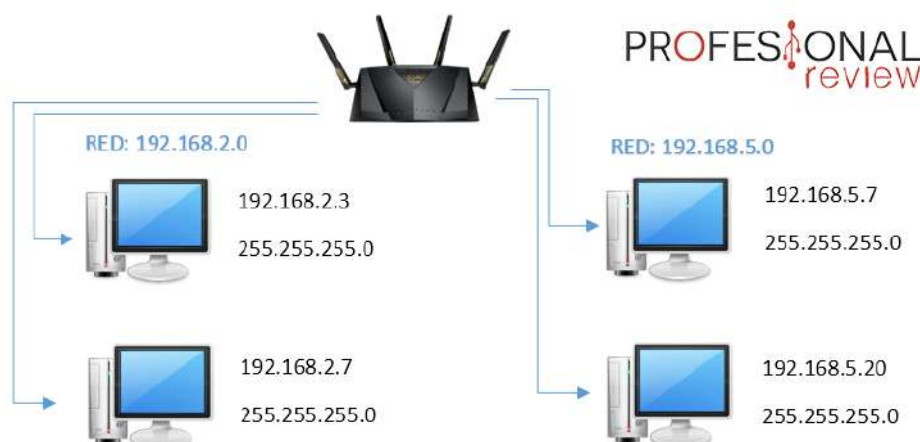


Imagen: Ejemplo de la relación de redes IP

Fuente <https://www.profesionalreview.com/wp-content/uploads/2019/01/Direccionamiento-IP-02.png>

2.13 Protocolo IP versión 4

Discutir IPv4 es esencialmente discutir el Protocolo de Internet en general, ya que esta es (bastante contrariamente a la intuición) la primera versión ampliamente aplicable del protocolo

Protocolo de Internet

Es un protocolo de capa de red (OSI-ISO Layer 3) del conjunto de protocolos de Internet (también conocido como TCP/IP) que se utiliza para transmitir flujos de paquetes y datagramas a través de redes. Proporciona una única forma de direccionamiento para todos (mediante la asignación de direcciones IP a los dispositivos), lo que permite que los usuarios de diferentes redes se comuniquen. Después de todo, la comunicación entre diferentes redes es el principio básico de Internet (de ahí el nombre: inter significa "entre" y rojo es la abreviatura de "red").

En el modelo TCP/IP existen 4 capas, que de una forma u otra corresponden a las capas del modelo OSI. Estas son las capas de canal, Internet, transporte y aplicación (del 1 al 4, respectivamente). Cuando los datos se envían de un dispositivo a otro, se encapsulan en diferentes protocolos descendentes, es decir, en la capa de aplicación (que corresponde aproximadamente a las capas OSI-ISO 7, 6 y 5) se pueden envolver en el protocolo HTTP, luego en el transporte (OSI-ISO capa 4) - en TCP, en la capa de Internet (OSI-ISO capa 3) a IP y luego transportado a través de Ethernet en la capa de enlace de datos (correspondiente a las capas 1 y 2 de OSI-ISO). Cuando se reciben los datos, el proceso se invierte y el mensaje se "desempaqueta".

Enrutamiento

La característica distintiva de cualquier dispositivo es su dirección MAC, pero no es así como los dispositivos se comunican entre redes. Para eso, se les asigna una dirección IP que le dice al enrutador a dónde debe enviarse el paquete de datos. Puede pensar en

ello como un servicio postal: su carta no va directamente a su destinatario; primero se envía de un centro postal a otro para optimizar el proceso de entrega. De manera similar, los paquetes de datos se reenvían de un enrutador a otro hasta que alcanzan la dirección IP correcta.

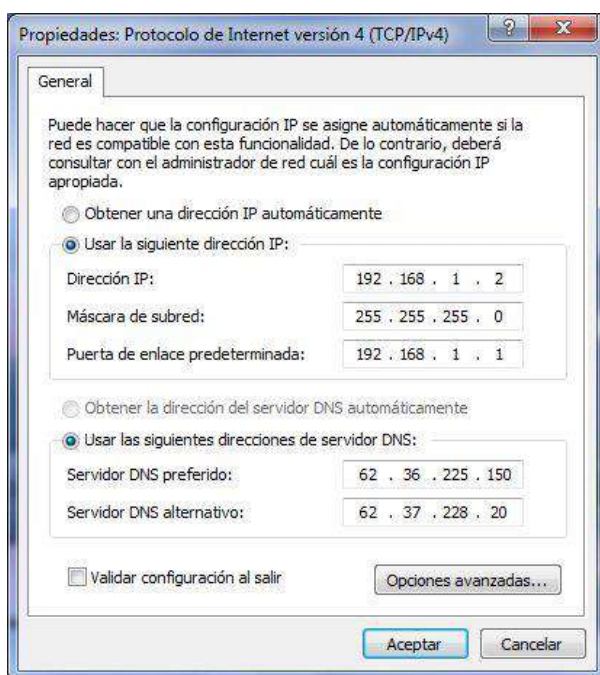


Imagen: Ejemplo como se ve la dirección IPv4 protocolo de internet

Fuente <https://redstelematicas.com/wp-content/uploads/2012/12/9.13.-configuracionIPmanual.jpg>

Cómo se ve la dirección IPv4

Los protocolos de Internet, específicamente el Protocolo de Internet Versión 4, definen cómo funciona el direccionamiento y cómo se identifican y ubican los nodos de red en una red. Una dirección IPv4 se representa como un valor de 32 bits que consta de cuatro octetos (4x8), generalmente con un punto decimal, por ejemplo: 172.140.153.12.

Los protocolos de Internet, específicamente el Protocolo de Internet Versión 4, definen cómo funciona el direccionamiento y cómo se identifican y ubican los nodos de red en una red. Una dirección IPv4 se representa como un valor de 32 bits que consta de cuatro octetos (4x8), generalmente con un punto decimal, por ejemplo: 172.140.153.12.

The anatomy of IPv4 address

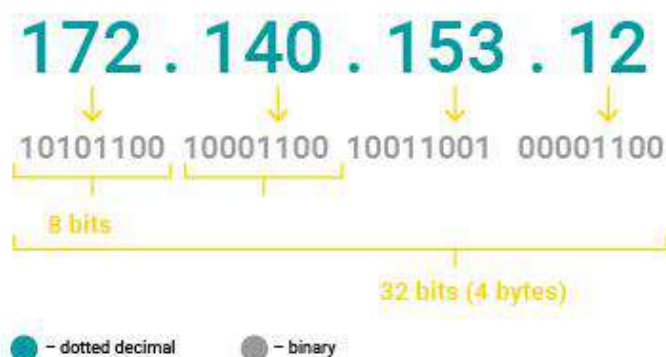


Imagen: Ejemplo como se ve la dirección IPv4

Fuente https://www.avsystem.com/media/dhcp_blog_ipv4-01.jpg

Anatomía de la dirección IPv4

Una dirección IPv4 en realidad se compone de dos partes: una parte identifica su red y la otra parte identifica un host (es decir, un dispositivo) en la red. Estas partes no son iguales y no son fijas, por lo que la dirección también tiene una "máscara de red" para definir la longitud de la parte de la red. En notación CIDR, este es el número después de la barra inclinada que determina cuántos bits de la dirección componen el prefijo de red.

Por ejemplo, 192.168.0.1/24 significa que la longitud del prefijo de la dirección IPv4 192.168.0.1 es de 24 bits, y la red a la que pertenece contiene direcciones de 192.168.0.0 a 192.168.0.255 (es decir, todas comparten los primeros 24 bits). bits).). bits). Al usar protocolos de enrutamiento como RIP, OSPF y BGP, los enrutadores se comunican entre sí las direcciones de red que les han asignado o "conocen" de otros enrutadores para que los paquetes se puedan reenviar a la red correcta. Entonces también el dispositivo correcto.

2.14 El protocolo ICMP

El Protocolo de mensajes de control de Internet (ICMP) es un protocolo de mensajes de error que utilizan los dispositivos de red, como los enrutadores, para generar mensajes de error de dirección IP de origen cuando los problemas de red impiden la entrega de paquetes IP. ICMP crea y envía mensajes a direcciones IP de origen que especifican puertas de enlace (como enrutadores, servicios o hosts) que no pueden acceder a Internet para entregar paquetes. Cualquier dispositivo en una red IP puede enviar, recibir o procesar mensajes ICMP. (ICMP no es un protocolo de transporte que envía datos entre sistemas).

Utilidades

ICMP es un protocolo de capa de red utilizado por enrutadores, dispositivos intermedios y hosts para enviar información de error o actualizaciones a otros enrutadores, dispositivos intermedios y hosts.

ICMP también se usa para diagnósticos de red, específicamente las utilidades de terminal ping y traceroute.

- **Rastreo.** La utilidad traceroute se usa para mostrar la ruta de enrutamiento físico entre dos dispositivos de Internet que se comunican entre sí. Traza una ruta de un enrutador a otro, a veces llamado salto. El uso de traceroute para diagnosticar problemas de red puede ayudar a los administradores a determinar el origen del retraso en la red.
- **Silbido.** La utilidad ping es una ruta de rastreo más simple. Envía pings, también conocidos como mensajes de solicitud de eco, y luego mide la cantidad de tiempo que tarda el mensaje en llegar a su destino y regresar a la fuente. Estas respuestas se denominan mensajes de respuesta de eco. Los pings son útiles para recopilar información de latencia sobre un dispositivo específico. Sin embargo, a diferencia de traceroute, ping no proporciona mapas de imágenes del diseño de enrutamiento.

Función

Es un protocolo sin conexión, lo que significa que el dispositivo no necesita abrir una conexión con el dispositivo de destino antes de enviar un mensaje. Esto difiere de, por ejemplo, TCP, donde se debe establecer una conexión antes de que se pueda enviar un mensaje, donde el protocolo de enlace TCP determina si dos dispositivos están listos.

Un paquete ICMP es un paquete IP con ICMP en la parte de datos IP. El mensaje ICMP también contiene el encabezado IP completo del mensaje original, por lo que el sistema final sabe qué paquete falló.

El encabezado ICMP aparece después del encabezado del paquete IPv4 o IPv6 y se identifica con el número de protocolo IP 1. El protocolo incluye tres parámetros que se describen a continuación. Los tres parámetros son seguidos por datos ICMP y el encabezado IP original que especifica el paquete fallido.

Parámetros

Las opciones de ICMP se encuentran en los encabezados de los paquetes y ayudan a identificar errores en los paquetes IP a los que se refieren. Un parámetro es similar a una etiqueta de envío en un paquete. Proporciona información de identificación sobre el paquete y los datos que contiene. Entonces, los protocolos y las herramientas de red que reciben mensajes ICMP saben cómo procesar los paquetes.

Los primeros 32 bits del encabezado del paquete de cada mensaje ICMP contienen tres campos informativos o parámetros.

Tipo. Los primeros 8 bits son los tipos de mensajes. Algunos tipos de mensajes comunes incluyen los siguientes:

- Tipo 0 -- Respuesta de eco
- Tipo 3 -- Destino inalcanzable
- Tipo 8 -- Eco
- Tipo 5 – Redirigir
-

El tipo proporciona una breve descripción del propósito del mensaje para que el dispositivo de red receptor sepa por qué está recibiendo el mensaje y cómo manejarlo.

Contraseña. Los siguientes 8 bits son un código de tipo de mensaje que proporciona información adicional sobre el tipo de error.

Suma de comprobación. Los últimos 16 bits proporcionan una verificación de integridad del mensaje. La suma de verificación indica la cantidad de bits en todo el mensaje y permite que las herramientas ICMP verifiquen la coherencia con el encabezado del mensaje ICMP para garantizar que se haya pasado todo el rango de datos.



Imagen: El protocolo ICMP ejemplo

Fuente https://cdn.ttgtmedia.com/rms/onlineimages/networking-icmp_destination_unreachable-h.png

2.15 El protocolo UDP

El Protocolo de datagramas de usuario (UDP) es un protocolo de comunicaciones utilizado en Internet para transmisiones particularmente sensibles al tiempo, como la reproducción de video o las búsquedas de DNS. Esto acelera el intercambio de datos al no establecer formalmente una conexión antes de que se envíen los datos. Esto permite que los datos se transmitan muy rápidamente, pero también introduce la pérdida de paquetes durante la transmisión y oportunidades de explotación en forma de ataques DDoS.

Función

El protocolo de datagramas de usuario es un protocolo de comunicación estandarizado que transfiere datos entre computadoras en una red. Sin embargo, a diferencia de otros protocolos como TCP, UDP simplifica la transferencia de datos al enviar paquetes (o, más específicamente, datagramas) directamente al receptor sin establecer primero una conexión bidireccional. UDP no indica el orden de transmisión de sus datagramas ni confirma su llegada.

Encabezado UDP

Los encabezados UDP comprimen los metadatos en datagramas y le dicen al dispositivo qué hacer con los datos enviados. El encabezado tiene cuatro campos de 2 bytes cada uno, como se muestra en el siguiente diagrama.

- El puerto de origen define el puerto del dispositivo emisor. Establecer este campo en 0 indica que el remitente no espera una respuesta de la computadora de destino.
- El puerto de destino define el puerto del dispositivo receptor y puede oscilar entre 0 y 65 535.
- La longitud define el valor en bytes del encabezado UDP y los datos de carga útil. El límite de este campo se basa en el protocolo IP subyacente para la transmisión de datos.
- Finalmente, la suma de verificación permite al receptor verificar la integridad tanto del encabezado como de la carga útil. Era opcional en IPv4 y obligatorio en IPv6.

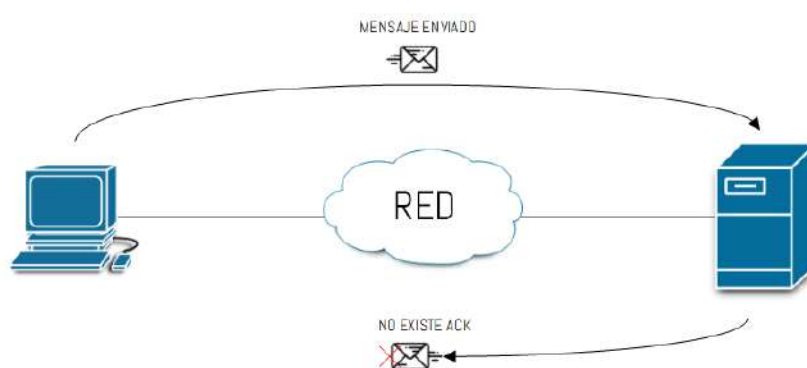


Imagen: El protocolo UDP ejemplo

Fuente: <https://netwgeeks.com/wp-content/uploads/2018/12/No-ACK-UDP.png>

2.16 El protocolo TCP

Es uno de los protocolos centrales del conjunto de protocolos de Internet. Se encuentra entre las capas de aplicación y de red que se utilizan para proporcionar servicios de entrega confiables. Un protocolo de comunicación orientado a la conexión que ayuda a intercambiar mensajes entre diferentes dispositivos a través de una red. El Protocolo de Internet (IP), que describe la tecnología para transferir paquetes de datos entre computadoras, funciona en conjunto con TCP.

TCP divide los datos en pequeños paquetes y los envía a la capa de Protocolo de Internet (IP). Luego, los paquetes se envían a su destino a través de otra ruta.

La capa TCP de su sistema espera a que se complete la transmisión y acusa recibo de todos los paquetes.

Características

Sistema de numeración de segmentos

- TCP rastrea qué segmentos se envían o reciben asignando un número a cada segmento.
- A los bytes de datos transmitidos se les asignan números de bytes específicos y a los segmentos se les asignan números secuenciales.

Orientado a la conexión

- Esto significa que el remitente y el receptor permanecen conectados hasta que se completa el proceso.
- Se conserva el orden de los datos. Es decir, el orden sigue siendo el mismo antes y después de la transmisión.

Dúplex completo

- En TCP, los datos se pueden transmitir del receptor al remitente o viceversa al mismo tiempo.
- Aumenta la eficiencia del flujo de datos entre el emisor y el receptor.

Control de flujo

- El control de flujo limita la velocidad a la que los remitentes envían datos. Esto se hace para garantizar una entrega confiable.
- El receptor insinúa constantemente al remitente cuántos datos puede recibir (utilizando una ventana deslizante).

Control de errores

- TCP implementa un mecanismo de control de errores para una transferencia de datos confiable
- El control de errores está orientado a bytes
- Los segmentos se comprueban para la detección de errores
- El control de errores incluye: gestión de segmentos dañados y segmentos perdidos, segmentos fuera de servicio, segmentos duplicados.

Control de congestión

- TCP tiene en cuenta el nivel de congestión en la red
- El nivel de congestión está determinado por la cantidad de datos enviados por un remitente

Ventajas

- Es un protocolo confiable.
- Proporciona un mecanismo de comprobación de errores, así como uno para la recuperación.
- Da control de flujo.
- Se asegura de que los datos lleguen al destino correcto en el orden exacto en que fueron enviados.
- Protocolo abierto, no propiedad de ninguna organización o individuo.

- Asigna una dirección IP a cada computadora en la red y un nombre de dominio a cada sitio, lo que hace que el sitio de cada dispositivo sea distinguible en la red.

Desventajas

- TCP está hecho para redes de área amplia, por lo que su tamaño puede convertirse en un problema para redes pequeñas con pocos recursos.
- TCP ejecuta varias capas para que pueda ralentizar la velocidad de la red.
- No es de carácter genérico. Es decir, no puede representar ninguna pila de protocolos que no sea la suite TCP/IP. Por ejemplo, no puede funcionar con una conexión Bluetooth.
- Sin modificaciones desde su desarrollo hace unos 30 años.

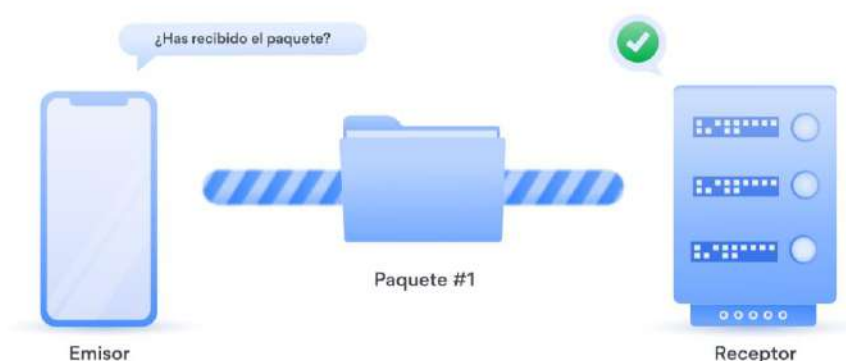


Imagen: El protocolo TCP

Fuente: <https://nordvpn.com/wp-content/uploads/blog-how-tcp-works-infographic-server-phone-es-1200x628-1.svg>

1.1 Dirección de red (Subneteo clase A)

Por: **Didier Joshua Lopez Juarez**

Una dirección de Clase A está formada por una dirección de red de 8 bits y una dirección del sistema principal o local de 24 bits. De las 128 direcciones de red posibles, dos están reservadas para casos especiales: la dirección de red 127 está reservada para las direcciones de retorno de bucle locales y una dirección de red de todo unos indica una dirección de difusión.

Hay 126 direcciones de red de Clase A posibles y 16.777.216 direcciones del sistema principal local posibles. En una dirección de Clase A, el bit más significativo se establece en 0.

Dirección de red (8 bits)	Dirección de host local (24 bits)		
01111101	00001101	01001001	00001111

Nota: el bit de orden superior (o primer bit) será siempre 0 en una dirección de clase A

Imagen: Figura 1. Dirección de Clase A

Fuente: <https://www.ibm.com/docs/es/aix/7.2?topic=addresses-class>

1.1.1 Tipos de clase

En función del tamaño y del alcance de la red de ordenadores, se puede establecer una diferenciación entre diversas dimensiones de red. Entre los tipos de redes más importantes se encuentran:

- **Personal Area Networks (PAN) o red de área personal:**

Personal Area Network (PAN) o red de área personal, aunque las técnicas de transmisión más habituales son la memoria USB o el conector FireWire. La variante inalámbrica Wireless Personal Area Network (WPAN) se basa en técnicas como Bluetooth, Wireless USB, Insteon, IrDA, ZigBee o Z-Wave.

- **Local Area Networks (LAN) o red de área local:**

Una red local de tales características puede incluir a dos ordenadores en una vivienda privada o a varios miles de dispositivos en una empresa. Asimismo, las redes en instituciones públicas como administraciones, colegios o universidades también son redes LAN.

- **Metropolitan Area Networks (MAN) o red de área metropolitana:**

La Metropolitan Area Network (MAN) o red de área metropolitana es una red de telecomunicaciones de banda ancha que comunica varias redes LAN en una zona geográficamente cercana. Por lo general, se trata de cada una de las sedes de una empresa que se agrupan en una MAN por medio de líneas arrendadas.

- **Wide Area Networks (WAN) o red de área amplia:**

Metropolitan Area Networks comunican puntos que se encuentran cerca unos de los otros en regiones rurales o en zonas de aglomeraciones urbanas, las Wide Area Networks (WAN) o redes de área amplia se extienden por zonas geográficas como países o continentes.

- **Global Area Networks (GAN) o red de área global:**

Una red global como Internet recibe el nombre de Global Area Network (GAN), sin embargo, no es la única red de ordenadores de esta índole. Las redes GAN utilizan la infraestructura de fibra de vidrio de las redes de área amplia (Wide Area Networks) y las agrupan mediante cables submarinos internacionales o transmisión por satélite.

1.1.2 Mascara de Subred

La máscara de red es una combinación de bits que sirve en el ámbito de las redes de ordenadores, cuya función es indicar a los dispositivos qué parte de la dirección IP es el número de la red, incluyendo la subred, y qué parte es la correspondiente al host.

CLASE A	Red		Host	
Octeto	1	2	3	4
Bits	11111111	00000000	00000000	00000000
Mascara (defecto)	255	0	0	0

Dirección de Red: Primer octeto (8 bits)

Dirección de Host: Últimos 3 octetos (24 bits)

CLASE B	Red		Host	
Octeto	1	2	3	4
Bits	11111111	11111111	00000000	00000000
Mascara x defecto	255	255	0	0

Dirección de Red: Primeros 2 octetos (16 bits)

Dirección de Host: Últimos 2 octetos (16 bits)

CLASE C	Red			Host
Octeto	1	2	3	4
Bits	11111111	11111111	11111111	00000000
Mascara x defecto	255	255	255	0

Dirección de Red: Primeros 3 octetos (24 bits)

Dirección de Host: Último octeto (8 bits)

Imagen: Mascaras de subred con sus clases.

Fuente: <https://underc0de.org/foro/redes/subneteo-clase-a-b-c/>

1.1.3 Redes Útiles

La función del Subneteo o Subnetting es dividir una red IP física en subredes lógicas (redes más pequeñas) para que cada una de estas trabaje a nivel envío y recepción de paquetes como una red individual, aunque todas pertenezcan a la misma red física y al mismo dominio.

1.1.4 Dirección IP

Una dirección IP es una dirección única que identifica a un dispositivo en Internet o en una red local. IP significa "protocolo de Internet", que es el conjunto de reglas que rigen el formato de los datos enviados a través de Internet o la red local.

1.1.4.1 Clase A

Son las que en sus primeros 8 bits contienen un bit 0 fijo (indicando clase A) y 7 bits variables. Es decir que se pueden diferenciar de 00000000=0 a 01111111=127.

Clases de Direcciones IP

Clase	Rango del primer número	Ejemplos
Clase A	0 - 126	10.52.152.65 100.1.22.34
Clase B	128 - 191	132.23.34.76 188.200.2.59
Clase C	192 - 223	192.188.66.33 200.9.176.50

Imagen: IP y sus clases.

Fuente: <https://www.goconqr.com/apunte/3350241/direcciones-ip>

1.1.5 Asignación de Subred

Las direcciones de Internet las asigna InterNIC, la organización que administra Internet. Estas direcciones IP se dividen en clases. Las más comunes son las clases A, B y C.

- Las redes de clase A usan una máscara de subred predeterminada de 255.0.0.0 y tienen de 0 a 127 como su primer octeto. La dirección 10.52.36.11 es una dirección de clase A. Su primer octeto es 10, que está entre 1 y 126, ambos incluidos.
- Las redes de clase B usan una máscara de subred predeterminada de 255.255.0.0 y tienen de 128 a 191 como su primer octeto. La dirección 172.16.52.63 es una dirección de clase B. Su primer octeto es 172, que está entre 128 y 191, ambos inclusive.
- Las redes de clase C usan una máscara de subred predeterminada de 255.255.255.0 y tienen de 192 a 223 como su primer octeto. La dirección 192.168.123.132 es una dirección de clase C. Su primer octeto es 192, que está entre 192 y 223, ambos incluidos.

1.1 Subneteo de clase B

Por: Byron Mansilla

El subneteo de una dirección IP de clase B implica dividir la red en subredes más pequeñas para aprovechar de manera eficiente las direcciones IP disponibles. Para subnetear una dirección IP de clase B, debes asignar una parte de los bits de host a los bits de red.

Una dirección IP de clase B tiene el formato: A.B.X.X, donde A y B representan los bits de red, y X representa los bits de host.

CLASE B	Red		Host	
Octeto	1	2	3	4
Bits	11111111	11111111	00000000	00000000
Mascara x defecto	255	255	0	0

Dirección de Red: Primeros 2 octetos (16 bits)
Dirección de Host: Últimos 2 octetos (16 bits)

Imagen: Subneteo de clase b

Fuente: <https://underc0de.org/foro/redes/subneteo-clase-a-b-c/>

1.1.1 No.de subredes

En una dirección IP de clase B, se pueden crear hasta 16,384 subredes. Esto se debe a que una dirección IP de clase B utiliza los primeros 16 bits para la parte de la red y los siguientes 16 bits para la parte de host.

Para calcular el número de subredes posibles, debes tener en cuenta los bits de host disponibles y utilizar la fórmula 2^n , donde "n" representa el número de bits de host utilizados para subredear.

Dirección IPv4/prefijo	Dirección de red	Dirección de broadcast	Cantidad total de bits de host	Cantidad total de hosts
192.168.100.25/28	192.168.100.16	192.168.100.31	4	14
172.30.10.130/30	172.30.10.128	172.30.10.131	2	2
10.1.113.75/19	10.1.96.0	10.1.127.255	13	8190
198.133.219.250/24	198.133.219.0	198.133.219.255	8	254
128.107.14.191/22	128.107.12.0	128.107.15.255	10	1022
172.16.104.99/27	172.16.104.96	172.16.104.127	5	30

Imagen: Numero de subredes de una ip

Fuente:

<https://ciscoconnasonson.blogspot.com/2015/08/calculo-de-subredes-ipv4.html>.

1.1.2 Tipos de clase

Clase A: Las direcciones IP de clase A se utilizaban para redes grandes. El primer octeto (8 bits) representaba la parte de red, y los tres octetos restantes se usaban para la parte de host. El rango de direcciones de clase A iba desde 1.0.0.0 hasta 126.0.0.0.

Clase B: Las direcciones IP de clase B se usaban para redes de tamaño mediano. Los primeros dos octetos (16 bits) representaban la

parte de red, y los dos octetos restantes se usaban para la parte de host. El rango de direcciones de clase B iba desde 128.0.0.0 hasta 191.255.0.0.

Clase C: Las direcciones IP de clase C se usaban para redes pequeñas. Los primeros tres octetos (24 bits) representaban la parte de red, y el último octeto se usaba para la parte de host. El rango de direcciones de clase C iba desde 192.0.0.0 hasta 223.255.255.0.

Clase D: Las direcciones IP de clase D se reservaron para el uso de grupos multidifusión. Estas direcciones comienzan con los primeros cuatro bits establecidos en 1110. El rango de direcciones de clase D iba desde 224.0.0.0 hasta 239.255.255.255.

Clase E: Las direcciones IP de clase E se reservaron para uso experimental o futuro. Estas direcciones comienzan con los primeros cuatro bits establecidos en 1111. El rango de direcciones de clase E iba desde 240.0.0.0 hasta 255.255.255.255.

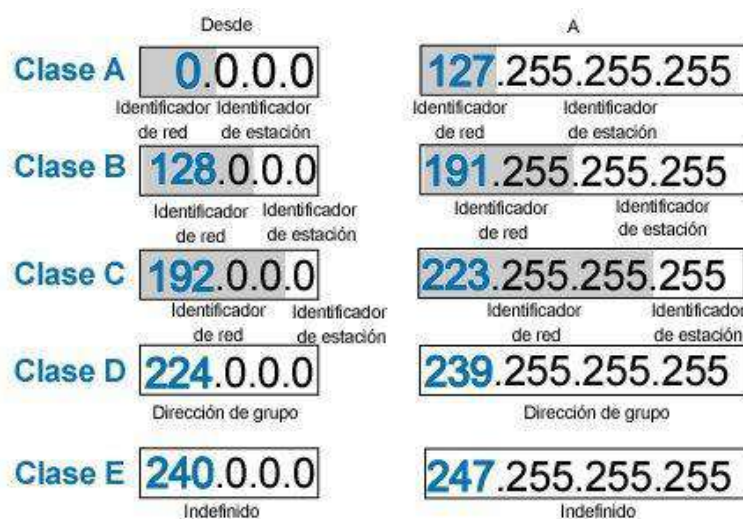


Imagen: Ejemplo de tipos de clases

Fuente:

<https://alejollagua.blogspot.com/2012/12/direccion-ip-clase-b-c-d-y-e.html>

1.1.3 Mascara de subred

Una máscara de subred es una combinación de bits que se utiliza para separar la parte de red y la parte de host de una dirección IP. Se aplica a una dirección IP para determinar qué parte de la dirección corresponde a la red y qué parte corresponde a los dispositivos en esa red. La máscara de subred se representa comúnmente como una serie de números separados por puntos, al igual que una dirección IP. Estos números representan los bits de la máscara de subred establecidos en 1 (bits de red) y los bits establecidos en 0 (bits de host).

Aquí hay algunos ejemplos de máscaras de subred comunes:
 Máscara de subred de clase A: 255.0.0.0 o /8 (en notación CIDR)
 Máscara de subred de clase B: 255.255.0.0 o /16
 Máscara de subred de clase C: 255.255.255.0 o /24.

Dirección IPv4	192	.	168	.	1	.	67
	11000000		10101000		00000001		01000011
Máscara de subred	255	.	255	.	255	.	0
	11111111		11111111		11111111		00000000
	11000000		10101000		00000001		00000000
Dirección de la red	192	.	168	.	1	.	0

Fuente:

<https://www.adslzone.net/reportajes/internet/que-es-mascara-de-subred/>

Imagen: Ejemplo de mascara de subred de una ip

1.1.4 Redes útiles

se refiere a las subredes o rangos de direcciones IP que se utilizan para asignar a dispositivos o redes específicas dentro de una red más grande. Estas redes útiles son aquellas que se utilizan para alojar hosts y permitir la conectividad y comunicación entre ellos. El término "redes útiles" generalmente se utiliza en el contexto del subneteo, donde se dividen las redes en subredes más pequeñas para aprovechar eficientemente las direcciones IP disponibles y organizar la infraestructura de red.

0	Dirección de la Red (7 bits)	Dirección del Host (24 bits)	CLASE A
1 0	Dirección de la Red (14 bits)	Dirección del Host (16 bits)	CLASE B
1 1 0	Dirección de la Red (21 bits)	Dirección del Host (8 bits)	CLASE C
1 1 1 0	Dirección Multicast (28 bits)		CLASE D
1 1 1 1 0	Uso Futuro (28 bits)		CLASE E

Imagen: Ejemplo de una red útil.

Fuente:

https://www.oas.org/en/citel/infocitel/2009/junio/protocolo2_e.asp

1.1.5 Dirección IP

Una dirección IP (Protocolo de Internet) es una etiqueta numérica única asignada a cada dispositivo conectado a una red de computadoras que utiliza el protocolo de Internet para comunicarse. Permite que los dispositivos se identifiquen y se comuniquen entre sí a través de la red.

Las direcciones IP se componen de una serie de números separados por puntos y se dividen en dos partes: la parte de red y la parte de host.

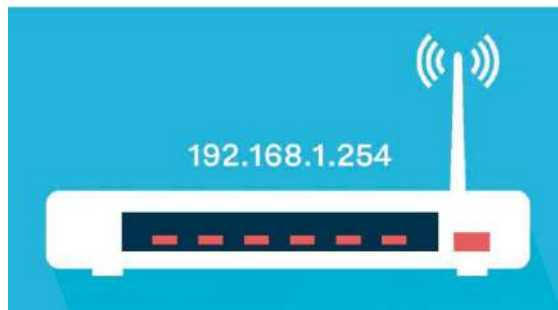


Imagen: ejemplo de una direccion ip

Fuente:

<https://blog.educacionit.com/2019/04/04/que-es-una-direccion-ip-puede-revelar-tu-ubicacion/>

1.1 Asignación de subred

Por: German Daniel Marquez Estrada

La asignación de subredes, también conocida como subdivisión de redes, es un proceso mediante el cual una red IP se divide en subredes más pequeñas. Esto se realiza mediante la manipulación de los bits de la dirección IP y la máscara de subred correspondiente.

La asignación de subredes es comúnmente utilizada en redes de computadoras para segmentar una red en subredes más pequeñas. Al dividir una red en subredes, se obtienen varios beneficios, como:

Mayor eficiencia en el uso de direcciones IP: Al subdividir una red, se pueden utilizar rangos de direcciones IP más pequeños, lo que permite un uso más eficiente de las direcciones disponibles.

Mayor control y seguridad: La asignación de subredes permite aplicar políticas de seguridad y control de acceso de manera más granular en diferentes partes de la red.

Mayor rendimiento: Al segmentar una red en subredes más pequeñas, se pueden reducir el tráfico de difusión y aumentar el rendimiento general de la red.

El proceso de asignación de subredes implica seleccionar una máscara de subred adecuada y definir los rangos de direcciones IP para cada subred. Esto se logra al determinar la cantidad de bits de la dirección IP que se utilizarán para la identificación de la red y la cantidad de bits que se reservarán para identificar los hosts dentro de cada subred.

La asignación de subredes se realiza generalmente utilizando el estándar IPv4 o el protocolo de Internet versión 4. Sin embargo, con la adopción creciente del estándar IPv6, que proporciona un espacio de direcciones mucho más amplio, la necesidad de asignación de subredes se reduce significativamente debido a la abundancia de direcciones disponibles en IPv6.

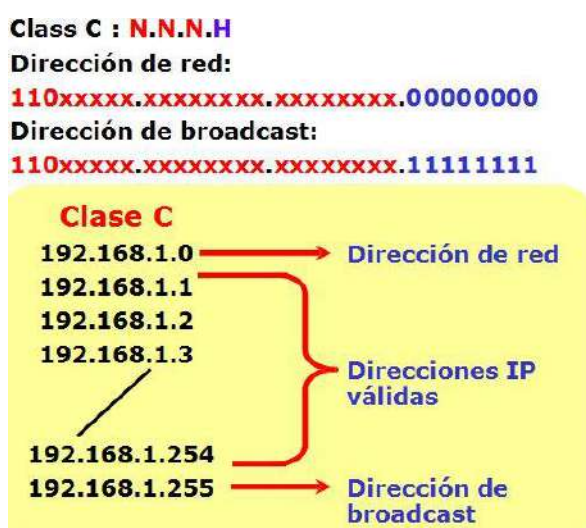


Imagen: Ejemplo de clase C de subneteo

Fuente: https://www.quia.com/files/quia/users/istomar/DIPS/Day-2_-_IP_addressing_-_20.JPG

1.1.1 Subneteo Clase C

La función del Subneteo o Subnetting es dividir una red IP física en subredes lógicas (redes más pequeñas) para que cada una de estas trabajen a nivel envío y recepción de paquetes como una red individual, aunque todas pertenezcan a la misma red física y al mismo dominio.

El Subneteo permite una mejor administración, control del tráfico y seguridad al segmentar la red por función.

También, mejora la performance de la red al reducir el tráfico de broadcast de nuestra red.

El ejercicio nos pedía un mínimo de 50 hosts por subred. Para esto utilizamos la fórmula $2^M - 2$, donde M es el número de bits "0" disponibles en la porción de host y - 2 porque la primer y última dirección IP de la subred no se utilizan por ser la dirección de la subred y broadcast respectivamente.

$2^6 - 2 = 62$ hosts por subred.

Los 6 bits "0" de la porción de host (fondo gris) son los vamos a utilizar según vayamos asignando los hosts a las subredes.

Nos dan la dirección de red Clase C 192.168.1.0 /24 para realizar mediante subneteo 4 subredes con un mínimo de 50 hosts por subred.

Lo vamos a realizar con 3 pasos:

Ya tenemos nuestra máscara de red adaptada que va a ser común a todas las subredes y hosts que componen la red.

Ahora queda obtener los hosts. Para esto vamos a trabajar con la dirección IP de red, específicamente con la porción de host (fondo gris).

Adaptar la Máscara de Red por Defecto a Nuestras Subredes

PASO 1: La máscara por defecto para la red 192.168.1.0 es:

PASO 2: Obtener Cantidad de Hosts por Subred

Usando la fórmula 2^N , donde N es la cantidad de bits que tenemos que robarle a la porción de host, adaptamos la máscara de red por defecto a la subred.

Se nos solicitaron 4 subredes, es decir que el resultado de 2^N tiene que ser mayor o igual a 4. Como vemos en el gráfico, para hacer 4 subredes debemos robar 2 bits a la porción de host.

Agregamos los 2 bits robados reemplazándolos por "1" a la máscara Clase C por defecto y obtenemos la máscara adaptada 255.255.255.192.

1.1.2 No. De subredes

El número de subredes posibles en el subneteo de una red de clase C depende de la cantidad de bits que se utilicen para el subneteo.

En una red de clase C, se tienen 8 bits disponibles para identificar los hosts dentro de la red (ya que los primeros 24 bits se reservan para identificar la red). Al utilizar n bits para el subneteo, se obtienen 2^n subredes posibles.

Dado que en una red de clase C se disponen de 8 bits para el subneteo, se pueden obtener $2^8 = 256$ subredes posibles. Sin embargo, algunas de estas subredes pueden estar reservadas para fines especiales, como la dirección de broadcast (todos los bits del octeto de host en 1) o la dirección de loopback (127.0.0.0/8).

Por lo tanto, en el subneteo de una red de clase C, generalmente se consideran menos de 256 subredes disponibles para su asignación. La cantidad exacta de subredes dependerá de los requisitos y la configuración específica de la red.

1.1.3 Dirección de red

La dirección de red es la dirección IP que se utiliza para identificar una red específica. En una dirección IP, los bits de la dirección de red indican la red a la que pertenece un dispositivo, mientras que los bits restantes se utilizan para identificar los hosts dentro de esa red.

En el caso de una dirección IPv4 de clase C, los primeros 24 bits (3 octetos) se utilizan para identificar la red, mientras que los últimos 8 bits (1 octeto) se utilizan para identificar los hosts.

Por ejemplo, si tenemos la dirección IP 192.168.0.100/24, los primeros 24 bits (192.168.0) corresponden a la dirección de red, y el último octeto (100) identifica un host específico dentro de esa red. En este caso, la dirección de red sería 192.168.0.0.

La dirección de red es importante para el enrutamiento de paquetes en una red. Los dispositivos y routers utilizan la dirección de red para determinar si un paquete debe ser enviado a través de la misma red local o si debe ser enrutado hacia otra red.

Es importante tener en cuenta que la dirección de red puede variar dependiendo de la máscara de subred utilizada. La máscara de subred define cuántos bits se utilizan para la identificación de la red y cuántos bits se utilizan para identificar los hosts dentro de la red.

1.1.4 Tipos de clase

Las direcciones IP se clasificaban en clases para determinar la cantidad de bits utilizados para identificar la red y los hosts en una dirección IP. Estas clases eran conocidas como Clase A, Clase B, Clase C, Clase D y Clase E. Sin embargo, con la implementación de enrutamiento basado en CIDR (Classless Inter-Domain Routing), la clasificación de clases ya no es tan relevante. Aun así, aquí tienes una descripción de las clases IP:

Clase A: Las direcciones de Clase A se identifican por tener el primer octeto reservado para identificar la red, mientras que los tres octetos restantes se utilizan para identificar los hosts. La primera dirección de Clase A comienza con los bits "0" en el primer octeto, lo que significa que va de 0.0.0.0 a 127.255.255.255. Las direcciones de Clase A se utilizaban principalmente para redes grandes.

Clase B: Las direcciones de Clase B utilizan los dos primeros octetos para identificar la red y los dos octetos restantes para identificar los hosts. La primera dirección de Clase B comienza con los bits "10" en los primeros dos octetos, lo que significa que va de 128.0.0.0 a 191.255.255.255. Las direcciones de Clase B se utilizaban comúnmente para redes de tamaño mediano.

Clase C: Las direcciones de Clase C utilizan los tres primeros octetos para identificar la red y el último octeto para identificar los hosts. La primera dirección de Clase C comienza con los bits "110" en los primeros tres octetos, lo que significa que va de 192.0.0.0 a 223.255.255.255. Las direcciones de Clase C se utilizaban para redes pequeñas.

Clase D: Las direcciones de Clase D están reservadas para multicast. Comienzan con los bits "1110" en los primeros cuatro octetos. Las direcciones de Clase D van de 224.0.0.0 a 239.255.255.255.

1.1.5 Redes Útiles

En una red de clase C, la cantidad de redes útiles se refiere a la cantidad de subredes disponibles para su uso, excluyendo la dirección de red y la dirección de broadcast. La cantidad de redes útiles puede variar dependiendo de la máscara de subred utilizada.

En una red de clase C, utilizando una máscara de subred de longitud fija (/24), se tienen 256 direcciones IP disponibles. Sin embargo, de estas 256 direcciones, una se reserva como dirección de red y otra como dirección de broadcast, lo que deja 254 direcciones IP utilizables para los hosts.

Si se utilizan máscaras de subred más largas, como /25, /26, /27, etc., se obtienen más subredes, pero con un número menor de direcciones IP utilizables en cada una.

Aquí hay algunos ejemplos de la cantidad de redes útiles para diferentes máscaras de subred de clase C:

Máscara de subred /24: 1 red útil (256 direcciones IP, excluyendo dirección de red y dirección de broadcast).

Máscara de subred /25: 2 redes útiles (128 direcciones IP en cada subred).

Máscara de subred /26: 4 redes útiles (64 direcciones IP en cada subred).

Máscara de subred /27: 8 redes útiles (32 direcciones IP en cada subred).

Máscara de subred /28: 16 redes útiles (16 direcciones IP en cada subred).

Es importante tener en cuenta que, en cada subred, una dirección se reserva como la dirección de red y otra como la dirección de broadcast, lo que reduce la cantidad de direcciones IP utilizables para los hosts.

La cantidad de redes útiles y las direcciones IP utilizables en cada subred pueden variar dependiendo de la configuración específica y los requisitos de la red. Es recomendable planificar cuidadosamente el tamaño de las subredes y las direcciones IP requeridas para satisfacer las necesidades de los hosts y dispositivos de la red.

1.1 ¿Qué es IP?

Por: Elder César Oswaldo Martínez Rustrian

La sigla "IP" significa "Protocolo de Internet". Básicamente, es una especie de identificación numérica asignada a cada dispositivo conectado a una red de Internet, ya sea una computadora, un teléfono inteligente o cualquier otro aparato.

1.1.1 ¿Qué es Asignación de Subred?

La asignación de subred es un proceso en el que se divide una red de computadoras más grande en redes más pequeñas llamadas subredes. Esto se hace para organizar y administrar de manera más eficiente los dispositivos dentro de la red.

1.1.2 ¿Qué es conexiones de red?

Las conexiones de red se refieren a la forma en que los dispositivos se comunican y comparten información entre sí a través de una red de computadoras. Cuando hablamos de conexiones de red, nos referimos a los enlaces o vínculos establecidos entre diferentes dispositivos, como computadoras, servidores, impresoras u otros dispositivos conectados en una red.

1.1.3 ¿Qué es la comunicación de cliente/servidor?

La comunicación de cliente/servidor es un modelo de interacción en el que un dispositivo, llamado cliente, solicita y obtiene servicios o recursos de otro dispositivo, llamado servidor. En este servidor y el cliente envía una petición al servidor y el servidor responde proporcionando la información o realizando la tarea solicitada.

1.1.4 ¿Qué es una conexión de red con gestor de base de datos?

Las conexiones de red con gestor de base de datos se refieren a la comunicación establecida entre una aplicación o cliente y un gestor de base de datos a través de una red de computadoras. En este contexto, un gestor de base de datos es un software especializado que permite almacenar, organizar y recuperar datos de manera eficiente.

1.1 W3C

Por: Adolfo Morales

El World Wide Web Consortium (W3C) es una organización internacional que se dedica a desarrollar estándares abiertos para la Web. Fue fundado en 1994 por Tim Berners-Lee, el inventor de la World Wide Web, con el objetivo de garantizar la interoperabilidad y el crecimiento continuo de la Web.

1.1.1 Importancia del W3C

El trabajo del W3C es fundamental para garantizar un desarrollo coherente y sostenible de la Web. Al establecer estándares abiertos y promover las mejores prácticas, el W3C fomenta la innovación, la interoperabilidad y la accesibilidad en la Web. Esto beneficia tanto a los usuarios finales como a los desarrolladores y empresas que construyen aplicaciones y servicios web.

El cumplimiento de los estándares del W3C también es importante para garantizar que los sitios web sean accesibles a personas con discapacidades, lo que promueve la inclusión y la igualdad de oportunidades en el entorno digital.

Además, el W3C desempeña un papel crucial en la promoción de la privacidad y la seguridad en la Web, asegurando que las tecnologías web estén diseñadas para proteger los datos personales y prevenir riesgos de seguridad.



Fuente: <https://www.loginradius.com/blog/engineering/w3c-validation/>

1.1.2 HTML

(HyperText Markup Language) es el lenguaje estándar utilizado para crear y diseñar páginas web. Fue desarrollado por el World Wide Web Consortium (W3C) con el propósito de estructurar el contenido de una página web y definir su apariencia. HTML utiliza etiquetas y elementos para marcar y organizar el contenido, permitiendo a los navegadores interpretar y mostrar la información correctamente.

1.1.3 Características principales de HTML:

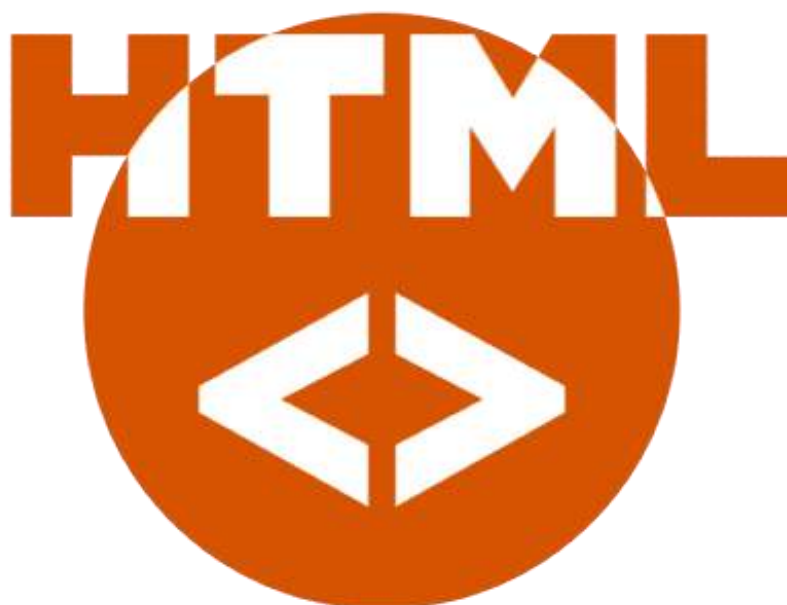
Estructura basada en etiquetas: HTML utiliza etiquetas para marcar y estructurar el contenido de una página web. Cada etiqueta tiene una función específica y se coloca entre corchetes angulares (< y >). Por ejemplo, la etiqueta <h1> se utiliza para definir un encabezado principal.

Estructura jerárquica: HTML sigue una estructura jerárquica, lo que significa que los elementos se anidan unos dentro de otros para formar una estructura de árbol. La etiqueta <html> es el elemento raíz y contiene todos los demás elementos de la página.

Etiquetas semánticas: HTML ofrece una variedad de etiquetas semánticas que permiten describir el significado y la estructura del contenido. Estas etiquetas, como <header>, <nav>, <section> y <footer>, proporcionan información adicional para los motores de búsqueda y ayudan a los navegadores y asistentes de lectura a interpretar correctamente la página.

Enlaces e hipertexto: HTML permite crear enlaces a otras páginas web, recursos externos o ubicaciones dentro de la misma página. Los enlaces se crean utilizando la etiqueta <a> (ancla) y permiten la navegación entre diferentes partes de la Web.

Imágenes y medios: HTML admite la inserción de imágenes, audio y video en una página web. Para mostrar imágenes, se utiliza la etiqueta , mientras que para reproducir audio y video se utilizan las etiquetas <audio> y <video> respectivamente.



Fuente: <https://desarrolloweb.com/home/html>

1.1.4 CSS

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para definir la apariencia y el diseño de un documento HTML. Se utiliza para describir cómo se deben mostrar los elementos HTML en una página web, controlando aspectos como el color, la tipografía, el espaciado, el diseño y otros estilos visuales.

CSS es un lenguaje de hojas de estilo utilizado para definir la apariencia visual de una página web, separando la presentación de la estructura del contenido HTML.

Proporciona un control completo sobre el diseño, los estilos y la apariencia de los elementos HTML, lo que permite crear diseños atractivos y coherentes en toda la web.

1.1.5 Características principales de CSS:

Separación de la estructura y la presentación: CSS permite separar la estructura del contenido HTML de su presentación visual. Mientras que HTML se utiliza para definir la estructura y el significado del contenido, CSS se encarga de aplicar estilos y diseños a esos elementos.

Selectores y reglas: CSS utiliza selectores para identificar los elementos HTML a los que se aplicarán los estilos. Los selectores pueden hacer referencia a etiquetas HTML específicas, clases, identificadores u otros atributos. Las reglas CSS se definen mediante un selector seguido de una serie de propiedades y valores, que determinan cómo se visualizarán los elementos seleccionados.

Cascada y herencia: CSS utiliza el concepto de cascada, lo que significa que múltiples reglas CSS pueden aplicarse a un mismo elemento y la aplicación final de los estilos se determina según una jerarquía de especificidad. Además, CSS permite la herencia de estilos, lo que significa que los estilos definidos en un elemento pueden ser heredados por sus elementos hijos, simplificando la aplicación de estilos en toda la página.

Unidades de medida: CSS proporciona diversas unidades de medida para especificar tamaños y dimensiones, como píxeles (px), porcentajes (%), em, rem, entre otros. Estas unidades permiten crear diseños flexibles y adaptables a diferentes dispositivos y tamaños de pantalla.

Estilos avanzados: CSS ofrece una amplia gama de características y propiedades para controlar la apariencia y el diseño de una página web. Esto incluye la manipulación de colores, fuentes, márgenes, rellenos, bordes, sombras, animaciones, transformaciones y más. CSS3, la versión más reciente en el momento de redactar este informe, introdujo numerosas características avanzadas y efectos visuales.



Fuente: <https://www.oxfordwebstudio.com/en/did-you-know/what-is-css>

1.1.6 Graficos

Modelo de caja en CSS:

Este gráfico muestra el modelo de caja de CSS, que representa cómo se estructura y se visualiza un elemento HTML en CSS. El gráfico ilustra las partes de la caja, como el contenido, el padding, el borde y el margen, y cómo interactúan entre sí para determinar la apariencia final del elemento.

Selectores CSS:

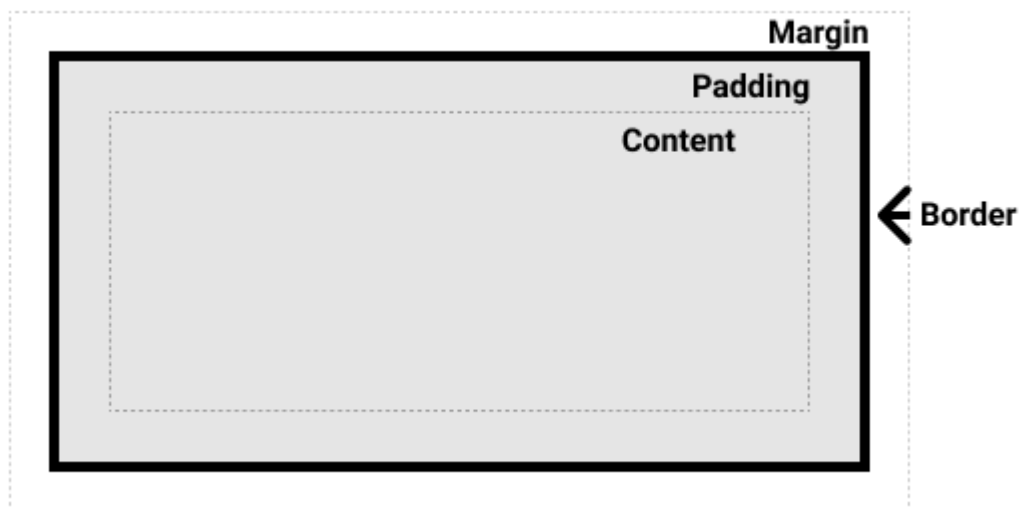
Este gráfico muestra diferentes tipos de selectores en CSS, que se utilizan para seleccionar y aplicar estilos a elementos HTML específicos. Incluye selectores de etiqueta, de clase, de ID, selectores descendientes y más.

Flexbox:

El gráfico de Flexbox muestra cómo funciona el modelo de diseño flexible de CSS, conocido como Flexbox. Muestra cómo se alinean y distribuyen los elementos dentro de un contenedor flex, utilizando propiedades como flex-direction, justify-content, align-items, entre otras.

Grid Layout:

El gráfico de Grid Layout ilustra cómo se organiza y distribuye el contenido utilizando el sistema de diseño de cuadrícula de CSS, conocido como Grid Layout. Muestra cómo se definen filas y columnas y cómo los elementos se colocan en la cuadrícula utilizando propiedades como grid-template-columns, grid-template-rows, grid-column, grid-row, etc.



Fuente: <https://developer.mozilla.org/es/>

1.3 Audio y Video

Por: Erick Estuardo Moran Pech

El elemento audio se usa para insertar contenido de audio en un documento HTML o XHTML. El elemento audio se agregó como parte de HTML 5. Puedes usar las características API de audio mejoradas - que son específicas de Gecko - para generar y manipular directamente secuencias de audio a partir de código JavaScript.

Contenido permitido	Contenido transparente , que contiene bien un atributo <code>src</code> , bien uno o más elementos <code><source></code> , seguido de contenido dinámico o contenido estático , sin ningún elemento de <code><video></code> o <code><audio></code> .
Omisión de etiquetas	Ninguna, deben estar presentes tanto las etiquetas de inicio como las de cierre.
Elementos primarios permitidos	Cualquier elemento que acepte contenido dinámico o cualquier elemento que acepte contenido estático .
Documento normativo	HTML5, sección 4.8.7

1.4 internacionalización

Es "el proceso de diseñar software de manera tal que pueda adaptarse a diferentes idiomas y regiones sin la necesidad de realizar cambios de ingeniería ni en el código". ¿Qué relación guarda la internacionalización con la accesibilidad web? Está claro que ofrecer un contenido en el idioma preferido del usuario facilita su uso y comprensión para todo el mundo, con o sin discapacidad.



<https://www.w3.org/WAI/WCAG1-Conformance.html.es>

En WCAG 1.0, la pauta 4 Identifique el idioma usado contiene la siguiente explicación y los siguientes puntos de verificación:

Use marcadores que faciliten la pronunciación o interpretación de texto abreviado o extranjero.

Cuando los desarrolladores de contenido especifican los cambios en el idioma de un documento, los sintetizadores de voz y los dispositivos braille pueden cambiar automáticamente al nuevo lenguaje, haciendo el documento más accesible a usuarios

multilingües. Los desarrolladores de contenido deberían identificar el idioma predominante del contenido de un documento (a través de un marcador o en el encabezado HTTP). Deberían también proporcionar la expansión de las abreviaturas y los acrónimos.

Además de apoyar a las ayudas técnicas, la identificación del idioma usado permite a los motores de búsqueda localizar las palabras claves e identificar los documentos en el idioma deseado. Los marcadores de idioma mejoran también la legibilidad de la Web para todo el mundo, incluso para aquellos con discapacidades de aprendizaje, cognitivas o sordera.

Cuando los cambios en las abreviaturas y el idioma no son identificados, pueden ser indescifrables para los lectores de pantalla y los dispositivos braille.

4.1 Identifique claramente los cambios en el idioma del texto del documento y en cualquier texto equivalente (Por ejemplo, leyendas). [Prioridad 1]

Por ejemplo en HTML, utilice el atributo "lang". En XML, utilice "xml:lang".

4.2 Especifique la expansión de cada abreviatura o acrónimo cuando aparezcan por primera vez en el documento. [Prioridad 3]

Por ejemplo, en HTML, use el atributo "title" de los elementos "ABBR" y "ACRONYM". Proporcionar la expansión en el cuerpo principal del documento también ayuda a la usabilidad del documento.

4.3 Identifique el idioma principal de un documento. [Prioridad 3]

Por ejemplo, en HTML, coloque el atributo "lang" en el elemento HTML. En XML, utilice "xml:lang". Los operadores de servidores podrían configurar sus servidores para aprovechar los mecanismos de transferencia del contenido del protocolo HTTP ([RFC2068], sección 14.13), de forma que los clientes puedan recibir automáticamente los documentos en el idioma seleccionado.

En WCAG 2.0, la pauta 3.1 Readable: Make text content readable and understandable contiene los siguientes criterios de éxito (success criterion):

3.1.1 Language of Page: The default human language of each Web page can be programmatically determined. (Level A)

3.1.2 Language of Parts: The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text. (Level AA)



<https://www.insuit.net/wp-content/uploads/2017/01/W3C.jpg>

1.1 W3C

Por: Maria Jose Noj Sor

El Consorcio WWW, en inglés: World Wide Web Consortium, es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo.

1.1.1 Redes Moviles

Una red móvil o celular es una red de telecomunicación con enlaces inalámbricos hacia y desde los nodos finales. Es una red formada por celdas de radio cada una con su propio transmisor, conocidas como estación base.

1.1.2 Que Es Privacidad

La privacidad puede ser definida como el ámbito de la vida personal de un individuo, quien se desarrolla en un espacio reservado, el cual tiene como propósito principal mantenerse confidencial.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: <https://www.google.com/search?q=imagen+ejemplo>



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: <https://www.google.com/search?q=imagen+ejemplo>

1.1 W3C

Por: José Antonio Osoy Crespo

1.1.1 Historia y Evolución

El W3C, o World Wide Web Consortium, fue fundado en 1994 por Tim Berners-Lee con el objetivo de desarrollar estándares abiertos para la web y promover su accesibilidad e interoperabilidad. Desde su inicio, el W3C ha sido un líder en la evolución de la web, trabajando en el desarrollo de tecnologías clave como HTML, CSS y protocolos de comunicación. A lo largo de los años, ha desempeñado un papel fundamental en la creación de estándares que han impulsado la funcionalidad y el diseño de los sitios web, así como la protección de la privacidad y la seguridad en línea. Mediante un proceso abierto y colaborativo, el W3C ha involucrado a expertos de todo el mundo y ha establecido pautas ampliamente adoptadas, como las WCAG, para garantizar que la web sea accesible para todos los usuarios y pueda seguir evolucionando de manera efectiva en el futuro.



<https://disenowebakus.net/w3c.php>

1.1.2 Versiones

Aunque el W3C no tiene "versiones" en el sentido tradicional, sí ha lanzado varias versiones de estándares y recomendaciones técnicas a lo largo de su historia. Algunos ejemplos notables incluyen:

1. HTML: El lenguaje de marcado HTML ha pasado por diferentes versiones a lo largo de los años, desde HTML 2.0 en 1995 hasta HTML5, la versión más reciente y ampliamente utilizada. Cada versión trajo consigo nuevas características y mejoras para el desarrollo de contenido web.



Fuente: <https://desarrolloweb.com/home/html/>

2. CSS: Las Cascading Style Sheets (CSS) también han evolucionado con el tiempo. CSS1 fue lanzado en 1996, seguido de CSS2 en 1998. Posteriormente, se introdujo CSS2.1, que se convirtió en un estándar estable en 2009. CSS3 es una especificación más amplia y modular, con varias partes en desarrollo y adopción gradual.



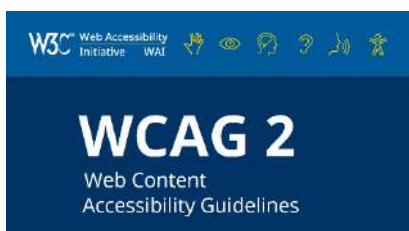
Fuente: https://play.google.com/store/apps/details?id=com.codeliber.css&hl=en_US

3. DOM: El Document Object Model (DOM) es una interfaz de programación que permite acceder y manipular la estructura de un documento HTML o XML. El DOM Level 1 se estableció en 1998, y se han lanzado versiones posteriores, como DOM Level 2 y DOM Level 3, para ampliar la funcionalidad y la compatibilidad.



Fuente: <https://desarrolloweb.com/articulos/que-es-el-dom.html>

4. WCAG: Las Web Content Accessibility Guidelines (WCAG) proporcionan pautas para hacer que los sitios web sean accesibles para personas con discapacidades. Han sido actualizadas en varias versiones, siendo WCAG 2.0 lanzada en 2008 y WCAG 2.1 lanzada en 2018.



Fuente: <https://www.w3.org/TR/WCAG21/>

1.1 Informe sobre el World Wide Web Consortium (W3C)

Por: **Diego Racanac**

El World Wide Web Consortium, conocido como W3C, es un consorcio internacional dedicado al desarrollo de estándares abiertos para la web. Fundado en 1994 por Tim Berners-Lee, el W3C desempeña un papel fundamental en la evolución de la web y su accesibilidad para todos los usuarios.

Objetivos del W3C:

El principal objetivo del W3C es promover la interoperabilidad y el crecimiento de la web a través de estándares técnicos. Algunos de los objetivos específicos del W3C incluyen:

- **Estándares abiertos:** El W3C trabaja para desarrollar y mantener estándares abiertos que permitan a los desarrolladores crear sitios web compatibles y consistentes en diferentes plataformas y dispositivos.
- **Accesibilidad:** El consorcio se esfuerza por hacer que la web sea accesible para todas las personas, incluyendo aquellas con discapacidades. A través de la creación de estándares de accesibilidad, como las Web Content Accessibility Guidelines (WCAG),
- **Innovación:** El W3C impulsa la innovación en la web al promover la adopción de nuevas tecnologías y estándares. Esto incluye el desarrollo de lenguajes de marcado como HTML y XML, así como el avance de tecnologías emergentes como la Web Semántica, la Inteligencia Artificial y el Internet de las Cosas.



Fuente: <https://gic.delaware.gov/web-standards/>

Estándares desarrollados por el W3C:

- El W3C ha desarrollado y mantiene una amplia gama de estándares que definen la forma en que se construye y se accede a la web. Algunos de los estándares más destacados incluyen:
- **HTML (HyperText Markup Language):** Es el lenguaje de marcado utilizado para crear la estructura y el contenido de las páginas web. El W3C ha publicado varias versiones de HTML, y la última versión es HTML5.
- **CSS (Cascading Style Sheets):** Es un lenguaje de estilo utilizado para controlar la presentación y el diseño de las páginas web. CSS permite separar el contenido del diseño y facilita la creación de sitios web con un aspecto visualmente atractivo y consistente.

WCAG (Web Content Accessibility Guidelines): Son pautas que proporcionan recomendaciones para hacer que los sitios web sean accesibles para personas con discapacidades.

1.2 Ejemplos de Navegadores Web

Los navegadores web son aplicaciones que permiten a los usuarios acceder y navegar por el contenido de Internet. Son herramientas esenciales para explorar sitios web, interactuar con aplicaciones en línea y consumir contenido en la World Wide Web.

Google Chrome:

Google Chrome, desarrollado por Google, es uno de los navegadores web más populares y ampliamente utilizados en el mundo. Se caracteriza por su velocidad, simplicidad y compatibilidad con estándares web.

Mozilla Firefox:

Mozilla Firefox es otro navegador web muy popular conocido por su enfoque en la privacidad y la seguridad. Firefox es un proyecto de código abierto y se centra en proporcionar una experiencia de navegación rápida y personalizable.

Safari:

Safari es el navegador web predeterminado en los dispositivos Apple, incluyendo iPhones, iPads y Macs. Ofrece una experiencia de navegación rápida y fluida, así como una integración estrecha con otros productos y servicios de Apple.

Microsoft Edge:

Microsoft Edge es el navegador web desarrollado por Microsoft y es el sucesor de Internet Explorer. Edge se basa en el motor de renderizado Blink (el mismo motor utilizado por Google Chrome), Edge se destaca por su integración con servicios de Microsoft, como Cortana y Microsoft 365.

Opera:

Opera es un navegador web conocido por su enfoque en la velocidad y la eficiencia. Ofrece características como bloqueo de anuncios integrado, VPN gratuita y un administrador de descargas incorporado. Opera también proporciona un conjunto de funciones personalizables y una interfaz de usuario elegante.

Brave:

Brave es un navegador web relativamente nuevo que se enfoca en la privacidad y la seguridad del usuario. Basado en el proyecto de código abierto Chromium, Brave bloquea anuncios y rastreadores de forma predeterminada para mejorar la velocidad de navegación y proteger la privacidad.

Conclusión:

Existen numerosos navegadores webs disponibles en la actualidad, cada uno con sus propias características y enfoques. Los ejemplos mencionados en este informe, como Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, Opera y Brave, son solo una selección de los navegadores más populares y utilizados en la actualidad.

2. Integración de CMS + CRM + ERP

Por: Pavel Palma

2.1.1 Concepto de CMS

Es un Sistema de gestión de contenidos. Este es un sistema online que nos permite poner en marcha un sitio web de una forma practica y rápida.

Este como su nombre lo dice, esta tiene una posibilidad de poder administrar los contenidos de una forma sencilla, es decir, como mantener un blog, un ecommerce o cualquier otro tipo de página web con actualizaciones constantes.



Fuente: <https://negociosyestrategia.com/blog/cms-que-es-gestor-contenidos/>

2.1.2 Concepto de CRM

Es una herramienta que nos ayuda a permitir a una empresa para reunir en un mismo lugar todas las comunicaciones, documentaciones y las actividades que tienen con los usuarios, leads y los clientes, ya que puedan ser actuales o potenciales.

Esta plataforma es mas que un software, esta se convierte en una estrategia para poner en el centro al cliente y poderle garantizar la mejores de las experiencias.

CRM son las siglas en Ingles de "Costumer Relationship Management".



Fuente: https://www.siteground.es/blog/crm-que-es/#¿Que_es_un_CRM

2.1.3 Concepto de ERP

El significado de las siglas de ERP es Enterprise Resource Planning o en español decía "Sistema de Planificación de Recursos Empresariales".

El ERP nos sirve para poder hacer cargo de distintas operaciones internas de una empresa, este lo realiza desde su producción hasta su distribución. Este paquete ERP logra automatizar los procesos empresariales, logrando aumentar la productividad y así poder lograr reducir los costes. Los ERP suponen una gran inversión para las empresas, pues en una encuesta los resultados dieron de que las empresas que usaron el ERP, aumentan su productividad.



Fuente: <https://emiralfg.com/que-es-erp-que-significa-para-que-sirve/>

2.2.1 Historia del CMS

El CMS se creó en 1995, pues de principio se sacó su sistema de administración de documentos con Vignette, este sistema web, logró inspirar a que en 1999 apareciera el primer CMS llamado Xpedio. Pues este software ya se podía decir que era un CMS de extremo a extremo.

Al ver tal potencial que tenían las plataformas que la utilizaban, lo empezaron a utilizar otras páginas como Drupal (2001) y WordPress (2003).



Fuente: <https://adrianacmslucero.wordpress.com/historia/>

2.2.2 Historia del CRM

El CRM empezó en los años 1970, después que evaluaran la satisfacción de los clientes, en 1982 Kate y Robert D. introdujeron el concepto de marketing de base de datos, es decir que la aplicación de métodos estadísticos para analizar y recopilar datos del fondo de comercio.



Fuente: <https://www.smartcommerce21.com/blog/crm-un-poco-de-historia>

2.2.1 Historia del ERP

El ERP o software de gestión nace en 1960, entonces los desarrolladores informáticos del sector productivo desarrollaron programas básicos para monitorizar inventarios, balances... el origen era gestionar y controlar el inventario.

En 1990, el MRP o ERP ya es funcional para controlar inventarios, producciones, gestiones administrativas y RRHH.



Fuente: <https://www.terabyte2003.com/erp-origen-evolucion/#:~:text=El%20ERP%20o%20software%20de,gestionar%20y%20controlar%20el%20inventario.>

1.1 Integración CMS+CRM+ERP

Por: Wilmer Patzán

Un CMS es una herramienta que permite a sus administradores crear, editar y publicar contenido directamente en su sitio sin escribir ningún código de programación.

La integración de ERP es el método por el cual una empresa conecta su software ERP con otras aplicaciones y procesos especializados, en todo su negocio

1.1.1 Implementación CMS

Sistema de Gestión de Contenidos. Es un sistema online que nos permite poner en marcha un sitio web de forma práctica y rápida. Pero, no es únicamente eso, sino que su gran ventaja, como su nombre lo dice, es la posibilidad de administrar contenidos dinámicos de forma sencilla, es decir, mantener un blog, un ecommerce o cualquier otro tipo de página web

1.1.2 Implementación CRM

Es una herramienta estratégica que permite a una empresa reunir en un mismo lugar todas las comunicaciones, documentación y actividad que tiene con sus usuarios, leads y clientes, ya sean actuales o potenciales. Y consecuentemente, gestionar con facilidad y eficacia las acciones futuras. Por tanto, esta plataforma es mucho más que un software. Es una estrategia vital para poner en el centro al cliente y garantizarle la mejor de las experiencias.

1.1.3 Implementación ERP

Los sistemas ERP pueden tomar los datos del pedido en la propia oficina del cliente y generar la orden de fabricación, si correspondiere, o despachar la mercadería y enviar la factura. Teniendo la información en un único sistema, en lugar de dispersarse entre muchos diferentes que no se pueden comunicar entre si

1.1.4 Software de gestión

Un software de gestión es un sistema informático integrado por múltiples herramientas que individualmente se utilizan para ejecutar tareas administrativas, y que en conjunto, simplifica los procesos operativos, productivos y burocráticos de una organización. Un programa de gestión empresarial es una herramienta imprescindible para tareas como, por ejemplo, el control de la liquidez, imprescindible para llevar una buena gestión de tu empresa, poder pagar a los proveedores a tiempo y cumplir tus objetivos de negocio.

Tipos de software de gestión.

-Software de gestión a medida o predefinido

-Software de gestión en la nube o en local



Fuente: <https://www.agenciaeplus.com.br/es/o-que-e-headless-cms/>



Fuente: <https://enzyme.biz/blog/que-es-crm-en-marketing>



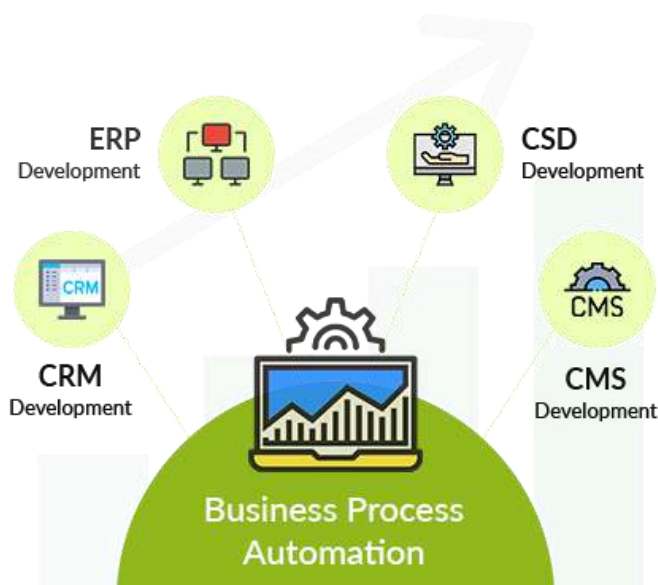
Fuente:

[https://www.semana.com/resizer/IFNYdMN353zQN3EV4Q9aMCPlwZw=/768x432/filters:format\(jpg\):quality\(50\)/cloudfront-us-east-1.images.arcpublishing.com/semana/HCRNYAZ4ZVD4DBSVO4NY6JDLUU.jpg](https://www.semana.com/resizer/IFNYdMN353zQN3EV4Q9aMCPlwZw=/768x432/filters:format(jpg):quality(50)/cloudfront-us-east-1.images.arcpublishing.com/semana/HCRNYAZ4ZVD4DBSVO4NY6JDLUU.jpg)

1.1 Integración CMS + CRM + ERP:

Por: Leonel de Jesús Erasmo Pérez Lúcas

La integración de un sistema de gestión de contenido (CMS, por sus siglas en inglés), un sistema de gestión de relaciones con el cliente (CRM) y un sistema de planificación de recursos empresariales (ERP) puede brindar numerosos beneficios a una organización al permitir la gestión eficiente de contenido, clientes y procesos empresariales en un solo entorno. Sin embargo, también puede presentar desafíos y desventajas que es importante considerar. A continuación, se presentan algunas ventajas y desventajas de la integración de CMS, CRM y ERP:



Fuente: <https://www.witsdigital.com/services/software-development>

1.1.1 Ventajas

- **Mejora de la eficiencia operativa:** La integración de estos sistemas permite una mayor automatización y optimización de los procesos empresariales, lo que puede mejorar la eficiencia operativa de una organización. Los datos se comparten entre los sistemas, evitando la duplicación de esfuerzos y reduciendo los errores manuales.
- **Mayor visibilidad y colaboración:** Al tener un sistema unificado, se facilita la colaboración entre departamentos y se mejora la visibilidad de los datos y actividades relacionadas con clientes, contenido y operaciones empresariales. Esto puede permitir una toma de decisiones más informada y una mejor comunicación interna.
- **Personalización y experiencia del cliente mejoradas:** Al combinar los datos del CRM y el CMS, se puede lograr una mayor personalización y proporcionar una experiencia del cliente más relevante. Los datos del CRM pueden ayudar a comprender mejor las necesidades y preferencias del cliente, mientras que el CMS permite ofrecer contenido personalizado en función de esos datos.
- **Mejor gestión de inventario y recursos:** La integración del ERP con el CMS y el CRM facilita una gestión más eficiente de inventarios y recursos empresariales. Se puede monitorear y administrar el inventario en tiempo real, lo que ayuda a evitar la escasez o el exceso de stock y optimiza la planificación de recursos.

1.1.2 Desventaja

- **Complejidad y costo de implementación:** La integración de CMS, CRM y ERP puede ser un proceso complejo y costoso. Requiere una planificación cuidadosa, desarrollo personalizado y configuración adecuada para asegurar que los sistemas funcionen juntos de manera efectiva. Además, puede ser necesario invertir en capacitación del personal para utilizar los sistemas integrados correctamente.
- **Posibles conflictos y problemas de compatibilidad:** La integración de sistemas puede dar lugar a problemas de compatibilidad o conflictos entre los diferentes sistemas. Esto puede resultar en la pérdida de datos o en la necesidad de realizar ajustes y personalizaciones adicionales para que los sistemas funcionen correctamente.
- **Dependencia de proveedores y actualizaciones:** Al integrar varios sistemas, es probable que se dependa de múltiples proveedores de software y se requieran actualizaciones periódicas de cada sistema. Esto implica la necesidad de coordinar las actualizaciones y puede generar complicaciones si uno de los proveedores no es compatible con las actualizaciones de los otros sistemas.

1.1.3 Seguridad Web

La seguridad en la web es un aspecto crítico cuando se integran CMS, CRM y ERP, ya que implica el manejo de datos sensibles de clientes y la gestión de recursos empresariales. Aquí hay algunos puntos importantes a tener en cuenta:

- **Protección de datos:** Es esencial contar con medidas de seguridad adecuadas para proteger la información confidencial de los clientes y los datos empresariales. Esto implica el uso de cifrado, autenticación de usuarios, control de acceso y auditorías periódicas.
- **Actualizaciones y parches:** Mantener los sistemas actualizados con las últimas correcciones de seguridad es fundamental para protegerse contra posibles vulnerabilidades conocidas. Esto incluye aplicar parches de seguridad y actual



Fuente: <https://www.techtitute.com/informatica/experto-universitario/experto-seguridad-informatica-comunicaciones>

2. Integración de CMS + CRM + ERP

Por: Krystel Adriany

2.7.1 Historia del CMS

El origen de los CMS se remonta a mediados de los años 90, cuando aparecen los primeros sistemas que permiten publicar contenido desde un ordenador personal. Se empieza a hablar de la gestión de contenidos desde Internet y en 1995 la empresa RedDot presenta un sistema gestor de contenidos basado en una base de datos.

2.7.2 Historia del CRM

El concepto de CRM comenzó a principios de la década de 1970, cuando se evaluó la satisfacción del cliente mediante encuestas anuales. En 1986, Pat Sullivan y Mike Muhney lanzaron un sistema de evaluación de clientes llamado ACT! que ofreció un servicio de gestión de contactos por primera vez. Fue entonces cuando Tom Siebel diseñó lo que se considera el primer CRM en 1993. Siebel presentó la primera aplicación móvil de CRM llamada Siebel Sales Handheld en 1999.

2.7.3 Historia del ERP

El ERP o software de gestión nace en 1960, entonces los desarrolladores informáticos del sector productivo desarrollaron programas básicos para monitorizar inventarios, balances... el origen era gestionar y controlar el inventario. En 1990, el MRP o ERP ya es funcional para controlar inventarios, producciones, gestiones administrativas y RRHH.

2.8.1 Vulnerabilidades del CMS

Una simple vulnerabilidad sirve de estímulo para lanzar ataques de gran escala. Con estos ataques masivos, pueden hacer daño de forma potencial en muchas webs. Algunas de estas vulnerabilidades podrían ser:

- Envío de parámetros sin validar o sin codificar.
- Errores en control de acceso
- Mal uso de gestión de Sesiones
- Cross-site scripting (XSS) erróneos
- Desbordamiento de búfer
- Comandos equivocados o con un mal funcionamiento

2.8.2 Vulnerabilidades del CRM

Pese a que los beneficios obtenidos ganan por mucho a los inconvenientes presentados por este sistema estas son algunas vulnerabilidades presentadas en el software:

- Gastos del software y del equipo técnico
- Resistencia a su uso
- Es posible que no se adapte a todos los negocios
- Cumplir con la Ley de Protección de Datos
- Requiere de capacitación para un uso adecuado por parte de empleados/administradores

2.8.3 Vulnerabilidades del ERP

Los sistemas ERP están destinados a minimizar los riesgos, pero a veces crean riesgos propios, algunos ejemplos de estos riesgos/vulnerabilidades podrían ser los siguientes:

- Tienen más de una plataforma ERP
- Existen múltiples instancias individuales de ERP
- La configuración de control de la aplicación no está activada
- Los plazos de implementación hicieron que se pasaran por alto algunos controles
- Intentos deliberados de evitar los controles
- Errores de entrada de datos

Anexos



Fuente: https://mixcore.org/assets/img/CMS_c3xocc.jpg



Fuente: <https://cdn2.hubspot.net/hubfs/2177779/crm.png>



Fuente: <https://bap-software.net/wp-content/uploads/2020/08/erp-package.jpg>

2. Integración CMS, CMR, ERP

Por: Rodrigo Ramírez.

El CMS es “el sistema de gestión de contenidos”, permite mantener el contenido de tu sitio web por uno mismo, permite añadir, actualizar y eliminar los textos e imágenes de la web de forma sencilla y segura.

El CMR es la “gestión de las relaciones con los clientes”, gestiona los datos internos, herramientas como los equipos de atención al clientes, marketing y ventas para realizar un seguimiento de los clientes.

Y el ERP es la “planificación de recursos empresariales”, reúne todo, la planificación, el proceso de fabricación, la gestión de inventarios, el envío, la nómina, los acreedores y el nombre, permite tomar decisiones más informadas.

Componentes del Marketing digital.



Fuente: <https://esquio.es/wp-content/uploads/2022/11/elementos-marketing-digital.jpg>

2.9 Soluciones

En el **CRM** podemos hallar a Microsoft Dynamics 365, donde ofrece: Planificación para órdenes de servicio, asignación y activación de recursos empresariales.

Gestione de forma remota y consulte los detalles de facturación e historiales de clientes.

Con el **ERP** se ofrece un sistema por **harperp**, donde se pueden generar estados financieros multi empresa en la nube y en tiempo real.

Integra, organiza y administra desde harpERP cada uno de los procesos de su negocio

Y para el **CMS** tenemos variados sistemas como WordPress, TYPO3, ¡Joomla!, Drupal, Contao y Neos están consideradas como las soluciones estándar para la administración profesional de sitios web.

Diferencias entre

Punto de Venta:	Sistema ERP:	Sistema CRM:
<ul style="list-style-type: none">• Operaciones básicas de una tienda.• Manejo de promociones.• Roles de empleados.• Registro de ventas.	<ul style="list-style-type: none">• Gestión de proyectos.• Recursos Humanos.• Contabilidad.• Reducción de costos.• Producción.• Facturación.• Cobranza.• Inventarios.	<ul style="list-style-type: none">• Atención y servicio al cliente.• Seguimiento de clientes.• Call Center.

SAP Logo | www.visualkgroup.com | VisualK Group Logo

Fuente: <https://visualkgroup.com/wp-content/uploads/2019/11/INFOGRAFIA-VK-CMRERPPV.png>

2.10 Tipos de Ataques

En los ataques podemos encontrar diferentes causas, pero la mayoría son para robar o causar algún tipo de estafa, algunos ataques podrían ser:

Phishing: consiste en correos electrónicos o mensajes de texto que aparentemente son enviados por fuentes confiables y que persuaden al destinatario a completar una acción.

Whalling: Están dirigidos a perfiles directivos como CEO's o CFO's y otros cargos altos de las organizaciones con el objetivo de robarles información confidencial a la que ellos tienen acceso.

Ransomware: secuestro de datos consiste en el bloqueo, por parte de un hacker, de un dispositivo electrónico.



Fuente: <https://www.piranirisk.com/hubfs/Ataques-ciberneticos-causas-tipos-y-consecuencias.webp>

1.1 Arquitectura y seguridad de las aplicaciones WEB

Por: Jose Gabriel Ramos Perez

La arquitectura web se conforma de las interfaces de usuario, las plataformas de middleware y las bases de datos son solo algunos ejemplos de los elementos de aplicación que conforman una arquitectura de aplicación web, que es un marco que describe cómo estos componentes interactúan y se relacionan entre sí. Consiste en una serie de elementos y una descripción de cómo interactúan lógicamente.

1.1.1 Que es un software web

El software web es cuando hablamos de servidores, ya sea que pertenezcan a una persona o una empresa que ofrece servicios de alojamiento, y el software que se instala en ellos, estamos hablando de máquinas que se utilizan para brindar servicios y realizar tareas para otras computadoras.

1.1.2 Usos del software Web

Puede realizar una variedad de tareas en varios sitios web, hablar con clientes o compañeros de trabajo, realizar cálculos, llevar a cabo procesos, intercambiar información y ofrecer servicios de forma remota. El software web es utilizado por todos estos programas en línea.

1.1.3 Porque Elegir una solución web

El hecho de que las aplicaciones web funcionen de manera similar independientemente de la instalación de la versión del sistema operativo del cliente es un beneficio significativo. La aplicación web se crea una vez y se ejecuta de manera uniforme en todos los sistemas operativos en lugar de tener que crear clientes para Windows, Mac OS X, GNU/Linux y otras plataformas.



Fuente: <https://blog.desdelinux.net/wp-content/uploads/2019/10/servidores-web-programas-libres-abiertos-imagen-contenido-blog-desdelinux-830x459.jpg>

1.1 Arquitectura de seguridad y de las aplicaciones web

Por: **Fernando Jose Raxon P.**

La arquitectura de seguridad y de las aplicaciones web es un enfoque estructurado para diseñar y desarrollar aplicaciones web seguras. Se centra en la protección de los datos sensibles, la prevención de ataques y la mitigación de riesgos en el contexto de las aplicaciones web.

La arquitectura de seguridad de las aplicaciones web se basa en varios principios y componentes clave que ayudan a garantizar la protección de la aplicación y de los datos de los usuarios. Algunos de los elementos fundamentales de una arquitectura de seguridad de aplicaciones web

1.1.1 Autenticación y autorización

Estos mecanismos se utilizan para verificar la identidad de los usuarios y permitirles acceder a los recursos adecuados. Se pueden utilizar técnicas como contraseñas, tokens de sesión, autenticación de dos factores, entre otros.

1.1.2 Gestión de sesiones

Es importante establecer y administrar sesiones seguras para los usuarios que acceden a la aplicación. Esto implica utilizar técnicas como tokens de sesión seguros, expiración de sesiones inactivas, renovación de tokens, entre otros.

1.1.3 Protección contra ataques

La arquitectura de seguridad debe incluir medidas para protegerse contra diversos ataques, como inyección de código SQL, ataques de cross-site scripting (XSS), falsificación de solicitudes entre sitios (CSRF) y otros. Esto puede implicar el uso de validación de datos de entrada, sanitización de datos, protección contra scripting malintencionado, entre otros.

1.1.4 Auditorías y registros

Es importante llevar un registro de las actividades de la aplicación web, incluyendo registros de acceso, registros de errores y otros eventos relevantes. Esto ayuda a identificar posibles brechas de seguridad y permite una respuesta rápida en caso de incidentes.

Estos son solo algunos aspectos clave de la arquitectura de seguridad y de las aplicaciones web. Es importante tener en cuenta que la seguridad es un proceso continuo y en constante evolución, por lo que es fundamental seguir las mejores prácticas de seguridad y mantenerse actualizado sobre las nuevas amenazas y soluciones de seguridad.

1. Arquitectura y seguridad de las Aplicaciones WEB

3.7 ¿Dónde Encontrarlos?

La seguridad de las aplicaciones web se encuentra en varios niveles y áreas. Aquí tienes algunos puntos clave donde se enfoca la seguridad de las aplicaciones web:

- **Desarrollo seguro:** La seguridad debe incorporarse desde el inicio del proceso de desarrollo de una aplicación web. Esto implica seguir buenas prácticas de codificación, como la validación de entradas y el manejo adecuado de errores, así como utilizar marcos y bibliotecas seguras.
- **Autenticación y autorización:** Es importante implementar un sistema de autenticación robusto para verificar la identidad de los usuarios. Esto puede incluir contraseñas seguras, autenticación de dos factores y medidas adicionales como la autenticación biométrica.
- **Protección contra ataques de inyección:** Las aplicaciones web deben protegerse contra ataques de inyección, como la inyección de SQL y la inyección de comandos.
- **Manejo seguro de sesiones:** Las sesiones de los usuarios deben manejarse de manera segura para evitar ataques como el secuestro de sesiones.
- **Protección contra ataques de cross-site scripting (XSS):** Los ataques XSS son comunes en aplicaciones web. Para protegerse contra ellos, se deben aplicar técnicas de filtrado y validación de entrada.
- **Protección contra ataques de cross-site request forgery (CSRF):** Para prevenir los ataques CSRF, las aplicaciones web deben utilizar tokens CSRF y verificar el origen de las solicitudes para asegurarse de que provengan de fuentes legítimas.
- **Pruebas de seguridad:** Es fundamental realizar pruebas regulares de seguridad en las aplicaciones web para identificar posibles vulnerabilidades.
- **Actualizaciones y parches:** Mantener las aplicaciones web actualizadas con las últimas correcciones de seguridad es crucial para garantizar su seguridad continua.

3.8 Tipos de Licencia

La seguridad de las aplicaciones web no se rige por licencias específicas, sino que se basa en prácticas y estándares de seguridad establecidos. Sin embargo, existen diferentes herramientas y tecnologías relacionadas con la seguridad de las aplicaciones web que pueden tener sus propias licencias.

Software de seguridad de aplicaciones web: Existen diversas soluciones y herramientas de seguridad específicas para proteger las aplicaciones web, como firewalls de aplicaciones web (WAF), sistemas de gestión de vulnerabilidades y análisis estático y dinámico de código.

Bibliotecas y marcos de seguridad: Muchos lenguajes de programación y marcos de desarrollo web cuentan con bibliotecas y componentes específicos para la seguridad de las aplicaciones web.

Certificaciones y estándares de seguridad: Existen certificaciones y estándares reconocidos en el ámbito de la seguridad de las aplicaciones web, como el estándar OWASP (Open Web Application Security Project), que proporciona pautas y mejores prácticas para desarrollar aplicaciones web seguras.

3.9 Ventajas y Desventajas de un software serial

Ventajas

- **Control de acceso:** Un software serial permite un control preciso sobre quién puede acceder y utilizar el software. Al requerir una clave de licencia única, se puede limitar el acceso solo a los usuarios autorizados.
- **Generación de ingresos:** Si eres el desarrollador o propietario del software, el uso de licencias seriales puede generar ingresos al vender las licencias a los usuarios. Esto puede ser especialmente beneficioso en el caso de software comercial o con funcionalidades avanzadas.
- **Actualizaciones y soporte:** Al contar con licencias seriales, puedes ofrecer actualizaciones y soporte técnico a los usuarios que adquirieron una licencia válida. Esto permite mantener el software actualizado y brindar asistencia en caso de problemas o preguntas.
- **Protección contra piratería:** El uso de licencias seriales puede ayudar a proteger el software contra la piratería y el uso no autorizado. Al requerir una clave de licencia única, se dificulta la distribución ilegal del software.

Desventajas

- **Complejidad de administración:** La gestión de licencias seriales puede ser complicada y requerir un sistema adecuado para su administración. Esto implica la emisión, seguimiento y validación de las licencias.
- **Barrera para los usuarios:** El uso de licencias seriales puede generar una barrera para los usuarios que deben adquirir una licencia para acceder al software. Esto puede limitar la adopción del software, especialmente si hay alternativas gratuitas o de código abierto disponibles.
- **Posible obsolescencia:** En algunos casos, el uso de licencias seriales puede llevar a problemas de obsolescencia. Si el software requiere una activación en línea o una comunicación constante con un servidor de licencias, puede volverse inutilizable si el servidor deja de estar disponible o si la empresa que emitió las licencias ya no está en funcionamiento.
- **Dificultad para probar el software:** Al requerir una licencia para utilizar el software, puede dificultar la evaluación y prueba del mismo antes de adquirir una licencia. Esto puede ser una desventaja para los usuarios que deseen evaluar el software antes de tomar una decisión de compra.

3.10 Software con servicios Rentados

También conocidos como Software as a Service (SaaS), son aplicaciones de software que se entregan a los usuarios a través de internet como un servicio. En lugar de adquirir una licencia y descargar el software para instalarlo en sus propios dispositivos o servidores, los usuarios acceden al software a través de una conexión a internet y lo utilizan directamente desde el proveedor de servicios.

En un modelo SaaS, el proveedor del servicio es responsable de alojar, mantener y actualizar el software, así como de gestionar la infraestructura subyacente. Los usuarios generalmente pagan una tarifa periódica, como una suscripción mensual o anual, para acceder y utilizar el software.

1.1 Seguridad de aplicaciones web

Por: Ariel Ridriguez

La implementación de la seguridad en las aplicaciones web es un proceso que implica múltiples pasos y consideraciones. A continuación, te proporcionaré una descripción general de cómo suele llevarse a cabo:

Evaluación de riesgos: Antes de implementar la seguridad en una aplicación web, es importante realizar una evaluación de riesgos para identificar las posibles vulnerabilidades y amenazas. Esto implica analizar el entorno de la aplicación, identificar los activos valiosos, evaluar las posibles brechas de seguridad y determinar las posibles consecuencias de una violación de seguridad.

Diseño seguro: Una vez que se han identificado los riesgos, se debe diseñar la aplicación de manera segura. Esto incluye considerar aspectos como la autenticación de usuarios, el control de acceso, la gestión de sesiones, la protección de datos sensibles y la prevención de ataques comunes como la inyección de código SQL, los ataques de cross-site scripting (XSS) y los ataques de falsificación de solicitudes entre sitios (CSRF)

1.1.1 Software de desarrollo a medida

El software de desarrollo a medida se refiere a la creación de software específicamente diseñado para satisfacer las necesidades de una empresa o cliente en particular. A diferencia del software comercial o de paquetes estándar, que está diseñado para ser utilizado por una amplia gama de usuarios, el desarrollo a medida implica la creación de una solución personalizada que se adapta a los requisitos específicos de una organización.

El proceso de desarrollo a medida implica analizar detalladamente los requisitos del cliente, comprender su flujo de trabajo y las necesidades particulares de su negocio. Con esta información, los desarrolladores diseñan, crean y prueban el software desde cero, asegurándose de que cumpla con todas las especificaciones y requisitos definidos. Esto implica escribir código personalizado, configurar bases de datos, integrar sistemas existentes y, en general, crear una solución completamente adaptada a los objetivos y procesos de la empresa.

1.1.2 Ciclos de vida del Software

Los ciclos de vida del software se refieren a las distintas etapas por las que pasa un proyecto de desarrollo de software, desde su concepción inicial hasta su retirada o finalización. Estas etapas están diseñadas para gestionar de manera efectiva el proceso de desarrollo, desde la planificación y diseño hasta la implementación, pruebas, despliegue y mantenimiento del software. Hay varios modelos de ciclos de vida del software, pero el más comúnmente utilizado es el modelo en cascada y el modelo en espiral.

Modelo en cascada: Es un enfoque secuencial en el que cada etapa del ciclo de vida se lleva a cabo de manera lineal y secuencial, sin superposición. Las etapas típicas incluyen requisitos, diseño, implementación, pruebas y mantenimiento. Una vez que se completa una etapa, se pasa a la siguiente y no se permite volver atrás. Este modelo es adecuado para proyectos en los que los requisitos son estables y bien entendidos desde el principio.

Modelo en espiral: Este modelo combina elementos del enfoque lineal y la metodología iterativa. Se basa en la idea de realizar múltiples iteraciones a través de los diferentes pasos del ciclo de vida, lo que permite una mayor flexibilidad y adaptación a medida que se obtiene retroalimentación del cliente o usuario final. Cada iteración pasa por las etapas de identificación de riesgos, análisis y resolución de problemas, desarrollo y evaluación. Este modelo es adecuado para proyectos grandes y complejos, donde los riesgos y cambios son comunes.

1.1.3 Subtitulo

Desarrolladores de aplicaciones web: Los desarrolladores de software son responsables de implementar buenas prácticas de seguridad durante el proceso de desarrollo de la aplicación. Esto implica la correcta gestión de contraseñas, el uso de medidas de autenticación adecuadas, el filtrado y validación de datos de entrada, y la protección contra ataques comunes, como inyecciones de código SQL o XSS (Cross-Site Scripting).

Equipos de seguridad de la información: Las organizaciones a menudo tienen equipos dedicados a la seguridad de la información. Estos equipos se encargan de evaluar y mitigar los riesgos de seguridad en las aplicaciones web. Realizan pruebas de penetración, análisis de vulnerabilidades y monitoreo continuo para identificar y solucionar posibles problemas de seguridad.

Administradores de sistemas: Los administradores de sistemas son responsables de garantizar que los servidores y sistemas en los que se ejecutan las aplicaciones web estén configurados correctamente y actualizados con los últimos parches de seguridad. También se encargan de implementar medidas de seguridad a nivel de infraestructura, como firewalls y sistemas de detección de intrusiones.

Usuarios finales: Los usuarios finales también desempeñan un papel importante en la seguridad de las aplicaciones web. Es fundamental que los usuarios sigan buenas prácticas de seguridad, como el uso de contraseñas fuertes, no compartir información confidencial y estar alerta ante posibles ataques de phishing o ingeniería social.



Fuente:

<https://www.hostinger.es/tutoriales/wp-content/uploads/sites/7/2022/06/seguridad-aplicaciones-web.jpg>

1.1 Arquitectura y seguridad de las aplicaciones web

Por: Iker Rodolfo Rojas Reyes

La arquitectura y seguridad de las aplicaciones web se refiere a los principios y prácticas utilizados para diseñar, desarrollar y proteger aplicaciones web. Estos dos aspectos están estrechamente relacionados, ya que una arquitectura bien diseñada puede ayudar a mejorar la seguridad de una aplicación web.

La arquitectura de una aplicación web se refiere a la estructura y organización general de los componentes que la componen. Esto incluye el diseño de la interfaz de usuario, la lógica empresarial, la capa de acceso a datos y la infraestructura subyacente. Una arquitectura bien diseñada debe ser escalable, modular y fácil de mantener.

1.1.1 Tiempos de desarrollo

Los tiempos de desarrollo de aplicaciones web pueden variar significativamente según diversos factores, como la complejidad del proyecto, el tamaño del equipo de desarrollo, la experiencia del equipo, los requisitos del cliente y las tecnologías utilizadas. Sin embargo, puedo darte una idea general de los tiempos de desarrollo estimados para diferentes etapas del proceso:

- **Planificación y análisis:** Esta etapa implica comprender los objetivos del proyecto, definir los requisitos y realizar un análisis detallado. El tiempo dedicado a esta etapa puede variar, pero generalmente puede llevar de 1 a 4 semanas, dependiendo de la complejidad del proyecto.
- **Diseño:** En esta etapa, se crea el diseño visual de la aplicación web, incluyendo la estructura de la información, el diseño de la interfaz de usuario y la experiencia del usuario. El tiempo dedicado al diseño puede variar, pero suele tomar de 2 a 6 semanas.

1.1.2 Obsolescencia tecnológica o de desarrollo

La obsolescencia tecnológica o de desarrollo se refiere al fenómeno en el cual una tecnología o un enfoque de desarrollo se vuelve desactualizado, ineficiente o poco compatible con las nuevas necesidades y avances en el campo de la tecnología. Esto puede ocurrir por varias razones, incluyendo:

- **Avances tecnológicos:** A medida que surgen nuevas tecnologías y enfoques de desarrollo más eficientes y poderosos, los métodos y tecnologías antiguas pueden volverse obsoletos. Por ejemplo, en el desarrollo web, tecnologías como Flash o Silverlight han quedado obsoletas con la popularización de HTML5.
- **Cambios en los requisitos y estándares:** A medida que los requisitos de los usuarios y los estándares de la industria evolucionan, las tecnologías y enfoques de desarrollo existentes pueden volverse inadecuados para satisfacer estas nuevas demandas. Por ejemplo, la creciente necesidad de aplicaciones móviles ha llevado a un cambio en el enfoque de desarrollo hacia tecnologías y frameworks móviles.

1.1.3 Ejemplos de implementacion

Aplicaciones de comercio electrónico: Estas aplicaciones permiten a los usuarios realizar compras en línea. Pueden incluir características como catálogos de productos, carritos de compra, opciones de pago y seguimiento de pedidos.

Redes sociales: Las redes sociales en línea, como Facebook, Twitter o Instagram, son ejemplos populares de aplicaciones web. Permiten a los usuarios registrarse, conectarse con otros usuarios, compartir contenido, enviar mensajes y más.

Plataformas de blogs: Las plataformas de blogs, como WordPress o Blogger, permiten a los usuarios crear y publicar contenido en línea. Estas aplicaciones suelen tener características como la creación de publicaciones, la gestión de comentarios y la personalización de la apariencia del blog.

1. MYSQL

Por: Ronaldo Ruano

MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado y de código abierto. Fue desarrollado por primera vez en 1995 y ha ganado una gran popularidad debido a su facilidad de uso, rendimiento confiable y su comunidad activa de usuarios y desarrolladores.

1.1. Principales características de MySQL:

Modelo de datos relacional: MySQL se basa en el modelo de datos relacional, lo que significa que los datos se organizan en tablas compuestas por filas y columnas. Esto permite establecer relaciones entre diferentes tablas y realizar consultas complejas utilizando el lenguaje SQL (Structured Query Language).

Escalabilidad y rendimiento: MySQL es conocido por su capacidad de manejar grandes volúmenes de datos y un alto rendimiento en entornos de producción. Puede gestionar eficientemente cargas de trabajo tanto pequeñas como empresariales, y ofrece opciones de configuración para optimizar el rendimiento de acuerdo con las necesidades específicas.

Seguridad: MySQL ofrece diversas características de seguridad para proteger los datos almacenados. Esto incluye opciones de autenticación y control de acceso para garantizar que solo los usuarios autorizados puedan acceder y modificar los datos. También admite cifrado de datos para proteger la información confidencial.

Replicación y alta disponibilidad: MySQL permite la replicación de bases de datos, lo que significa que se pueden mantener copias exactas de la base de datos en diferentes servidores. Esto proporciona alta disponibilidad y tolerancia a fallos, ya que si un servidor falla, otro puede asumir su lugar sin interrupciones en el servicio.

Compatibilidad con estándares: MySQL es compatible con los estándares de SQL definidos por ANSI (American National Standards Institute) y otras organizaciones. Esto significa que las aplicaciones desarrolladas para MySQL pueden ser fácilmente migradas a otros sistemas de bases de datos compatibles con SQL.

1.2. Entorno de Trabajo.

El entorno de trabajo de MySQL se refiere al conjunto de herramientas y componentes utilizados para administrar, desarrollar y mantener una base de datos MySQL. Estos son algunos de los elementos clave del entorno de trabajo de MySQL:

Servidor MySQL: El servidor MySQL es el componente central de MySQL. Es el motor de base de datos que gestiona el almacenamiento, recuperación y manipulación de los datos. El servidor MySQL se ejecuta en un sistema operativo y proporciona servicios de bases de datos a través de una red o localmente.

1.2.1. MySQL Client:

MySQL proporciona una variedad de clientes que permiten interactuar con el servidor MySQL. Los clientes pueden ser herramientas de línea de comandos, interfaces gráficas de usuario o bibliotecas de programación que permiten desarrollar aplicaciones que se conectan al servidor MySQL. Algunos ejemplos populares de clientes MySQL son la línea de comandos de MySQL (`mysql`), MySQL Workbench y phpMyAdmin.

1.2.2. Lenguaje SQL:

SQL (Structured Query Language) es el lenguaje utilizado para interactuar con una base de datos MySQL. Permite realizar diversas operaciones, como crear, modificar y consultar bases de datos, tablas y registros. SQL proporciona comandos para realizar consultas, actualizaciones, inserciones y eliminaciones de datos, así como para definir estructuras de base de datos y establecer relaciones entre tablas.

1.2.3. Herramientas de administración:

MySQL ofrece una variedad de herramientas para administrar y mantener una base de datos. Estas herramientas incluyen el MySQL Shell, que es una interfaz de línea de comandos avanzada con características adicionales, como soporte para múltiples lenguajes de programación y capacidades de scripting. También está disponible MySQL Workbench, una herramienta gráfica de administración de bases de datos que permite realizar tareas como el diseño de esquemas, la ejecución de consultas y la monitorización del rendimiento.

1.2.4. Bibliotecas y APIs:

MySQL proporciona bibliotecas y APIs para una variedad de lenguajes de programación, lo que facilita el desarrollo de aplicaciones que se conectan y utilizan una base de datos MySQL. Algunas de las bibliotecas y APIs más populares son MySQL Connector/Python, MySQL Connector/J (para Java), MySQL Connector/NET (para .NET) y PHP MySQLi Extension (para PHP).

1.2.5. Documentación y recursos en línea:

MySQL cuenta con una amplia documentación oficial que cubre todos los aspectos de su uso y administración. Esta documentación incluye guías, tutoriales, referencias de comandos y ejemplos de código. Además, la comunidad de usuarios de MySQL es muy activa y ofrece foros de discusión, blogs y otros recursos en línea donde se pueden encontrar consejos, soluciones a problemas comunes y nuevas características.

1.3. Instalación.

La instalación de SQL, específicamente de un sistema de gestión de bases de datos como MySQL, puede variar dependiendo del sistema operativo que estés utilizando. A continuación, te proporcionaré una descripción general de los pasos generales para instalar MySQL en diferentes plataformas:

1.3.1. Windows:

Descarga el instalador de MySQL desde el sitio web oficial de MySQL.

Ejecuta el archivo de instalación descargado y sigue las instrucciones del asistente de instalación.

Durante el proceso de instalación, puedes elegir la configuración predeterminada o personalizar opciones como la ubicación de instalación, el tipo de instalación (servidor, cliente o ambos), y establecer una contraseña para el usuario root de MySQL.

Una vez completada la instalación, puedes acceder a MySQL a través de la línea de comandos de MySQL (utilizando el comando 'mysql' en la ventana de comandos) o mediante herramientas gráficas como MySQL Workbench.

1.3.2. macOS:

Descarga el instalador de MySQL desde el sitio web oficial de MySQL.

Ejecuta el archivo de instalación descargado y sigue las instrucciones del asistente de instalación.

Durante el proceso de instalación, puedes elegir la configuración predeterminada o personalizar opciones como la ubicación de instalación y establecer una contraseña para el usuario root de MySQL.

Una vez completada la instalación, puedes acceder a MySQL a través de la línea de comandos de MySQL (utilizando el comando 'mysql' en la Terminal) o mediante herramientas gráficas como MySQL Workbench.

1.3.3. Linux:

En la mayoría de las distribuciones de Linux, MySQL está disponible en los repositorios de paquetes y puede ser instalado usando el gestor de paquetes predeterminado.

Abre la terminal y actualiza la lista de paquetes disponibles ejecutando el comando correspondiente a tu distribución (por ejemplo, 'sudo apt update' en Ubuntu).

Luego, instala el paquete de MySQL usando el comando correspondiente a tu distribución (por ejemplo, 'sudo apt install mysql-server' en Ubuntu).

Durante el proceso de instalación, se te solicitará establecer una contraseña para el usuario root de MySQL.

Una vez completada la instalación, puedes acceder a MySQL a través de la línea de comandos de MySQL (utilizando el comando 'mysql' en la terminal) o mediante herramientas gráficas como MySQL Workbench.

1.4. Creación de Base de Datos.

La creación de una base de datos en MySQL implica seguir algunos pasos simples utilizando el lenguaje de consulta estructurado (SQL). A continuación, te mostraré cómo crear una base de datos en MySQL:

1.4.1. Conéctate al servidor MySQL:

Abre una terminal o una herramienta de línea de comandos y conecta al servidor MySQL utilizando el comando `mysql` seguido de las opciones de conexión, como el nombre de usuario y la contraseña. Por ejemplo:

```
...
```

```
mysql -u usuario -p
```

```
...
```

Esto te pedirá ingresar la contraseña del usuario especificado.

1.4.2. Crea la base de datos:

Una vez conectado al servidor MySQL, puedes crear una nueva base de datos utilizando el comando `CREATE DATABASE`. Por ejemplo, para crear una base de datos llamada "mi_basedatos", puedes ejecutar el siguiente comando:

```
...
```

```
CREATE DATABASE mi_basedatos;
```

```
...
```

1.4.3. Verifica la creación de la base de datos:

Para asegurarte de que se haya creado correctamente, puedes ejecutar el comando `SHOW DATABASES;`. Esto mostrará una lista de todas las bases de datos existentes en el servidor, incluyendo la que acabas de crear.

```
...
```

```
SHOW DATABASES;
```

```
...
```

1.4.4. Utiliza la base de datos:

Después de crear la base de datos, puedes comenzar a utilizarla mediante el comando `USE`. Por ejemplo:

```
...
```

```
USE mi_basedatos;
```

```
...
```

A partir de este punto, todos los comandos que ejecutes se aplicarán a la base de datos "mi_basedatos" a menos que especifiques lo contrario.

Esos son los pasos básicos para crear una base de datos en MySQL. Sin embargo, es importante tener en cuenta que existen muchas opciones y configuraciones adicionales que se pueden aplicar durante la creación, como establecer el conjunto de caracteres y

la clasificación predeterminados, definir opciones de almacenamiento, entre otros. Puedes consultar la documentación oficial de MySQL para obtener información más detallada sobre las opciones de creación de bases de datos específicas y cómo personalizarlas según tus necesidades.



Imagen: MySQL Logo

Fuente: <https://hoplasoftware.com/mysql-sistema-de-gestion-de-bases-de-datos-relacionales/>

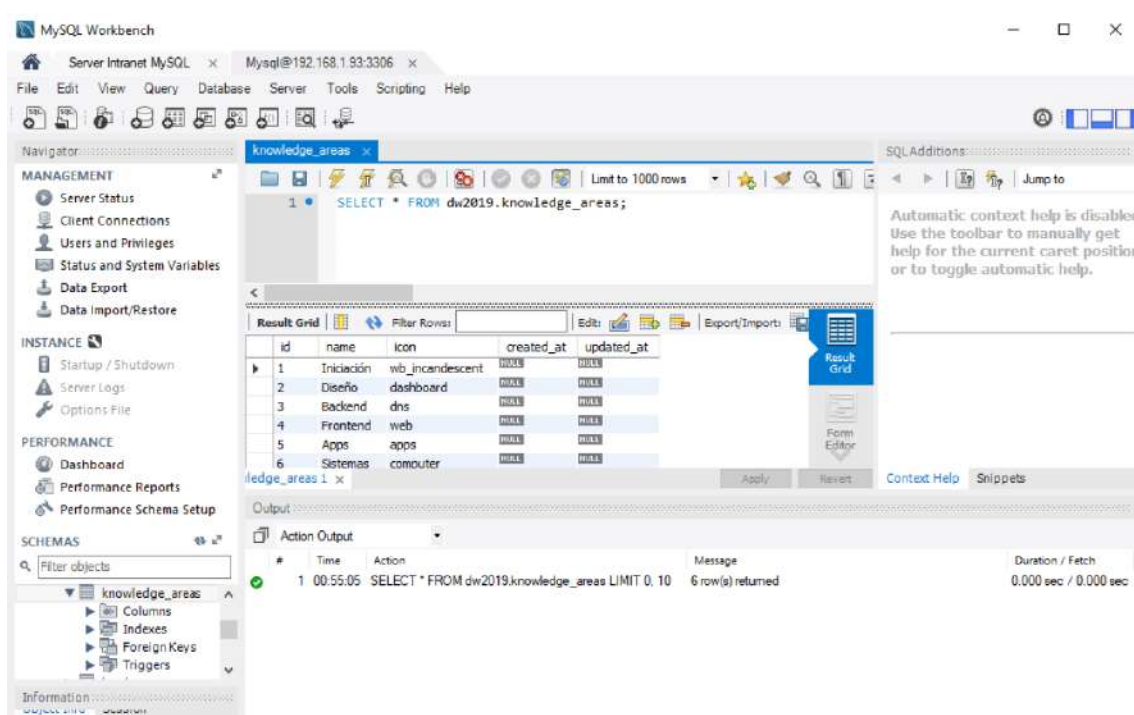


Imagen: Interfaz de Usuario de MySQL

Fuente: <https://www.arsys.es/blog/interfaces-graficas-mysql>



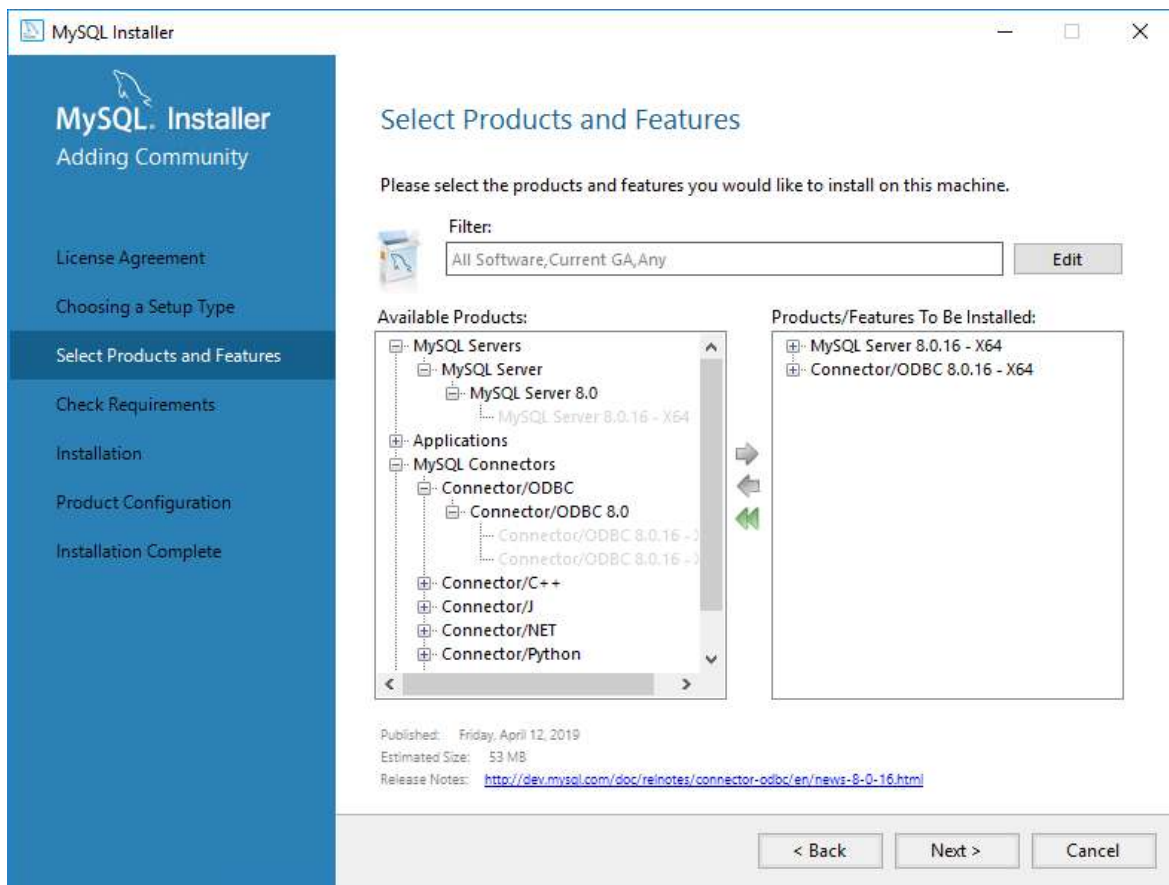


Imagen: Instalación de MySQL

Fuente: https://help.eset.com/protect_install/80/es-CL/mysql_windows.html

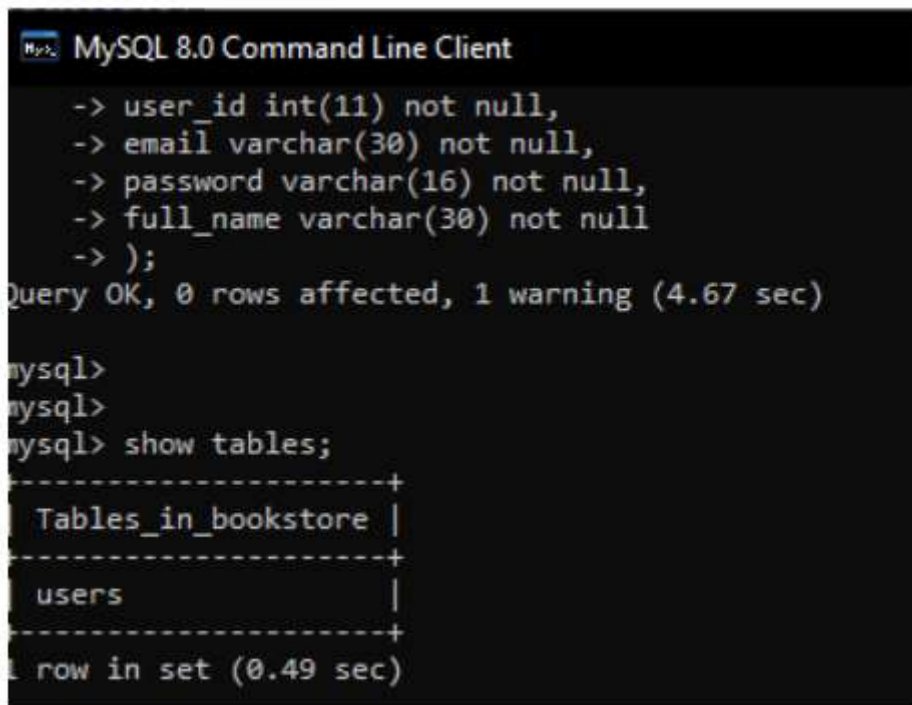


Imagen: Creación de Bases de Datos en MySQL

Fuente: <https://www.topcoder.com/thrive/articles/creating-a-database-using-mysql-command-line-client>

1.1 Mostrar base de datos y tablas

Por: **Gabriela Santos**

Para mostrar las bases de datos en MySQL desde la consola, puedes seguir los siguientes pasos:

- Abre una terminal o línea de comandos.
- Inicia sesión en MySQL como usuario con privilegios suficientes para acceder a las bases de datos. Puedes usar el siguiente comando y luego ingresar tu contraseña cuando se solicite "mysql -u tu_usuario -p".
- Una vez que hayas iniciado sesión, se abrirá la consola de MySQL. Ahora puedes ejecutar el siguiente comando para ver todas las bases de datos disponibles "SHOW DATABASES;".
- Esto mostrará una lista de las bases de datos existentes en el servidor de MySQL.
- Para seleccionar una base de datos específica y trabajar con ella, utiliza el comando USE seguido del nombre de la base de datos. Por ejemplo "USE nombre_de_la_base_de_datos;"
- Una vez que hayas seleccionado una base de datos, puedes ejecutar consultas y realizar otras operaciones relacionadas con ella.
- Recuerda que es importante tener los permisos adecuados para ver y acceder a las bases de datos en MySQL. Si encuentras algún problema de acceso, asegúrate de tener los privilegios necesarios o consulta con el administrador del sistema.

1.1.1 Tipos de datos básicos

En MySQL, existen varios tipos de datos básicos que puedes utilizar para definir la estructura de tus tablas. A continuación, te menciono algunos de los tipos de datos más comunes:

- **INTEGER:** Es utilizado para almacenar números enteros. Puedes especificar la longitud opcionalmente, como INTEGER(10). También hay variantes como TINYINT, SMALLINT, MEDIUMINT y BIGINT, que representan enteros con diferentes rangos.
- **DECIMAL:** Se utiliza para almacenar números decimales precisos. Debes especificar la precisión y la escala, como DECIMAL(10,2), donde 10 es el total de dígitos y 2 es la cantidad de decimales.

- **FLOAT y DOUBLE:** Son tipos de datos que almacenan números de punto flotante con precisión limitada (FLOAT) o mayor precisión (DOUBLE). Puedes especificar la precisión opcionalmente, como `FLOAT(8,2)`, donde 8 es el total de dígitos y 2 es la cantidad de decimales.
- **CHAR y VARCHAR:** Se utilizan para almacenar cadenas de caracteres. CHAR tiene una longitud fija, mientras que VARCHAR tiene una longitud variable. Por ejemplo, `CHAR(10)` almacena siempre 10 caracteres, mientras que `VARCHAR(255)` puede almacenar hasta 255 caracteres.
- **DATE:** Almacena una fecha en formato 'YYYY-MM-DD'. Permite almacenar fechas desde el año 1000 hasta el año 9999.
- **TIME:** Almacena una hora en formato 'HH:MM:SS'.
- **DATETIME:** Combina una fecha y una hora en formato 'YYYY-MM-DD HH:MM:SS'. Puede almacenar fechas desde el año 1000 hasta el año 9999.
- **BOOLEAN:** Almacena valores booleanos, que pueden ser TRUE o FALSE. En MySQL, se representan como 1 (TRUE) o 0 (FALSE).

1.1.2 Valores Null

En MySQL, puedes insertar valores nulos en una tabla utilizando la palabra clave NULL. A continuación, te muestro cómo puedes insertar valores nulos en una tabla desde la consola de MySQL:

- Supongamos que tienes una tabla llamada usuarios con las columnas id, nombre y edad.
- Inicia sesión en MySQL desde la consola utilizando el comando `mysql -u tu_usuario -p`.
- Inserta los valores en la tabla utilizando el comando `INSERT INTO nombre_de_la_tabla (columna1, columna2, columna3) VALUES (valor1, valor2, valor3);`.

Si deseas insertar un valor nulo en alguna columna, simplemente puedes utilizar la palabra clave NULL en lugar de un valor específico. Por ejemplo `INSERT INTO usuarios (id, nombre, edad) VALUES (1, 'John Doe', NULL);`.

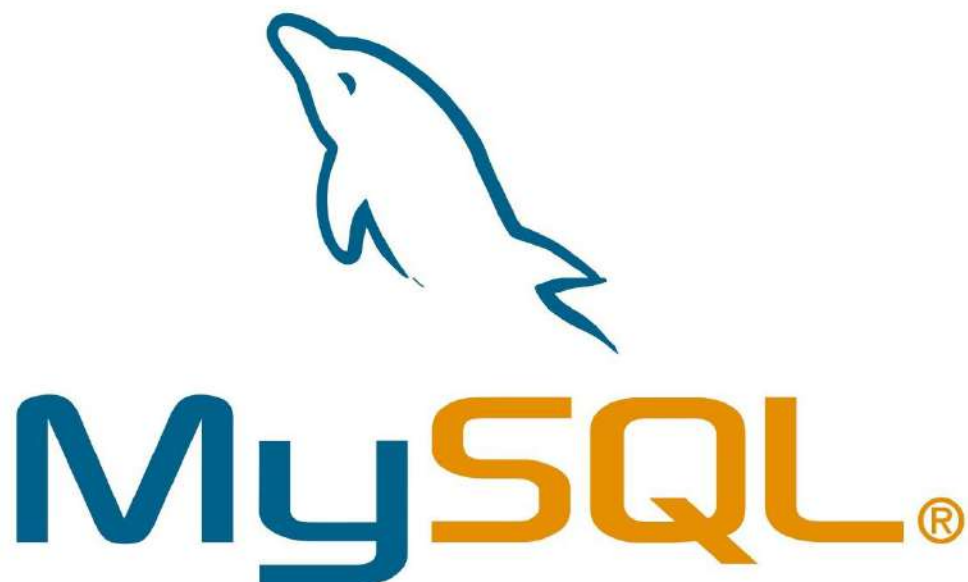


Imagen: Logo de MySQL

Fuente: <https://www.anerbarrena.com/wp-content/uploads/2016/05/mysql.jpg>

Tipo de campo	Tamaño de Almacenamiento
CHAR(n)	n bytes
VARCHAR(n)	n + 1 bytes
TINYBLOB, TINYTEXT	Longitud+1 bytes
BLOB, TEXT	Longitud +2 bytes
MEDIUMBLOB, MEDIUMTEXT	Longitud +3 bytes
LOBLOB, LONGTEXT	Longitud +4 bytes
ENUM ('value1', 'value2', ...)	1 ó dos bytes dependiendo del número de valores
SET('value1', 'value2', ...)	1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores

Imagen: Tipos de datos básicos

Fuente: <https://rahsuarez.files.wordpress.com/2010/07/mysqlcaracter.png>

1.1 DML

Por: Dilan Sarceño

Un lenguaje de manipulación de datos (Data Manipulation Language, o DML en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.

1.1.1 CREATE

Nos permite crear una tabla.

Sintaxis: `CREATE TABLE Nombre_Tabla (campo datatype, campo datatype);`

Ejemplo: `CREATE TABLE Alumnos (ID_Alumno INT, Nombre VARCHAR (50), Direccion VARCHAR (100));` –Creamos una tabla en sql llamada Alumnos con los campos ID_Alumno, Nombre y Dirección.

1.1.2 ALTER

Nos permite modificar la estructura de una tabla. suele ser usado para modificar las tablas agregando campos para este ejemplo se agregará un campo

Sintaxis: `ALTER TABLE Nombre_Tabla ADD campo datatype;`

Ejemplo: `ALTER TABLE Alumnos ADD Fecha_Nac DATETIME;` –Agregamos el campo en sql Fecha_Nac a la tabla Alumnos

1.1.3 DROP

Nos permite eliminar una tabla.

Sintaxis: `DROP TABLE Nombre_Tabla;`

Ejemplo: `DROP TABLE Alumnos;` –Eliminamos la tabla en sql Alumnos



Fuente: https://www.shutterstock.com/shutterstock/photos/1981460534/display_1500/stock-vector-dml-mean-data-manipulation-language-data-acronyms-letters-and-icons-vector-illustration-1981460534.jpg

1.1.4 TRUNCATE TABLE

Nos permite eliminar todos los registros de una tabla ver artículo.

Sintaxis: TRUNCATE TABLE Nombre_Tabla;

Ejemplo: TRUNCATE TABLE Alumnos; –Eliminamos todos los registros de la tabla Alumnos



Fuente: https://sites.google.com/site/sqlismysin/_/rsrc/1461169917292/config/customLogo.gif?revision=3

1.1 Lenguaje de Definición de Datos (DDL)

Por: Dulce María Shuyá Baten

El Lenguaje de Definición de Datos (DDL, por sus siglas en inglés) es una parte fundamental en la administración de bases de datos. Es utilizado para crear, modificar y eliminar la estructura de una base de datos, lo cual incluye la definición de tablas, columnas, índices, restricciones y otros elementos clave. El Lenguaje de Definición de Datos (DDL) es una parte integral de la administración de bases de datos, ya que permite la creación, modificación y eliminación de la estructura de la base de datos. Comprender los comandos básicos de DDL, así como la aplicación de restricciones, es esencial para diseñar y mantener bases de datos eficientes y confiables.

1.1.1 Select

La introducción al temario select sobre DDL abarca una comprensión profunda de los conceptos fundamentales necesarios para manipular la estructura de la base de datos de manera precisa y eficiente. A medida que los estudiantes se sumergen en este temario, se adentrarán en el fascinante mundo de la creación de tablas, donde aprenderán a diseñar y definir la estructura de las tablas, incluyendo columnas, tipos de datos y restricciones.

- **Modificación de tablas:** Se estudia la instrucción ALTER TABLE para realizar modificaciones en tablas existentes, como agregar o eliminar columnas, cambiar tipos de datos y restricciones.
- **Eliminación de tablas:** Se explora la instrucción DROP TABLE para eliminar tablas de una base de datos, junto con las precauciones necesarias antes de llevar a cabo esta acción.
- **Creación y gestión de índices:** Se enseña cómo crear índices en tablas para mejorar el rendimiento de las consultas, considerando diferentes tipos de índices.
- **Creación y gestión de vistas:** Se muestra cómo crear vistas, que son consultas guardadas como objetos, para facilitar el acceso a datos específicos de una tabla.
- **Otros objetos de la base de datos:** Se mencionan brevemente otros objetos como procedimientos almacenados, desencadenadores (triggers) y secuencias.
- **Consideraciones de seguridad:** Se destacan las consideraciones de seguridad al utilizar instrucciones DDL, los privilegios necesarios y las recomendaciones para proteger la estructura de la base de datos.

1.1.2 Insert

En el lenguaje de definición de datos (DDL) se utilizan diferentes instrucciones para crear y modificar la estructura de la base de datos, como la creación de tablas, índices y restricciones. Sin embargo, la instrucción INSERT no se considera típicamente parte del DDL, sino más bien una instrucción del lenguaje de manipulación de datos (DML). La instrucción INSERT se utiliza para agregar una nueva fila de datos a una tabla existente en una base de datos. Los valores se proporcionan para cada columna específica en la tabla en la que se está realizando la inserción. La instrucción INSERT también puede utilizarse para insertar múltiples filas a la vez, proporcionando conjuntos de valores separados por comas en la cláusula VALUES. Cada conjunto de valores debe estar encerrado entre paréntesis.

Algunas bases de datos pueden proporcionar extensiones o variantes de la instrucción INSERT que permiten realizar inserciones condicionales o insertar valores predeterminados en ausencia de valores proporcionados explícitamente.

1.1.3 Update

En el lenguaje de manipulación de datos SQL (Structured Query Language), existen dos categorías principales de comandos: DDL (Data Definition Language) y DML (Data Manipulation Language). Mientras que DDL se utiliza para definir y modificar la estructura de la base de datos, DML se enfoca en la manipulación de los datos almacenados en la base de datos. En este informe, nos centraremos en la función UPDATE en DDL.

Concepto de la función UPDATE en DDL:

La función UPDATE en DDL es un comando que permite modificar los datos existentes en una tabla de la base de datos. Se utiliza para realizar cambios en los valores de una o varias columnas de una o varias filas en una tabla. El comando UPDATE se encuentra dentro de la categoría DML, aunque su función principal es modificar los datos en lugar de extraerlos.

Sintaxis del comando UPDATE:

La sintaxis básica del comando UPDATE es la siguiente:

UPDATE nombre_tabla

SET columna1 = valor1, columna2 = valor2, ...

[WHERE condición];

WHERE condición (opcional): permite especificar una condición que debe cumplirse para que se realice la actualización. Si no se proporciona la cláusula WHERE, se modificarán todos los registros de la tabla.

1.1.4 Delete

En el lenguaje de definición de datos (DDL, por sus siglas en inglés), la función DELETE es utilizada para eliminar registros o filas de una tabla en una base de datos. Aunque comúnmente asociamos la instrucción DELETE con el lenguaje de manipulación de datos (DML), también existe una forma de utilizarla en el contexto del lenguaje de definición de datos. En este informe, exploraremos la función DELETE en DDL, sus características principales y cómo se utiliza.

La sentencia DELETE en el lenguaje de manipulación de datos se utiliza para eliminar filas de una tabla en una base de datos. A diferencia de las sentencias de DDL, como CREATE TABLE o ALTER TABLE, que se utilizan para definir o modificar la estructura de la base de datos, la sentencia DELETE se utiliza para manipular los datos dentro de la base de datos.

```

MySQL 5.6 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.6.12 MySQL Community Server (GPL)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT * FROM eli.publisher;
+-----+-----+-----+
| idpublisher | name   | phone |
+-----+-----+-----+
|          1 | edu4java | 4444  |
|          2 | eli     | 4343  |
+-----+-----+-----+
2 rows in set (0.01 sec)

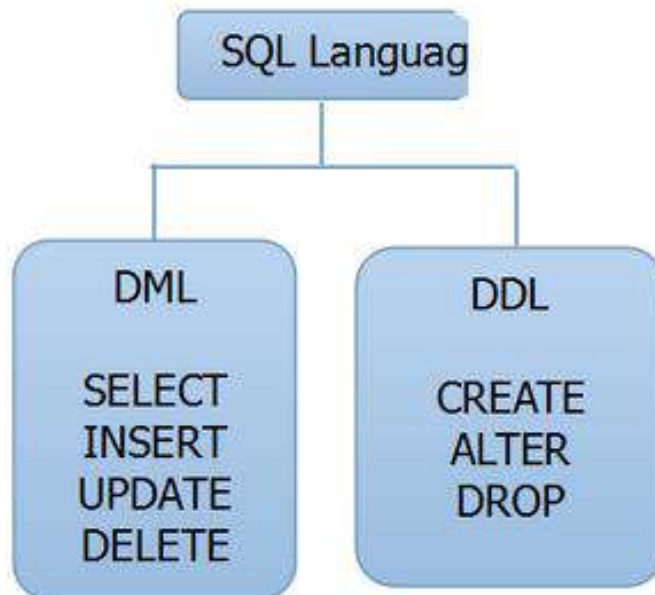
mysql> _
    
```

Fuente: <http://www.edu4java.com/img/sql/selectFromSql4.png>

```

INSERT INTO nombre_tabla (columna1, columna2, ...,
VALUES (valor1, valor2, ..., valorN);
    
```

Fuente: <http://www.edu4java.com/img/sql/selectFromSql4.png>



Fuente: <https://i.stack.imgur.com/2y9AY.png>

2. Framework Web y Scripting.

Por: Maryori Sipac.

2.1 Un framework web es un conjunto de herramientas y librerías predefinidas que proporcionan una estructura y funcionalidades comunes para el desarrollo de aplicaciones web. Estas herramientas y librerías permiten a los desarrolladores simplificar y agilizar el proceso de creación de aplicaciones web al proporcionar una base sólida sobre la cual construir. Un framework web suele incluir componentes como un enrutador (router) para gestionar las rutas de las páginas, un sistema de plantillas para facilitar la creación de la interfaz de usuario, una capa de abstracción de la base de datos para interactuar con la persistencia de datos, y funciones para el manejo de formularios, validación de datos, autenticación de usuarios, entre otros.

Los frameworks web también suelen seguir patrones y convenciones establecidas, lo que facilita el trabajo en equipo y la colaboración, ya que los desarrolladores pueden entender y trabajar con código escrito por otros en el mismo framework.



1 Framework web y Scripting

Por: Edgar Soc

1.1 Patrón MVC

El patrón MVC (Model-View-Controller) es un patrón de diseño ampliamente utilizado en el desarrollo de aplicaciones web y scripting. Este patrón se utiliza para separar la lógica de negocio de la interfaz de usuario, lo que permite un mayor modularidad, reutilización de código y mantenibilidad del sistema.

En el contexto de un framework web, el patrón MVC se aplica de la siguiente manera:

- **Modelo (Model):** El modelo representa los datos y la lógica de negocio de la aplicación. Es responsable de interactuar con la base de datos o cualquier otra fuente de datos, realizar cálculos y aplicar reglas de negocio. El modelo no tiene conocimiento sobre la interfaz de usuario o cómo se presentan los datos.
- **Vista (View):** La vista es responsable de la presentación de los datos al usuario. Puede ser una página HTML, una plantilla, una interfaz gráfica u otro medio de representación visual. La vista obtiene los datos necesarios del modelo y los muestra al usuario de forma adecuada.
- **Controlador (Controller):** El controlador actúa como intermediario entre el modelo y la vista. Recibe las interacciones del usuario a través de la interfaz de usuario y coordina la respuesta adecuada. El controlador consulta al modelo para obtener los datos necesarios, realiza cualquier procesamiento adicional y luego actualiza la vista con los resultados.

1.2 Model 2

El término "Model 2" se refiere a un estilo arquitectónico utilizado en el desarrollo de aplicaciones web y scripting. También conocido como Modelo-Vista-Controlador (MVC) basado en Servlets, el modelo 2 es una variante del patrón MVC adaptada específicamente para el entorno web.

En el modelo 2, los componentes principales son:

- **Controlador frontal (Front Controller):** Es el punto de entrada centralizado para todas las solicitudes entrantes. El controlador frontal es responsable de recibir las solicitudes del cliente, analizar la URL y determinar qué controlador debe manejarla. También se encarga de

realizar cualquier preprocesamiento necesario antes de enviar la solicitud al controlador adecuado.

- Controladores (Controllers): Los controladores son responsables de procesar las solicitudes entrantes y coordinar la lógica de negocio relacionada. Reciben los parámetros y datos de la solicitud, interactúan con el modelo para realizar operaciones en la base de datos o en otras fuentes de datos, y preparan los datos necesarios para la vista. Luego, seleccionan la vista apropiada y le pasan los datos para su presentación.
- Vistas (Views): Las vistas son responsables de la presentación de los datos al usuario. Pueden ser páginas HTML, plantillas o cualquier otro formato de presentación. Las vistas reciben los datos preparados por el controlador y los utilizan para generar la respuesta que se enviará al cliente.
- Modelo (Model): El modelo representa los datos y la lógica de negocio de la aplicación, similar a la descripción anterior del patrón MVC. El modelo puede incluir clases que interactúan con la base de datos, servicios web, APIs u otras fuentes de datos. Los controladores consultan el modelo para obtener o actualizar los datos necesarios y luego los envían a las vistas para su presentación.

En resumen, el modelo 2 es una adaptación del patrón MVC para el desarrollo web y scripting. Utiliza un controlador frontal para recibir las solicitudes, controladores para procesarlas, vistas para presentar los resultados y un modelo para gestionar los datos y la lógica de negocio.

4.1 Instalación de XAMPP

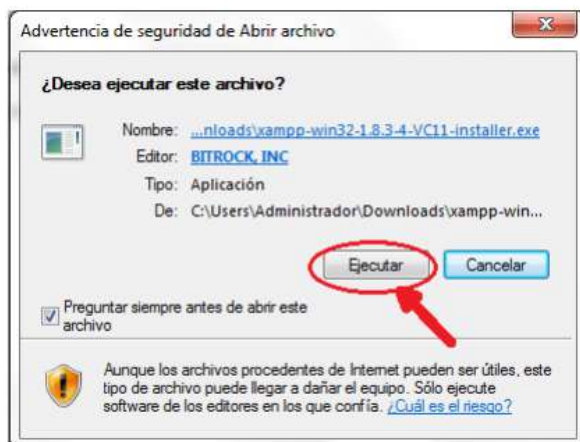
Por: Juan Alejandro Tiney Garcia

XAMPP es una pila AMP que significa multiplataforma: Contiene algunas herramientas de software administrativo adicionales como PHPMyAdmin (para acceso a la base de datos), servidor FTP FileZilla, servidor de correo Mercury y servidor JSP Tomcat.

- Ejecutar el archivo.exe



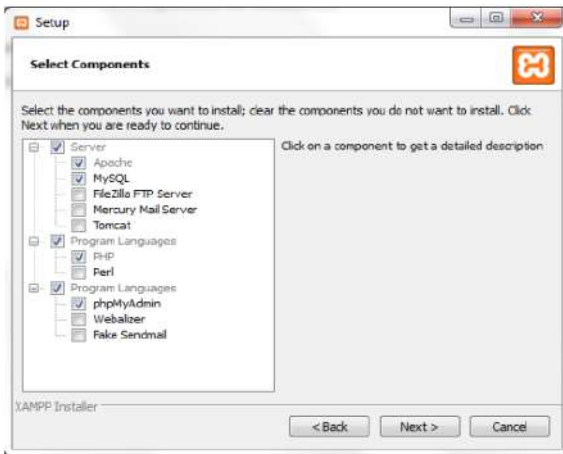
- Al hacer doble click sobre el ejecutable descargado, comenzará la instalación en sí.



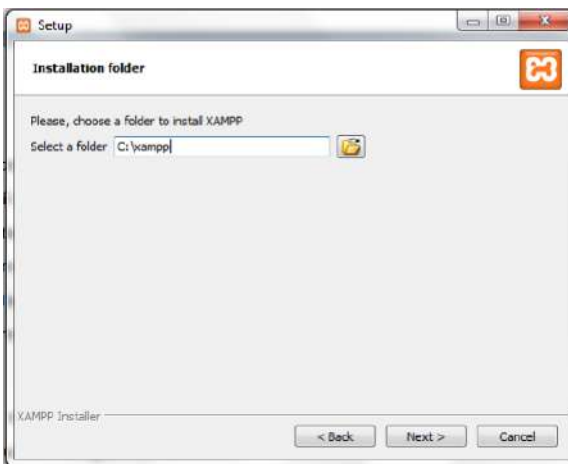
- Al pulsar sobre ejecutar, comienza la instalación, apareciendo:



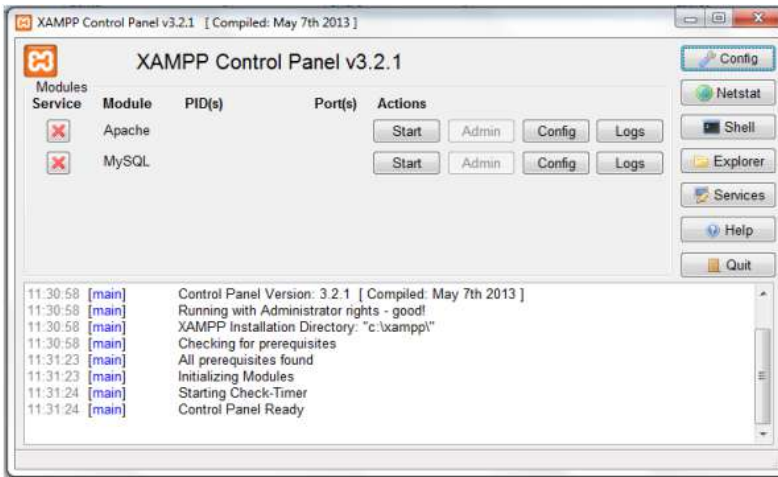
- Debemos pulsar sobre "Next" para continuar.



- debemos seleccionar la carpeta donde deseamos sea instalado XAMPP.



- Por último, iniciamos el xampp.



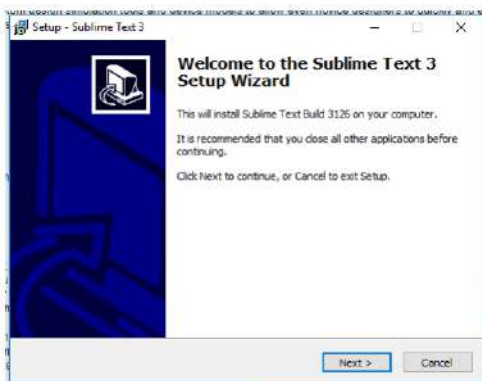
<https://biblus.us.es/bibing/proyectos/abreproy/12197/fichero/Manual+de+instalacion.pdf>

4.2 Instalación de sublime text

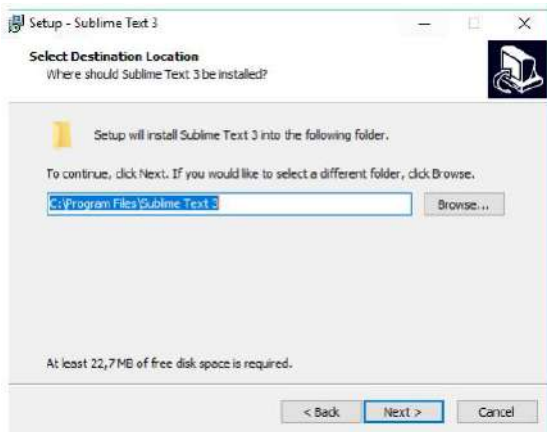
- Bajar ejecutable desde la página del centro de software.
- Click sobre el ejecutable.



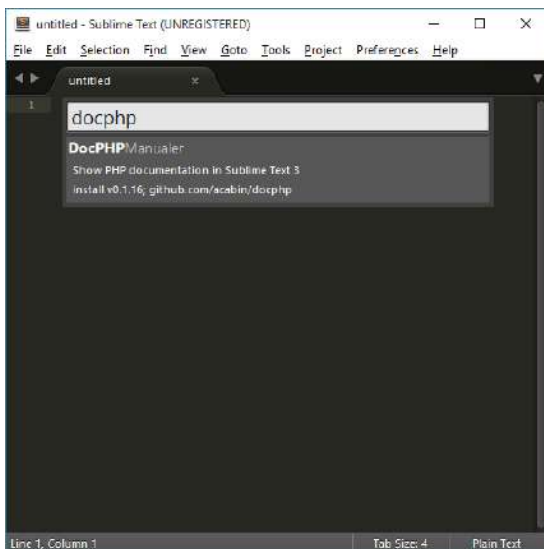
- Ejecutar y seguir el asistente de instalación hasta finalizar.



- Aceptar las condiciones y dejando por default o cambiándolo a preferencia.



- Y por último finalizar la instalación y proceder a usarlo.



<https://dc722jrlp2zu8.cloudfront.net/media/django-summernote/2016-02-15/dc5ac6c7-e2c7-4491-a110-8ad00a39c44a.png>

4.3 XML

El lenguaje de marcado extensible (XML) nos permite definir y almacenar datos de forma compatible. XML permite el intercambio de información entre los sistemas de computación, como sitios web, bases de datos entre otros.



https://rubiko.mx/wp-content/uploads/2021/09/archivo-xml_s.jpeg

4.4 ¿Qué es XML?

XML, son las siglas en inglés de Extensible Markup Language, es un metalenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium, cuya función es mostrar la información de manera estructurada, documentos, datos, configuración, facturas, entre otros.



<https://static.vecteezy.com/system/resources/previews/000/367/045/original/xml-vector-icon.jpg>

4.5 Principales Componentes.

Un archivo de lenguaje de marcado extensible (XML) es un documento basado en texto que se puede guardar con la extensión .xml. Puede

escribir XML de forma similar a otros archivos de texto. Para crear o editar un archivo XML, puede usar cualquiera de las siguientes opciones:

- Editores de texto como Notepad o Notepad++
- Editores XML en línea
- Navegadores web

Cualquier archivo XML incluye los siguientes componentes.

- Documento XML
- Declaración XML
- Elementos XML
- Atributos XML
- Contenido XML

<https://aws.amazon.com/es/what-is/xml/>

4.6 Tipos de documentos XML

- **Documentos bien formados**

Un documento XML debe estar bien formado, es decir debe cumplir las reglas de sintaxis de la recomendación XML. Para que un documento esté bien formado, al menos debe cumplir los siguientes puntos:

- El documento contiene únicamente caracteres Unicode válidos.
- Hay un elemento raíz que contiene al resto de elementos.
- Los nombres de los elementos y de sus atributos no contienen espacios.
- El primer carácter de un nombre de elemento o de atributo puede ser una letra, dos puntos (:) o subrayado (_).
- El resto de caracteres pueden ser también números, guiones (-) o puntos (.).

- **Documentos validos**

Un documento XML bien formado puede ser válido. Para ser válido, un documento XML debe:

- incluir una referencia a una gramática,
- incluir únicamente elementos y atributos definidos en la gramática,
- cumplir las reglas gramaticales definidas en la gramática.

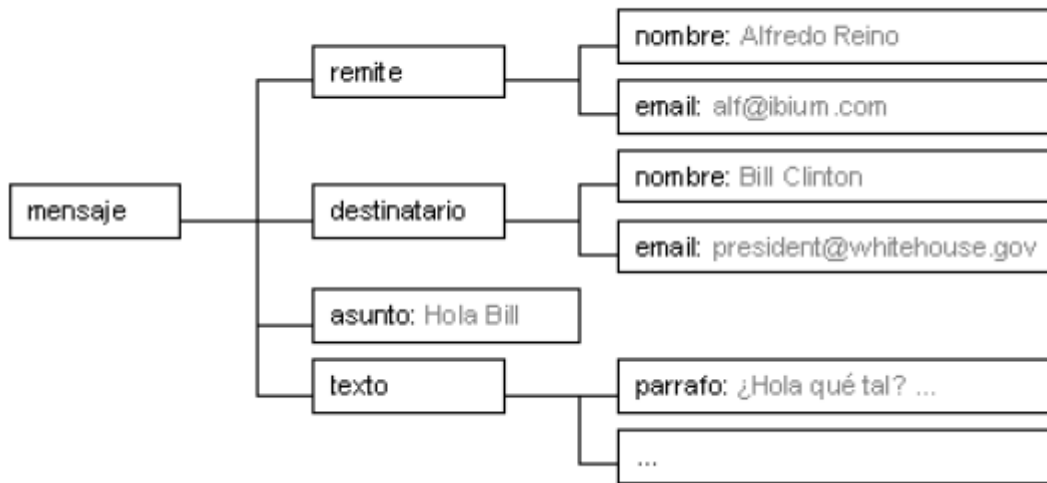
Existen varias formas de definir una gramática para documentos XML, las más empleadas son:

- DTD (Document Type Definition = Definición de Tipo de Documento). Es el modelo más antiguo, heredado del SGML.
- XML Schema. Es un modelo creado por el W3C como sucesor de las DTDs.
- Relax NG. Es un modelo creado por OASIS, más sencillo que XML Schema.

<https://www.mclibre.org/consultar/xml/lecciones/xml-conceptos-basicos.html>

4.7 Estructura de documentos XML

Aunque a primera vista, un documento XML puede parecer similar a HTML, hay una diferencia principal. Un documento XML contiene datos que se autodefinen, exclusivamente. Un documento HTML contiene datos mal definidos, mezclados con elementos de formato. En XML se separa el contenido de la presentación de forma total.



<https://desarrolloweb.com/articulos/intro-xml-y-estructura.html>

1.1 Framework web y scripting

Por: Jeremy Tobar

Un framework web es una estructura de software que proporciona herramientas y bibliotecas para facilitar el desarrollo de aplicaciones web. Estos frameworks establecen una estructura y un conjunto de convenciones que permiten a los desarrolladores crear aplicaciones web de manera más eficiente y organizada.

En cuanto al término "scripting", se refiere a la práctica de escribir scripts, que son programas de computadora pequeños y específicos que automatizan tareas o realizan operaciones simples. Los scripts suelen estar escritos en lenguajes de programación interpretados y se utilizan para realizar acciones repetitivas o procesar datos de manera eficiente. Algunos ejemplos comunes de lenguajes de scripting incluyen Python, JavaScript y Perl. Los frameworks web a menudo proporcionan soporte para la escritura y ejecución de scripts como parte de su funcionalidad.

1.1.1 Ejemplo

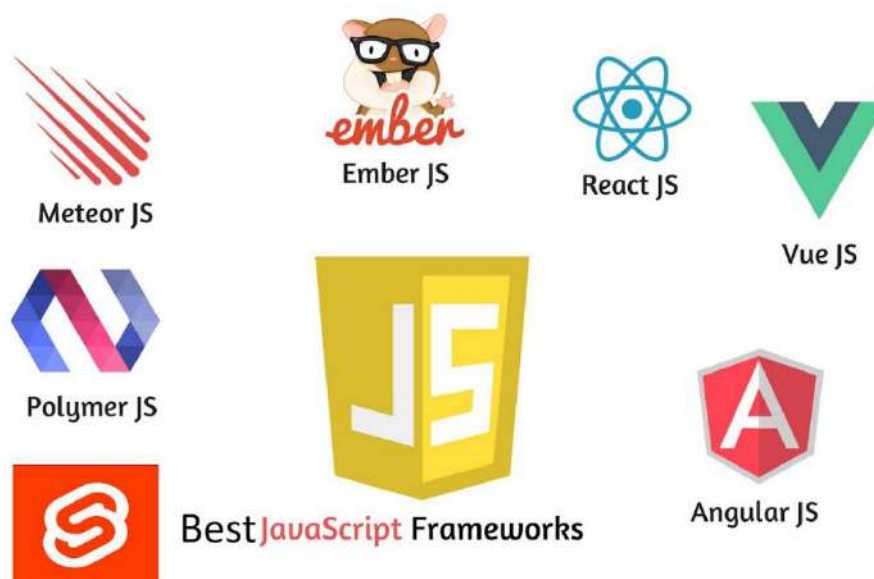
Django: Es un framework web de alto nivel escrito en Python. Proporciona una arquitectura MVC, ORM incorporado, enrutamiento URL, manejo de formularios y una gran cantidad de características para el desarrollo rápido y seguro de aplicaciones web.

Ruby on Rails: También conocido como Rails, es un framework web escrito en Ruby. Utiliza el patrón de diseño MVC y ofrece una estructura de convención sobre configuración. Rails es conocido por su enfoque en la simplicidad y la productividad del desarrollador.

Laravel: Es un framework web de código abierto basado en PHP. Laravel sigue el patrón MVC y proporciona una sintaxis elegante y expresiva para el desarrollo de aplicaciones web. Ofrece características como enrutamiento, migraciones de base de datos, autenticación, caché y mucho más.

Flask: Es un framework web minimalista escrito en Python. Está diseñado para ser simple y fácil de usar, pero lo suficientemente flexible como para admitir aplicaciones web de cualquier tamaño. Flask se enfoca en la extensibilidad y permite a los desarrolladores elegir las herramientas y bibliotecas que desean utilizar.

Express.js: Es un framework web rápido y minimalista para Node.js. Express.js proporciona una capa de abstracción sobre el servidor HTTP de Node.js, lo que facilita la creación de aplicaciones web y API. Es conocido por su enfoque sencillo y su amplia compatibilidad con middleware.



Estos son solo algunos ejemplos de frameworks web populares, pero hay muchos más disponibles en diferentes lenguajes de programación.

Fuente: https://www.entorno5.com/wp-content/uploads/2021/09/0_nlp1qOLSydDraQqT.png

1.1.2 Hitoria

La historia de los frameworks web se remonta a la necesidad de simplificar y agilizar el proceso de desarrollo web. A medida que las aplicaciones web se volvieron más complejas, se hizo evidente que había tareas comunes que se repetían en cada proyecto. Los frameworks web surgieron como una solución para modularizar y reutilizar esas tareas, proporcionando una base sólida y predefinida sobre la cual los desarrolladores pueden construir sus aplicaciones.

1.1 Framework web y scripting

Por: Diego Gabriel Vasquez Rosales

Es un framework diseñado para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web. Este tipo de frameworks intenta aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web. Es un término informal que se usa para designar un lenguaje de programación que se utiliza para manipular, personalizar y automatizar las instalaciones de un sistema existente.

1.1.1 Aplicaciones

El framework en la aplicación es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo de software. Utilizar un framework permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código.

1.1.2 Tecnologías

Es un conjunto de herramientas y librerías que se utilizan para desarrollar aplicaciones más fácilmente y de manera más eficiente.



Imagen: Ejemplo del tamaño solicitado (información)

Fuente:

https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.wearemarketing.com%2Fuploads%2Fmedia%2Fdefault%2F0001%2F21%2F1778829c66b7800a38a5c6ec92827c482fc056cb.jpeg&tbnid=LliFirLU-RMLM&vet=12ahUKEwj7y6-9sKr_AhWFIYQIHfhIAIMQMygGegUIARDAAQ..i&imgrefurl=https%3A%2F%2Fwww.wearemarketing.com%2Fes%2Fblog%2Fframeworks-en-el-desarrollo-web-las-mejores-practicas-para-tu-negocio-online.html&docid=N0QBfpZn09SLM&w=1600&h=900&q=framework%20tecnologias&client=opera-gx&ved=2ahUKEwj7y6-9sKr_AhWFIYQIHfhIAIMQMygGegUIARDAAQ

3. Laravel

Por: Eduardo Velásquez

Laravel es un popular framework de desarrollo de aplicaciones web basado en PHP. Fue creado por Taylor Otwell en 2011 y ha ganado una gran popularidad y reconocimiento en la comunidad de desarrollo debido a su elegante sintaxis, su enfoque en la legibilidad del código y su amplio conjunto de herramientas y características.

Laravel sigue el patrón de diseño MVC (Modelo-Vista-Controlador), lo que facilita la separación de la lógica empresarial y la presentación visual en una aplicación web.



Imagen: Logo de Laravel

Fuente: <https://blogs.masterhacks.net/wp-content/uploads/2023/02/laravel1.png-1.webp>

3.1 Entorno de trabajo

El entorno de trabajo de Laravel se compone de un servidor web, el uso de Composer para gestionar dependencias, el poderoso Artisan para la administración de comandos, el sistema de plantillas Blade, la interacción con bases de datos a través de Eloquent y la posibilidad de integrarse con herramientas de control de versiones. Estos componentes en conjunto proporcionan un entorno sólido y eficiente para el desarrollo de aplicaciones web con Laravel.

Laravel puede ejecutarse en varios servidores web populares como Apache, Nginx, o el servidor web integrado de PHP, lo que brinda flexibilidad al desarrollador para elegir el entorno que mejor se adapte a sus necesidades.

Laravel proporciona una potente interfaz de línea de comandos llamada Artisan. Con Artisan, se pueden ejecutar comandos para realizar tareas comunes de desarrollo, como generar código, ejecutar migraciones de base de datos, limpiar la caché, entre otros.

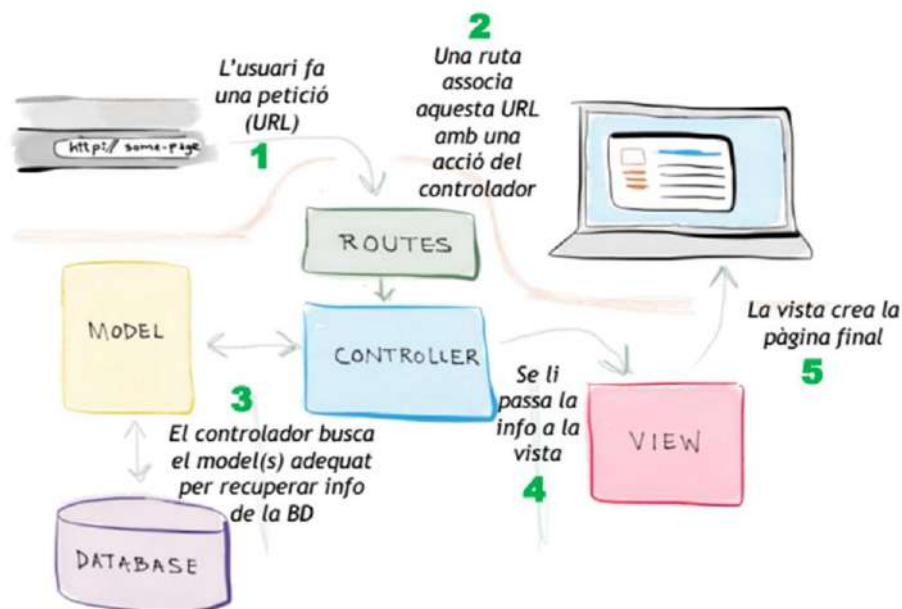


Imagen: Cómo usar Laravel

Fuente: [https://www.kiwop.com/wp-](https://www.kiwop.com/wp-content/uploads/2023/05/br86SEFocYWm8J3U8VTcUIJtroMFNzca7ct1fdlwmiov_aAI4bV7YhoMdJIA004czA2yEHPiPr1h-oblCc3MoDjCKD0A2pTWtuDZzMRz9I8NGLTo_3-EblcHdg74DXRQ9DfTaA3QJUzh8RuZDM1vRoMto04IZmAttf4FFDlqOirPVB1xPcjjl9Y-UdBDtw.png)

[content/uploads/2023/05/br86SEFocYWm8J3U8VTcUIJtroMFNzca7ct1fdlwmiov_aAI4bV7YhoMdJIA004czA2yEHPiPr1h-oblCc3MoDjCKD0A2pTWtuDZzMRz9I8NGLTo_3-EblcHdg74DXRQ9DfTaA3QJUzh8RuZDM1vRoMto04IZmAttf4FFDlqOirPVB1xPcjjl9Y-UdBDtw.png](https://www.kiwop.com/wp-content/uploads/2023/05/br86SEFocYWm8J3U8VTcUIJtroMFNzca7ct1fdlwmiov_aAI4bV7YhoMdJIA004czA2yEHPiPr1h-oblCc3MoDjCKD0A2pTWtuDZzMRz9I8NGLTo_3-EblcHdg74DXRQ9DfTaA3QJUzh8RuZDM1vRoMto04IZmAttf4FFDlqOirPVB1xPcjjl9Y-UdBDtw.png)

3.2 Comandos

- **Para crear un nuevo proyecto de Laravel:**

`composer create-project --prefer-dist laravel/laravel nombre-proyecto`

- **Ejecutar el servidor web local de Laravel:**

`php artisan serve`

- **Generar migración de base de datos:**

`php artisan make:migration nombre_migracion`

- **Ejecutar todas las migraciones de una pendiente:**

`php artisan migrate`

- **Crear un nuevo controlador:**

`php artisan make:controller NombreControlador`

- **Crear un nuevo modelo:**

`php artisan make:model NombreModelo`

- **Generar una nueva solicitud:**

`php artisan make:request NombreSolicitud`

- **Generar una nueva factoría:**

`php artisan make:factory NombreFactoria --model=NombreModelo`

- **Generar una nueva semilla:**

`php artisan make:seeder NombreSemilla`

- **Ejecutar todas las semillas:**

`php artisan db:seed`

3.3 Node JS

Node.js es un entorno de ejecución de código JavaScript en el servidor que proporciona un enfoque eficiente y escalable para el desarrollo de aplicaciones del lado del servidor. Con su modelo de eventos, amplio ecosistema de módulos y rendimiento rápido, Node.js se ha convertido en una opción popular para el desarrollo web y otras aplicaciones de red.



Imagen: Logo de node.js

Fuente: https://upload.wikimedia.org/wikipedia/commons/thumb/d/d9/Node.js_logo.svg/1200px-Node.js_logo.svg.png

3.4 Composer

Composer es una herramienta esencial para el desarrollo de aplicaciones PHP, ya que facilita la gestión de dependencias y asegura que las bibliotecas necesarias estén disponibles y actualizadas en tu proyecto. Al simplificar la gestión de dependencias, Composer contribuye a la eficiencia y la reutilización de código en el ecosistema de desarrollo de PHP.

3.4.1 Características

- Gestión de dependencias: Composer resuelve automáticamente las dependencias de tu proyecto, descargando e instalando las versiones correctas de las bibliotecas

requeridas. También maneja las dependencias transitivas, asegurando que todas las bibliotecas requeridas estén disponibles.

- Repositorios de paquetes: Composer utiliza repositorios de paquetes para buscar y descargar las dependencias. El repositorio principal es Packagist, donde se alojan miles de paquetes de PHP disponibles para su uso.
- Autocarga de clases: Composer genera automáticamente un archivo de autocarga de clases (autoload.php), que se encarga de cargar automáticamente las clases de las dependencias en tu proyecto, eliminando la necesidad de incluir manualmente los archivos de las bibliotecas.
- Scripts personalizados: Composer permite la ejecución de scripts personalizados en diferentes momentos del ciclo de vida del proyecto, como antes o después de la instalación o actualización de dependencias. Esto es útil para realizar tareas adicionales, como la generación de archivos de configuración o la ejecución de pruebas automatizadas.



Imagen: Logo de Composer

Fuente: <https://upload.wikimedia.org/wikipedia/commons/2/26/Logo-composer-transparent.png>

1.1 Laravel

Por: Ricardo Mauricio Ventura Martínez

1.1.1 Nueva página

Primero se crea un archivo llamado inicio.blade.php dentro de resources/views. La extensión blade.php indica que se trata de una plantilla de blade que te permite realizar multitud de tareas dentro de tu vista de una manera muy sencilla

Segundo crea tu ruta en el archivo routes/web.php con el siguiente código: `Route::get('/', 'HomeController@inicio');`. Esto le dice a Laravel que dentro de la clase HomeController ejecute el método inicio.

Paso 3 Introduce en tu controlador la acción que quieres que se realice, creando la función inicio y diciéndole que cargue la vista inicio, y con eso ya tendrías tu vista en laravel y cargaría la vista. El código sería el siguiente:

```
1  <?php
2  namespace App\Http\Controllers;
3
4  use Illuminate\Http\Request;
5
6
7  class HomeController extends Controller
8  {
9      public function inicio(){
10         return view('inicio');
11     }
12
13
14 }
```

Fuente: [Crear una página en Laravel - Miguel Angel López WEB \(mialtoweb.es\)](https://mialtoweb.es/crear-una-pagina-en-laravel/)

1.1.2 Controladores

Los controladores en Laravel son una parte importante de las aplicaciones y forman parte del patrón MVC (Modelo-Vista-Controlador). Su función es definir el código a ejecutar como comportamiento frente a una acción solicitada dentro de la aplicación.

El controlador sabe qué métodos del modelo debe invocar, ya sea para actualizar cierta información o para obtener ciertos datos, así como las vistas que deben presentar la información como respuesta al usuario después de la realización de las acciones necesarias.

Los controladores están localizados en la carpeta `app/Http/Controllers` y puedes organizarlos en subcarpetas si lo deseas. Como otras clases de Laravel, están dentro del sistema de autocarga de clases, por lo que estarán disponibles siempre que los necesites en la aplicación

1.1.3 Listas

En Laravel, las colecciones son una forma de trabajar con arrays de datos. La clase `Illuminate\Support\Collection` proporciona una envoltura fluida y conveniente para trabajar con ellos.

Las colecciones son inmutables, lo que significa que cada método de colección devuelve una instancia completamente nueva de la colección.

A continuación, vamos a generar un listado de productos utilizando una ruta y una vista. Primero declaremos la ruta en `routes/web.php`:

```

1. <?php
2.
3. Route::get('/productos', function () {
4.     $products = \App\Product::all();
5.
6.     return view('products', ['products' => $products]);
7. });

```

Fuente: <https://styde.net/creacion-de-un-listado-de-productos-con-categoria-con-laravel-y-eloquent/>

Luego creamos una plantilla de Blade en `resources/views/products.blade.php` con el siguiente código:

```

1. <!doctype html>
2. <html lang="es">
3. <head>
4.     <meta charset="UTF-8">
5. </head>
6. <body>
7.     <ul>
8.         @foreach($products as $product)
9.             <li>
10.                <a href="{{ url('productos/'.$product->slug) }}">
11.                    <strong>{{ $product->title }}</strong>
12.                </a> -
13.                <a href="{{ url('productos/categoria/'.$product->category->slug) }}">
14.                    {{ $product->category->title }}
15.                </a>
16.            </li>
17.        @endforeach
18.    </ul>
19. </body>
20. </html>

```

Fuente: <https://styde.net/creacion-de-un-listado-de-productos-con-categoria-con-laravel-y-eloquent/>

1.1.4 Enlaces de páginas

El helper url genera una URL completa para una ruta dada, mientras que el método route genera una URL para una ruta con nombre.

```
Route::get('/post/{id}', function ($id) {  
    //  
})->name('post.show');
```

Puedes generar un enlace a esta ruta utilizando el método route de la siguiente manera:

```
$url = route('post.show', ['id' => 1]);
```

Esto generará una URL como <http://your-app.com/post/1>.

También puedes generar enlaces utilizando el helper url de la siguiente manera:

```
$url = url('/post/1');
```

Esto también generará una URL como <http://your-app.com/post/1>.

3. Laravel

Por: Ricardo Yoc

3.7 Listas Dinámicas

Se refieren a la funcionalidad de cargar y actualizar selectores (elementos de selección) de forma asincrónica, sin necesidad de recargar toda la página. Esto permite que los usuarios interactúen con un selector y obtengan opciones adicionales basadas en su selección, sin interrupciones en la experiencia de usuario.

En una lista dinámica típica, al seleccionar un valor en un selector principal, se envía una solicitud AJAX al servidor para obtener los datos relacionados con la opción seleccionada. Luego, esos datos se utilizan para actualizar dinámicamente un selector secundario o mostrar los resultados relacionados en la página web.

Laravel, junto con JavaScript y la funcionalidad de carga asincrónica de datos, proporciona una forma eficiente de implementar listas dinámicas y mantener una interacción fluida con los usuarios..

Registro de Horas. Semana 24-Julio-07 || 30-Julio-07

PROYECTOS	TAREAS	SUBTAREAS	LUN	MAR	MIE	JUE	VIE	SAB	DOM		
--- Proyectos ---	--- Tareas ---	--- Subtareas ---	-	-	-	-	-	-	-	✖	✓
--- Proyectos ---	--- Tareas ---	--- Subtareas ---	-	-	-	-	-	-	-	✖	✓
--- Proyectos ---	--- Tareas ---	--- Subtareas ---	-	-	-	-	-	-	-	✖	✓

Nuevo Objetivo

- 1
- 2
- 3
- 4
- 5

Imagen: Listas Dinámicas Laravel
Fuente: <https://i.stack.imgur.com/eG8yg.jpg>

3.8 Diseño

El diseño en Laravel se refiere a la estructura y organización de los archivos y componentes de tu aplicación Laravel, así como a la forma en que se implementa la lógica de presentación y la interfaz de usuario. Aquí hay algunos aspectos clave del diseño en Laravel:

- **Patrón de diseño MVC:** Laravel sigue el patrón de diseño Modelo-Vista-Controlador (MVC) para separar las preocupaciones y organizar el código de manera clara y mantenible.
- **Vistas y plantillas:** Laravel utiliza el motor de plantillas Blade para generar las vistas HTML. Blade proporciona una sintaxis sencilla y poderosa para mezclar código PHP con HTML, lo que permite una gestión eficiente de las vistas y una separación clara entre la lógica y la presentación.
- **Rutas y controladores:** Laravel utiliza el enrutamiento para asignar las URL a las acciones del controlador correspondiente.

3.9 Funciones

Las funciones se refieren a los bloques de código que realizan tareas específicas y pueden ser invocadas en diferentes partes de una aplicación. Las funciones en Laravel se utilizan para encapsular la lógica de negocio, realizar operaciones de base de datos, procesar solicitudes HTTP, manipular datos y mucho más.

- **Funciones de ruta:** Se definen en el archivo routes/web.php y se utilizan para definir las rutas de la aplicación.
- **Funciones de controlador:** Los controladores se utilizan para agrupar la lógica relacionada con una determinada entidad o funcionalidad de la aplicación.
- **Funciones de modelo:** Los modelos se utilizan para interactuar con la base de datos y representar los datos en forma de objetos.
- **Funciones de la clase "DB" (fachada de base de datos):** Estas funciones permiten realizar consultas y manipulaciones de datos sin la necesidad de escribir consultas SQL directamente.
- **Funciones de sesión:** Estas funciones permiten almacenar y recuperar datos en la sesión del usuario, lo que es útil para mantener el estado de la aplicación o almacenar información temporal.

```
public function getHome()
{
    $tipoUser = Auth::user()->tipoUser;
    $userID = Auth::user()->user_id;

    $eventos = obtenerEvento($tipoUser, $userID);

    $data = ['eventos'=>$eventos];
    return view('dashboard.eventos.home', $data);
}

public function obtenerEvento($tipoUser, $userID)
{
    ...
}
```

Imagen: Uso de Funciones

Fuente: <https://i.stack.imgur.com/007JL.png>

3.10 Alertas

Las alertas son mensajes que se utilizan para mostrar información importante, notificaciones, mensajes de éxito o mensajes de error a los usuarios de una aplicación web. Estas alertas se utilizan para proporcionar retroalimentación visual sobre el resultado de una acción o para comunicar información relevante.

Laravel proporciona una funcionalidad incorporada para trabajar con alertas a través de la sesión flash. La sesión flash es una característica que permite almacenar datos en la sesión por una única solicitud y, luego, recuperar esos datos en la siguiente solicitud.

Para mostrar alertas en Laravel, puedes utilizar la clase `Session` o la fachada `Session` para almacenar los mensajes en la sesión flash. A continuación, puedes recuperar esos mensajes en la siguiente solicitud y mostrarlos en tu interfaz de usuario.

Las alertas en Laravel son una forma efectiva de proporcionar retroalimentación visual y comunicar información importante a los usuarios de tu aplicación web.



Imagen: Alertas con Laravel

Fuente: <https://styde.net/notificaciones-y-alertas-con-el-componente-stydehtml/>

TEMARIO QUINTO COMPUTACIÓN

Autoridades

Instituto Emiliani Somascos

Comunidad Somasca
Obras Somasca en Guatemala

Lic. Raúl Hernández Chacón
Director Técnico-Administrativo Instituto Emiliani Somascos

Lic. Henry Caal
Sub-director Instituto Emiliani Somascos

Lic. Juan Carlos Morales
Coordinador Académico

Prof. David Subuyuj
Coordinador Técnico

Armando Garcia
Coordinación de Pastoral

Bachillerato Industrial y Perito con especialidad en Computación

Silvio Ernesto Castro
Asesor de Práctica Supervisada

Christian Morales
Asesor de Práctica Supervisada

Promoción 2023

6to Computación

Nombres

Sección "A"

- Edwin Eduardo Alvarez Arriaga
- Diego Angel de Jesús Alvizures Orantes
- David Nehemias Bautista Patzán
- Brayan Ismael Chóm Pacajoj
- Cristian Alexander Coromac López
- Willian Estuardo Cuevas Rivas
- Nohelia Guadalupe Culajay Oxcal
- Jenifer Celeste Cuyan Gomez
- Angel Daniel De León Alvarado
- Kevin Alexander De Leon Andrade
- Marta Alicia del Carmen Diaz Cifuentes
- Oscar Alexander Echeverría López
- Aroldo David Escobar López
- Thania Paola García Cos
- David Daniel Gerónimo Vaquiax
- Patrick Leonardo Gómez Paz
- Joshua Alessandro González Gonzáles
- Brandon David Grijalva Marroquín
- Braulio José Alexander Guarcax Lima
- Eddy Junior Guerrero Gomez
- Ricardo Alberto Hernández Bautista
- José Santiago Herrera Furlán
- Cristian Antonio Im Monterroso
- José Carlos Jordán Cardona
- Angel Enrique Juarez Asencio
- Gustavo Rafael Juárez Mendoza
- María Emilia Larios Contreras
- Glendy Sucely Lopez Cruz
- Didier Joshua Lopez Juarez
- Byron Aaron Mansilla Fuentes
- German Daniel Marquez Estrada
- Elder César Oswaldo Martínez Rustrian

Sección "B"

- Adolfo Isaác Morales Delgado
- Erick Estuardo Morán Pech
- María José Noj Sor
- José Antonio Osoy Crespo
- Diego Manuel Fernando Pacheco Racanac
- Pavel Alexander Palma Carrillo
- Erwin Gabriel Patzán Roca
- Wilmer Estuardo Patzán Vail
- Leonel De Jesús Erasmo Pérez Lúcas
- Krystel Adriany Popa Patzán
- Rodrigo Fernando Ramírez Guzmán
- José Gabriel Ramos Pérez
- Fernando José Raxón Pirir
- Hilda Aileen Raymundo Diaz
- Gabriel Eduardo Rivas Morales
- Ariel Ernesto Rodríguez Morales
- Iker Rodolfo Rojas Reyes
- David José Gabriel Romero Alvarado
- Ronaldo Hezler Adriel Ruano Valle
- Gabriela Alexandra Santos Carrera
- Dilan Elizandro Sarceño Jacinto
- Dulce María Shuyá Baten
- Maryori Omeydi Sipac Sesam
- Edgar Laureano Soc Rac
- Juan Alejandro Tiney García
- Jeremy Jhosef Tobar Lopez
- Jonathán Javier Tubac Arreaza
- Diego Gabriel Vásquez Rosales
- Eduardo Steve Velásquez Gómez
- Ricardo Mauricio Ventura Martínez
- Axel Daniel Yoc Garcia
- Mario Ricardo Yoc Hernández

Centenario Somasco en América

Cien años de la comunidad somasca en América, aquel lejano 5 de octubre de 1921 en el cual tuvimos el arribo en el puerto de la libertad, en el vecino país del Salvador, el que significa el primer apostolado somasco. Posteriormente en 1927 comenzaron las obras en Guatemala. Pensar que tenemos ya en este año 2023 más de 100 años de nosotros también ser partícipes de una obra que sigue hasta el día de hoy creciendo y expandiéndose, un mensaje de perseverancia que nos muestra que no todo llega rápidamente, pero que en algún momento todo da sus frutos. Solamente Dios sabe qué es lo que hace y cómo lo hace, cuál es el rumbo de las cosas, de su corazón nacen las cosas que son buenas, la ayuda, la sabiduría, la inteligencia, solo vienen de él, sus planes para todos en los que en él confían, serán siempre de bendición. Él es el que da las herramientas para que podamos salir adelante, siempre interfiere en todo lo que nos pasa en el día a día, en cada hora, en cada minuto, cada instante de nuestras vidas está planeado desde mucho antes de que nosotros conociéramos el primer instante de vida.

Eso significa la comunidad, una comunidad que significó una luz, una muestra de que el amor divino, que la mano de Dios aún está con nosotros y que seguirá estando, de la mano de todos aquellos colaboradores y de todos aquellos corazones que están conectados con la espiritualidad, que son verdaderos devotos a la obra verdaderamente cristiana, que mucha falta hace en estos tiempos tan difíciles.

Una comunidad que ha hecho muchas obras de bien para la sociedad, fundando instituciones educativas, seminarios, con el fin siempre de acercar más a las personas a Jesús, que es la verdadera visión y vocación que siempre se ha querido plasmar y transmitir hacia cada persona que está involucrada en esta obra masiva de devoción, caridad y trabajo. Lo que representó San Jerónimo a partir de su conversión, y que demostró toda su vida a partir de aquella experiencia propia que tuvo con la Virgen María. Su virtud es llevar el mensaje de salvación a partir de la ayuda hacia aquellos que más lo necesitan.

La pandemia que vivimos en estos últimos dos años, sin duda significó muchas cosas perdidas, familias que perdieron trabajo, que están pasando angustias económicas, incluso familias en dónde muchos miembros ya no están, suena como un mensaje pesimista, pero es la verdadera realidad en la cual nos encontramos viviendo, pero cuando hacemos algo, por ayudar a alguien más estamos contribuyendo al propósito de nuestras vidas.

Cuando empezamos a ver más allá de lo que nuestra perspectiva común ve, cuando dejamos de centrarnos en lo que pasamos, o a aún con problemas, luchas externas e internas, podemos ver aquellas realidades en las que nosotros, podemos intervenir, y quizá no cambiemos la realidad de la persona, hay situaciones realmente difíciles, pero, sí podemos dar una palabra incluso que venga del corazón, y dar ánimo nuevamente a una persona.

Porque la obra de caridad es algo más allá que ser llamado propiamente un santo, porque la caridad está en cada uno de aquellos corazones dispuestos a sacrificar algo, por servir, por ayudar a los demás. Porque la caridad requiere de intencionalidad también, el deseo y el sentido del deber, de dar sin esperar recibir nada a cambio, realmente eso es lo que completa una verdadera conversión, porque si recibimos la salvación, y la

misericordia de Cristo, cómo no tener misericordia de aquellos que alguna vez fuimos nosotros.

El Centenario Somasco en América, significa más que solo 100 años, significan años de trabajo, de sacrificio, de cambios, de una transformación que incluso todos buscamos a diario. Quizá lo más significativo más allá de toda la historia que engloba estos cien años de la Comunidad Somasca en América, tenemos un proyecto de vida que busca reunir a todas aquellas personas marginadas, necesitadas, dirigirse a los niños, como el mismo Jesús lo haría en su tiempo. Resulta más que grata su presencia en América y propiamente en Guatemala, y deseamos que, así como tenemos un centenario se sigan cumpliendo más y más años de nuevos retos, pero sobre todo de practicar el amor al prójimo, el mensaje de salvación por medio de las obras.



Introducción

6to Computación

A lo largo de este trabajo como alumnos del 6to grado en bachiller industrial y perito con especialidad en computación, daremos a conocer cada uno de los temas vistos e impartidos por nuestros docentes en el ciclo escolar 2021 que corresponde al grado de Quinto grado de nuestra especialidad ya antes mencionada.

Explicaremos y daremos a conocer en cada tema que en este documento se presenta, cada uno de los aspectos, características, aplicaciones y utilidades de dichos temas, tratando de dejar claro cada uno de ellos.

Abarcando de una manera clara y directa todo lo visto en ese ciclo escolar, principalmente orientado a la programación en C++, programación orientada a objetos en C#, conexiones a bases de datos, Visual Basic, entre muchos otros temas que serán de apoyo y guía para retroalimentar y fomentar en los estudiantes el correcto desarrollo de aplicaciones y proyectos que necesiten de estos tipos de software, lenguajes y conocimientos.

Justificación

6to Computación

El motivo por el cual todos los estudiantes realizamos la recopilación de toda la información que se presenta en este documento, es principalmente para que cada uno de nosotros tenga acceso a un medio de estudio y retroalimentación de todos los temas comprendidos, no solamente este año, sino también para un ámbito laboral y profesional que como futuros graduandos no estamos lejos a alcanzar, Cada uno podrá estudiar y repasar para mejorar y poner en práctica cada una de nuestras habilidades adquiridas a lo largo de todo este ciclo escolar. Ya que será información que no solo nos servirá este año, sino que, si nos esmeramos y ponemos empeño en cada una de las prácticas y competencias en cada tema, vamos a adquirir los conocimientos necesarios que nos servirán en un grado de educación superior universitaria y a lo largo de toda nuestra vida laboral.

Objetivos

6to Computación

En el siguiente documento tenemos una recopilación de todos los temas vistos en el primer año de nuestra carrera, todos estos temas son tanto prácticos como técnicos, este documento nos apoya a la retroalimentación sobre todo lo que hemos aprendido en el transcurso de los años.

El principal objetivo de este documento es poder estudiar todos los temas que contiene e introducirnos nuevamente a los temas, y así no dejarlos a un lado ni olvidarlos, ya que tanto estos como los que vemos este último año de carrera son de suma importancia para nuestra realización como profesionales.

Nuestro temario está dividido en clases y capítulos, cada clase tiene cuatro capítulos representando los bimestres transcurridos. En cada uno de los capítulos se pueden identificar los temas principales que vimos todo ese año. Leer el temario es una buena práctica para repasar los conocimientos adquiridos con anterioridad.

Los objetivos específicos de nuestro temario son:

- Retroalimentarnos respecto a los temas pasados.
- Explicar de forma fácil de retener, clara y concisa cada tema tocado en este documento.
- Brindar ejercicios prácticos que ayudan a la retención de información.

El temario tiene como fin el estudio del mismo para los graduandos que cursen el presente año 2023.

Temas

Quinto grado

Tecnología Vocacional II

Capítulo I

1. VISUAL STUDIO .NET POO. Y REPORTERIA DE DATOS .NET.
 - 1.1. Formularios de Windows.
 - 1.2. Controles Windows.
 - 1.3. Diferencias entre Acceso y SQL.
 - 1.4. Conexión SQL Server 2014.
 - 1.5. CRUD.
 - 1.6. Codificación Avanzada.
 - 1.7. Anti errores y Try Catch.
 - 1.8. Enlace a SQL Server.
 - 1.9. Diseño de Reporte.
 - 1.10. Campos calculados.
 - 1.11. Instalación de ReportView.
 - 1.12. Instalación de CrystalReport.
 - 1.13. ReportView.
 - 1.14. Relacionar y Diseñar.
 - 1.15. Impresión de reports.
 - 1.16. CrystalReport.
 - 1.17. Relacionar y Diseñar.
 - 1.18. Diferencias entre ReportView y CrystalReport.
 - 1.19. Impresión de reportes.

Capítulo II

2. Java Consola.
 - 2.1. Entorno de trabajo ide's.
 - 2.2. Introducción a Java.
 - 2.3. ¿Qué es Java?
 - 2.4. Instalar Java IDE's.
 - 2.5. Variables de entorno.
 - 2.6. Estructura general de un p.
 - 2.7. Salida de datos en consola.
 - 2.8. Compilador primario de Java.
 - 2.9. Variables e Identificadores.
 - 2.10. Tipos de datos.
 - 2.11. Constantes.
 - 2.12. Operadores lógicos, aritméticos.
 - 2.13. Aritméticos.
 - 2.14. Lógicos.
 - 2.15. Relacionales.
 - 2.16. Expresiones.
 - 2.17. Reglas de prioridad.
 - 2.18. Entrada de datos en consola.
 - 2.19. Clases buffered.

- 2.20. Clase scanner.
- 2.21. Sentencias condiciones.
 - 2.21.1. Simple y Doble.
 - 2.21.2. Múltiples o anidadas.
 - 2.21.3. Swich Case.
- 2.22. Estructuras de repetición.
 - 2.22.1. While.
 - 2.22.2. Do While y For repetitivas.
 - 2.22.3. Anidadas.
- 2.23. Vectores y matrices.
 - 2.23.1. Arreglos.
 - 2.23.2. Unidimensionales.
 - 2.23.3. Bidimensionales.
- 2.24. Métodos.
- 2.25. Procedimientos.
- 2.26. Funciones.

Capítulo III

- 3. JAVA POO SWING.
 - 3.1. Programación Orientada a Objetos.
 - 3.2. Estructura de una clase.
 - 3.3. Atributos y Ámbitos de variables.
 - 3.4. Métodos accesores (get y set).
 - 3.5. Colaboración de clases.
 - 3.6. Herencia.
 - 3.7. Polimorfismo.
 - 3.8. Sobre carga de métodos.
 - 3.9. Arreglos de objetos.
 - 3.10. Swing Introduction.
 - 3.11. Java JButton.
 - 3.12. Java JLabel.
 - 3.13. Java JTextField.
 - 3.14. Java JTextArea.
 - 3.15. Java JPasswordField.
 - 3.16. Java JCheckBox.
 - 3.17. Java JRadioButton.
 - 3.18. Java JComboBox.
 - 3.19. Java JTable Java JList.
 - 3.20. Java JOptionPane.
 - 3.21. Java JScrollBar.
 - 3.22. Java JMenuItem & JMenu.
 - 3.23. Java JPopupMenu.
 - 3.24. Java JCheckBoxMenuItem.
 - 3.25. Java JProgressBar.
 - 3.26. Java JColorChooser.
 - 3.27. Java JTabbedPane.
 - 3.28. Java JSlider open link.
 - 3.29. Java JSpinner, Java JDialog, JPanel.
 - 3.30. Java JFileChooser.
 - 3.31. Java JToggleButton.
 - 3.32. Java JToolBar, JViewport.
 - 3.33. Java JFrame, JComponent.

- 3.34. Java JDesktop, JEditorPane, JScrollPane, JSplitPane y JTextPane.

Capítulo IV

- 4. Sistema Android.
 - 4.1. Historia.
 - 4.2. Adquisición por parte de Google.
 - 4.3. Historial de actualizaciones.
 - 4.4. Versiones.
 - 4.5. Arquitectura.
 - 4.6. Seguridad, privacidad y vigilancia.
 - 4.7. Usos y dispositivos.
 - 4.8. Diseño y desarrollo.
 - 4.9. Mercadotecnia.
 - 4.10. Entorno Android Studio.
 - 4.10.1. Instalación de para Android Studio.
 - 4.10.2. Pasos para crear el primer proyecto.
 - 4.10.3. Capturar el clic de un botón
 - 4.10.4. Controles RadioGroup y RadioButton.
 - 4.10.5. Control CheckBox.
 - 4.10.6. Control Spinner.
 - 4.10.7. Control ListView (lista de String).
 - 4.10.8. Control ImageButton.
 - 4.10.9. Notificaciones sencillas en clase Toast.
 - 4.10.10. Control EditText.
 - 4.10.11. Lanzar un segundo "Activity".
 - 4.10.12. Lanzar un segundo "Activity" y pasar.
 - 4.10.13. Instalación Android en un dispositivo.
 - 4.10.14. Layout (LinearLayout).
 - 4.10.15. Layout (TableLayout).
 - 4.10.16. Layout (FrameLayout).
 - 4.10.17. Layout (ScrollView y LinearLayout).
 - 4.10.18. Icono de la aplicación.
 - 4.10.19. Reproducción de audio.
 - 4.10.20. Reproducción, pausa, continuación y detención de un archivo de audio.
 - 4.10.21. Reproducción de audio.
 - 4.10.22. Reproducción de audio utilizando el reproductor propio de Android (v. Intent).
 - 4.10.23. Grabación de audio mediante el grabador provisto por Android (via Intent).
 - 4.10.24. Captura de audio mediante la clase MediaRecorder.

Análisis de Sistemas II

Capítulo I

1. SQL (Structured Query Language).

- 1.1. Instalación SQL server 2014.
- 1.2. Configuración SQL server.
- 1.3. Sintaxis SQL.
- 1.4. Comentar código.
- 1.5. SQL SELECT.
- 1.6. SQL SELECT DISTINCT.
- 1.7. SQL Y, O, No.
- 1.8. SQL Ordenar Por.
- 1.9. SQL Insertar y actualización.
- 1.10. Select de SQL Top.
- 1.11. SQL mínimo y máximo.
- 1.12. Conde SQL, Media, Suma.
- 1.13. SQL como Crear SQL DB.
- 1.14. SQL comodines y Check.
- 1.15. alias de SQL Inner Join.
- 1.16. SQL Left Join y Right Join.
- 1.17. SQL Full Join.
- 1.18. SQL Self Join, Union..
- 1.19. SQL Group By y Any, All
- 1.20. SQL Procedimientos.
- 1.21. SQL Comentarios de db.
- 1.22. SQL de copia de seguridad DB.
- 1.23. SQL Crear la tabla y DROP.
- 1.24. Modificar tabla.
y restricciones.
- 1.25. SQL no nulos y único.
- 1.26. SQL de clave principal.
- 1.27. SQL clave externa
- 1.28. Inyecciones SQL.
- 1.29. SQL predeterminado.
- 1.30. SQL incremento automático.
- 1.31. alojamiento SQL.
- 1.32. Modificar tabla y restricciones.
- 1.33. SQL no nulos y único.
- 1.34. SQL de clave principal.
- 1.35. SQL clave externa.
- 1.36. Inyecciones SQL.
- 1.37. SQL predeterminado.
- 1.38. SQL incremento automático.
- 1.39. Alojamiento SQL.

Capítulo II

2. GNU/Linux en un entorno de trabajo.

- 2.1. Introducción.
- 2.2. Objetivo.
- 2.3. Créditos.

- 2.4. Licencia.
- 2.5. Notación y convenciones.
- 2.6. ¿Qué son Unix y.
- 2.7. Unix, Linux y GNU.
- 2.8. Directorios y comandos.
 - 2.8.1. Archivos y directorios.
 - 2.8.2. El árbol de directorios.
 - 2.8.3. Directorio de trabajo.
 - 2.8.4. Refiriéndose al directorio.
 - 2.8.5. Primeros pasos en Linux.
 - 2.8.6. Moviéndonos por el entorno.
 - 2.8.7. Viendo el contenido de los directorios.
 - 2.8.8. Creando directorios.
 - 2.8.9. Copiando archivos.
 - 2.8.10. Moviendo archivos.
 - 2.8.11. Borrando archivos y directorios.
 - 2.8.12. Viendo el contenido de archivos.
 - 2.8.13. Obteniendo ayuda en línea.
 - 2.8.14. Sumario de comandos básicos.
 - 2.8.15. Operaciones sobre directorios.
 - 2.8.16. Operaciones sobre arc.
 - 2.8.16.1. Otras utilidades.
 - 2.8.16.2. Caracteres "comodín".
- 2.9. SHELL.
 - 2.9.1. Comunicación entre p.
 - 2.9.2. Entrada y salida estándar.
 - 2.9.3. Redirigiendo la E y S.
 - 2.9.4. Redirección no Destructiv.
 - 2.9.5. Permisos de archivos.
 - 2.9.6. Tipos de permisos.
 - 2.9.7. Interpretando los permisos de archi.
 - 2.9.8. Programación del Shell.
 - 2.9.9. Administración del S.
 - 2.9.10. Distribuciones de Linux.

Capítulo III

- 3. Conectar Java con MySQL en Apache NetBeans.
 - 3.1. Instalación de Mysql Workbench 8.
 - 3.2. Instalación de Wamp o Xamp..
 - 3.3. JDBC con MySQL
 - 3.4. Formularios de registros.
 - 3.5. Consultas.
 - 3.6. Formularios con consultas.
 - 3.7. Consulta de tabla de MySQL.
 - 3.8. Leer resultados..
 - 3.9. Baja y modificación de datos de una tabla de MySQL
 - 3.10. Generar comentarios en Código.

- 3.11. Estructura de datos.
 - 3.11.1 Listas, Colas y Pilas.
- 3.12. Estructuras dinámicas.
 - 3.12.2 Litas tipo Pila.
 - 3.12.3 Litas tipo Cola.
 - 3.12.4 Problemas de aplicación.
 - 3.12.5 Listas genéricas.
 - 3.12.6 Listas genéricas ordenadas.
 - 3.12.7 Listas genéricas doblemente encadenadas.
 - 3.12.8. Listas genéricas circulares.
- 3.13. Árboles binarios.
- 3.14. Teoría de grafos.
- 3.15. Tipos de árboles binarios
- 3.16. Recorridos en profu.
- 3.17. Recorridos en pre-orden.
- 3.18. Recorridos en post orden.
- 3.19. Recorridos en in orden.
- 3.20. Recursividad: Conceptos.
- 3.21. Problemas donde conviene aplicar la recursividad.
- 3.22. Implementación Java.
- 3.23. Ejemplos de la vida cotidiana.

Capítulo IV

4. Creando entornos y diseños web con XML.

- 4.1. Instalación de XAMPP.
- 4.2. Instalación de Sublime Text.
- 4.3. XML.
- 4.4. ¿Qué es XML?
- 4.5. Principales componentes.
- 4.6. Tipos de documentos XML.
- 4.7. Estructura de documentos XML.
- 4.8. Elementos de XML.
- 4.9. Etiquetas de XML.
- 4.10. Atributos de XML.
- 4.11. Nombres XML.
- 4.12. Referencias y entidades de documentos XML.

- 4.13. Entidades.
- 4.14. Creación de documentos XML.
- 4.15. Comentarios en XML.
- 4.16. Declaración de documentos XML.
- 4.17. Comprobación de documentos XML.
- 4.18. HTML.
- 4.19. ¿Qué es XHTML?
- 4.20. Diferencias entre HTML y XML.
- 4.21. Elementos XHTML.
- 4.22. Atributos XHTML.
- 4.23. Documentos XHTML
- 4.24. CSS.
- 4.25. Sintaxis y Selectores CSS.
- 4.26. Hojas de estilo en CSS.
 - 4.26.1. CSS externo.
 - 4.26.2. CSS interno.
 - 4.26.3. CSS en línea.
- 4.27. Comentarios en CSS.
- 4.28. Colores.
- 4.29. Fondos.
- 4.30. Márgenes y Bordes.
- 4.31. Relleno y Altura en CSS.
- 4.32. Modelos de Caja en CSS.
- 4.33. Esquemas.
- 4.34. Texto y Fuentes en CSS.
- 4.35. Iconos y Enlaces.
- 4.36. Listas y tablas.
- 4.37. Pantalla CSS.
- 4.38. Barras de Navegación.
- 4.39. Menú Desplegable.
- 4.40. Galería de imágenes.
- 4.41. Formularios.
- 4.42. Diseño de Sitio Web.

Tecnología Vocacional II

Quinto Grado

1.1 Formulario de Windows

Por: Edwin Alvarez

System.Windows.Forms contiene clases para crear aplicaciones basadas en ventanas que aprovechan las ventajas de las características avanzadas de la interfaz de usuario disponibles en el sistema operativo Microsoft Windows.



Fuente: <https://lh3.googleusercontent.com/3xJdDaMmrKsMiHxIctWLzk3Swb7QiyIVh9aqYIZ5RpTGJ7osTI5js370QVCUKQijIS-9SEeR2BJ9sPsFIIZNXbIfjLK52fR2mIxDezB>

Descripción: Se ejemplifica lo que es una login para poder ingresar al sistema.

1.1.1 Controles Windows

GroupBox: Componente que muestra un marco alrededor de un grupo de controles con un título opcional.

Button: Componente que permite implementar un Botón de pulsación.

Label: Proporciona información en tiempo de ejecución o texto descriptivo para un control.

TextBox: Componente que permite ingresar una línea de texto y editarlo.

PictureBox: Componente permite mostrar una imagen en el formulario.

MenuStrip: Permite implementar un menú de ítems y subítems de opciones en formularios Windows.

DataGridView: Muestra filas y columnas de datos en una cuadrícula que se puede personalizar.



Fuente: https://www.wikihow.com/images_en/thumb/5/57/Create-a-Print-Preview-Control-in-Visual-Basic-Step-2-Version-2.jpg/v4-460px-Create-a-Print-Preview-Control-in-Visual-Basic-Step-2-Version-2.jpg

Descripción: El cuadro de herramientas de los controladores que se pueden usar.

1.1.2 Diferencias entre Access y SQL

Access y SQL son dos tecnologías diferentes que se utilizan en el manejo y gestión de bases de datos.

Tipo de base de datos: Access es un sistema de gestión de bases de datos relacionales de escritorio que se utiliza principalmente en entornos de usuario individual. SQL, por otro lado, es un lenguaje de programación utilizado en el manejo de bases de datos relacionales de servidor.

Seguridad: SQL ofrece una seguridad más robusta que Access. SQL cuenta con herramientas de autenticación y autorización de usuarios, así como medidas de seguridad a nivel de servidor para proteger los datos almacenados en la base de datos. Access, por otro lado, tiene menos opciones de seguridad y está más enfocado en la seguridad de la aplicación de escritorio.

Costo: Access es más accesible en términos de costo, ya que se incluye en la suite de Microsoft Office y no requiere una gran inversión adicional en hardware y software. SQL, por otro lado, es una tecnología más costosa y requiere de un servidor dedicado y licencias de software adicionales.



Fuente: <https://geeks.ms/access/wp-content/uploads/sites/78/2016/04/SServerAccess.png>

Descripción: Logotipo de Access y SQL

1.1.3 Conexión SQL server 2014

Lo primero que hacemos es incluir el espacio de nombres: `using System.Data.SqlClient;` Este espacio de nombres define la clase `SqlConnection`. Para el evento `Clic` del `button1` implementamos primero la creación de un objeto de tipo `SqlConnection`: `SqlConnection conexion = new SqlConnection("server=DIEGO-PC ; database=base1 ; integrated security = true");` Abrimos la conexión: `conexion.Open();` Mostramos un mensaje que la conexión se abrió: `MessageBox.Show("Se abrió la conexión con el servidor SQL Server y se seleccionó la base de datos");` Cerramos la conexión: `conexion.Close();` Mostramos finalmente un mensaje del cierre de la conexión: `MessageBox.Show("Se cerró la conexión.");`

```

        Password="1234",
        InitialCatalog="1234"
    };
    using (SqlConnection connection = new SqlConnection(builder.ConnectionString))
    {
        Console.WriteLine("Conexión a base de datos");
        Console.WriteLine("-----");
        string query = "SELECT * FROM Alumno";
        using (SqlCommand cmd = new SqlCommand(query, connection))
        {
            connection.Open();
            using (SqlDataReader reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    Console.WriteLine($"{reader.GetString(0)}");
                }
            }
        }
    }
}

```

Fuente: <https://i.ytimg.com/vi/TineabHxDig/maxresdefault.jpg>

Descripción: Código demostrativo de la conexión en Visual Studio.

1.1.4 CRUD

CRUD es un acrónimo que se refiere a las operaciones básicas de gestión de datos que se realizan en la mayoría de los sistemas de gestión de bases de datos. CRUD significa:

Create (crear): operación para crear nuevos registros en una base de datos.

Read (leer): operación para leer y consultar los registros existentes en una base de datos.

Update (actualizar): operación para actualizar los registros existentes en una base de datos.

Delete (eliminar): operación para eliminar registros existentes en una base de datos.

Estas operaciones básicas son la base de la funcionalidad de muchos sistemas de gestión de bases de datos y se utilizan comúnmente en aplicaciones web y móviles para realizar acciones como crear, leer, actualizar y eliminar datos de una base de datos.

ID	Nombre	Categoría	Precio
100001	Arroz	Cereales	1000
100002	Pasta	Verduras	1200
100003	Leche	Lácteos	800
100004	Carne	Carnes	1500
100005	Huevos	Lácteos	600
100006	Pan	Panadería	400
100007	Aceite	Aceites	900
100008	Sal	Condimentos	300
100009	Azúcar	Condimentos	500
100010	Vinagre	Condimentos	700
100011	Miel	Lácteos	1100
100012	Jugo	Bebidas	1300
100013	Agua	Bebidas	200

Fuente: <https://rjcodeadvance.com/wp-content/uploads/2022/04/CRUD-with-MVP-WinForms-SQL-C-930x620.png>

Descripción: Ejemplificación de un CRUD con funciones.

1.6 Codificación Avanzada

Por: Diego Alvizures

La codificación avanzada es aquella que lleva ciertos elementos que ya no se realizan de forma secuencial, sino que llevan mucho mas proceso y análisis de parte del desarrollador. Un ejemplo de la codificación avanzada es la programación orientada a objetos, ya que muchas veces se trabaja con clases o múltiples archivos para poder realizar un proyecto funcional.

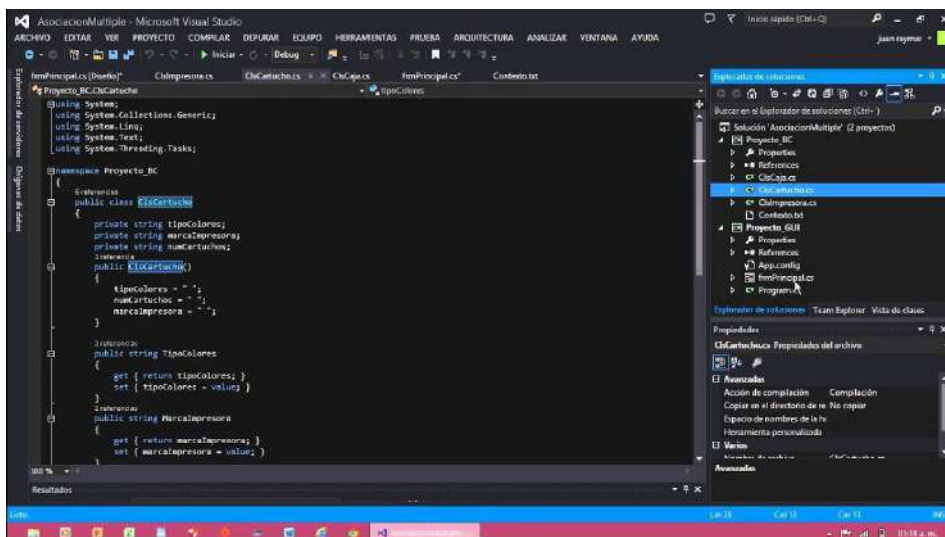


Imagen: Ejemplo de una codificación avanzada orientada a objetos

Fuente: https://www.youtube.com/watch?v=gla29-O_euo

1.7 Anti errores y Try Catch

Es un sistema que apunta a ser sencillo y simple. Se enfatiza en realizar cosas obvias, en las que detecta errores o evita que se cometan. El objetivo final es concretar un proceso o terminar un producto sin la posibilidad que de exista un defecto.

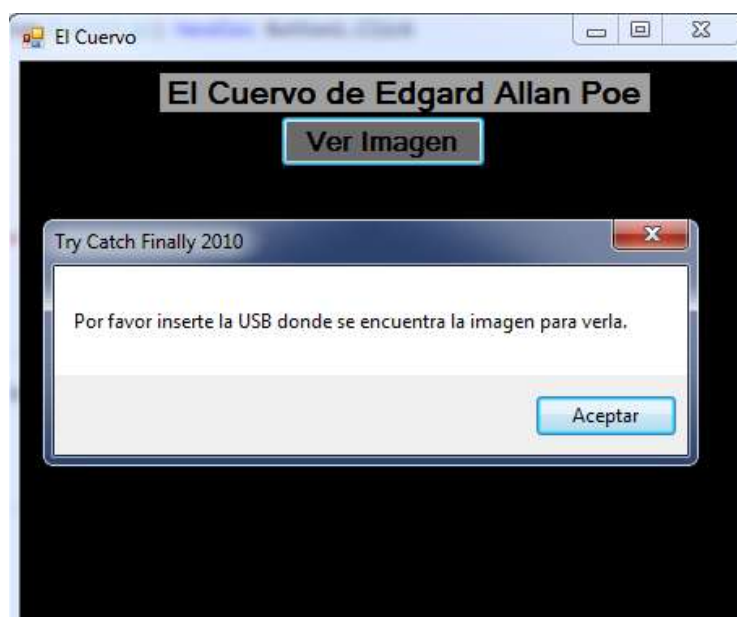


Imagen: Ejemplo de un anti-error al ingreso de una USB)

Fuente: <https://www.aulafacil.com/cursos/programacion/visual-basic-paso-a-paso/errores-en-visual-basic-l42178>

1.7.1 Try Catch

La declaración try...catch señala un bloque de instrucciones a intentar (try), y especifica una respuesta si se produce una excepción (catch). Deben estar presentes las llaves para iniciar la sentencia, tanto de inicio como de fin.

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Try
            Dim numero As Integer
            numero = TextBox1.Text
            Label1.Text = "El valor ingresado es: " & numero
        Catch ex As Exception
            MsgBox("Error al ingresar el dato.")
        End Try
    End Sub
End Class
```

Imagen: Ejemplo de un try-catch realizado en Visual Basic
 Fuente: <https://www.youtube.com/watch?v=usXgQiPjIOI>

1.8 Enlace a SQL Server.

Para realizar el enlace con SQL Server se deben hacer los siguientes pasos:

1. Vamos a irnos a origen de datos y vamos a seleccionar la opción de base de datos.

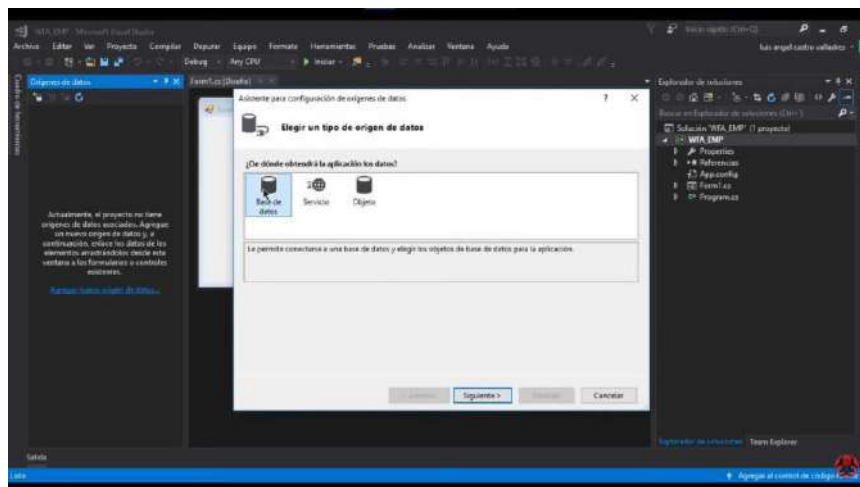


Imagen: Ejemplo de como conectar SQL Server a Visual Basic
 Fuente: <https://www.youtube.com/watch?v=CmWleAx4CFw&t=202s>

2. Cuando nos aparezca la siguiente ventana vamos a seleccionar conjunto de datos.

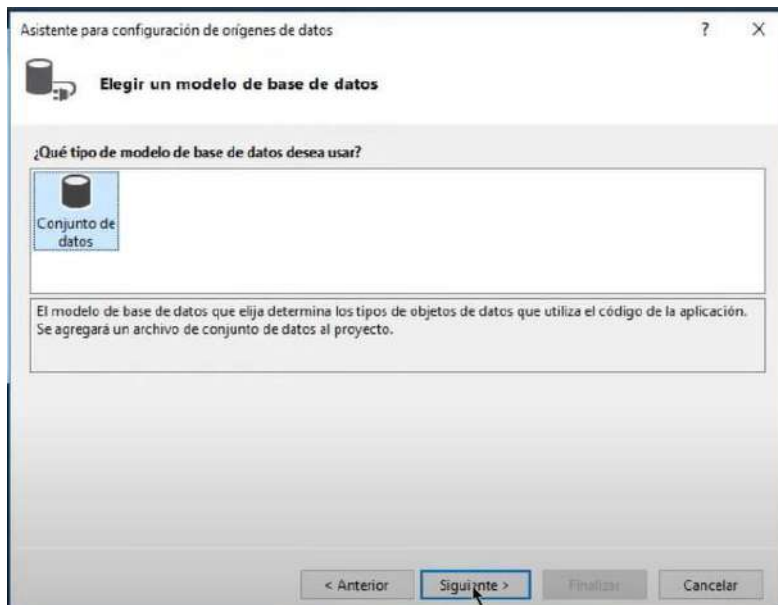


Imagen: Ejemplo de como conectar SQL Server a Visual Basic
Fuente: <https://www.youtube.com/watch?v=CmWleAx4CFw&t=202s>

- 3. Luego de realizar eso vamos a seleccionar nueva conexión y daremos click en el origen de datos Microsoft SQL Server.

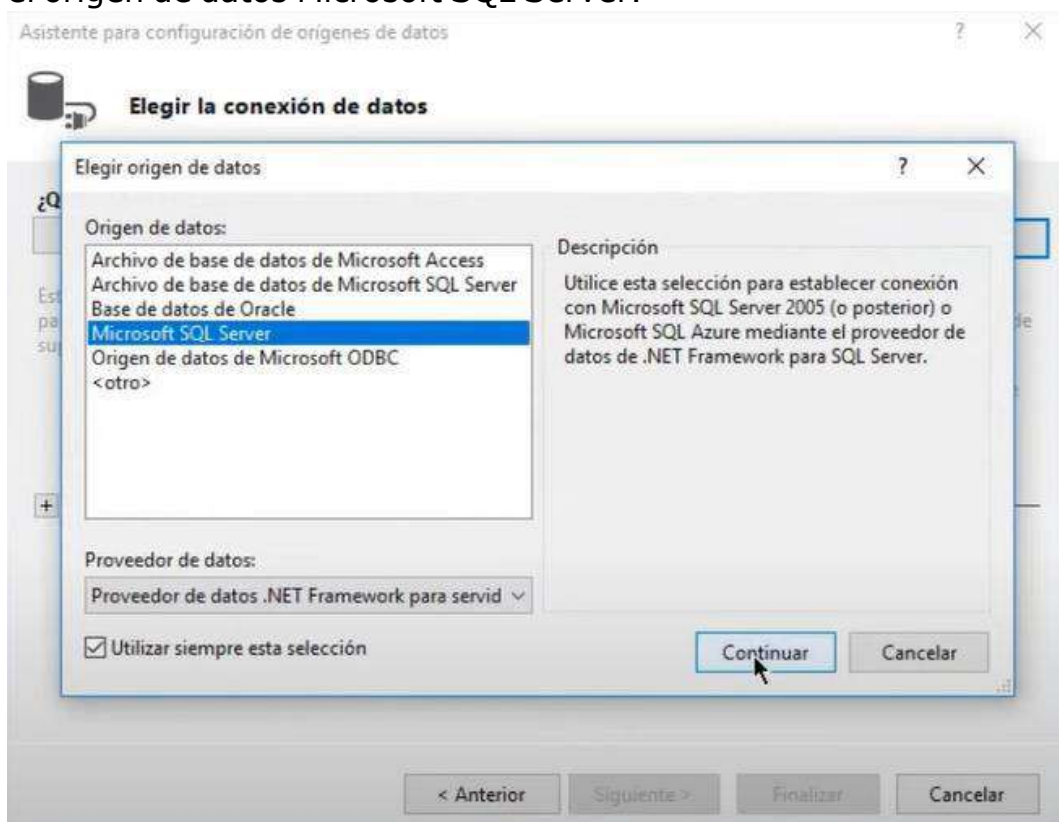


Imagen: Ejemplo de como conectar SQL Server a Visual Basic
Fuente: <https://www.youtube.com/watch?v=CmWleAx4CFw&t=202s>

- Seguido de eso nos aparecerá la siguiente ventana y tendremos que seleccionar nuestro servidor (Esto aparecerá automáticamente si tenemos iniciado nuestro SQL Server en nuestro ordenador).

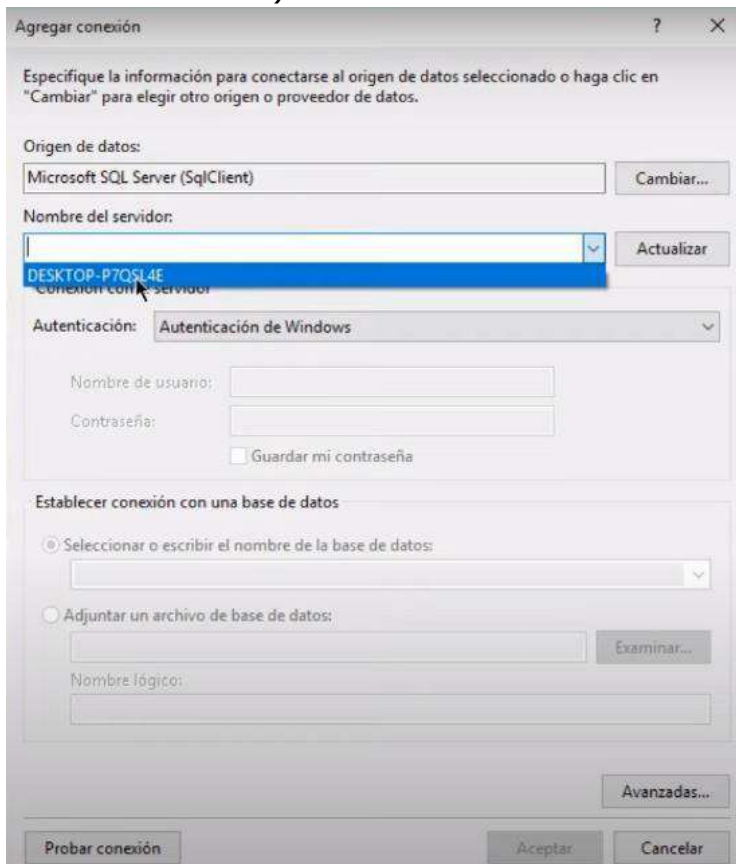


Imagen: Ejemplo de como conectar SQL Server a Visual Basic
 Fuente: <https://www.youtube.com/watch?v=CmWleAx4CFw&t=202s>

- Seguido de eso vamos a seleccionar nuestra base de datos y daremos click en probar conexión.

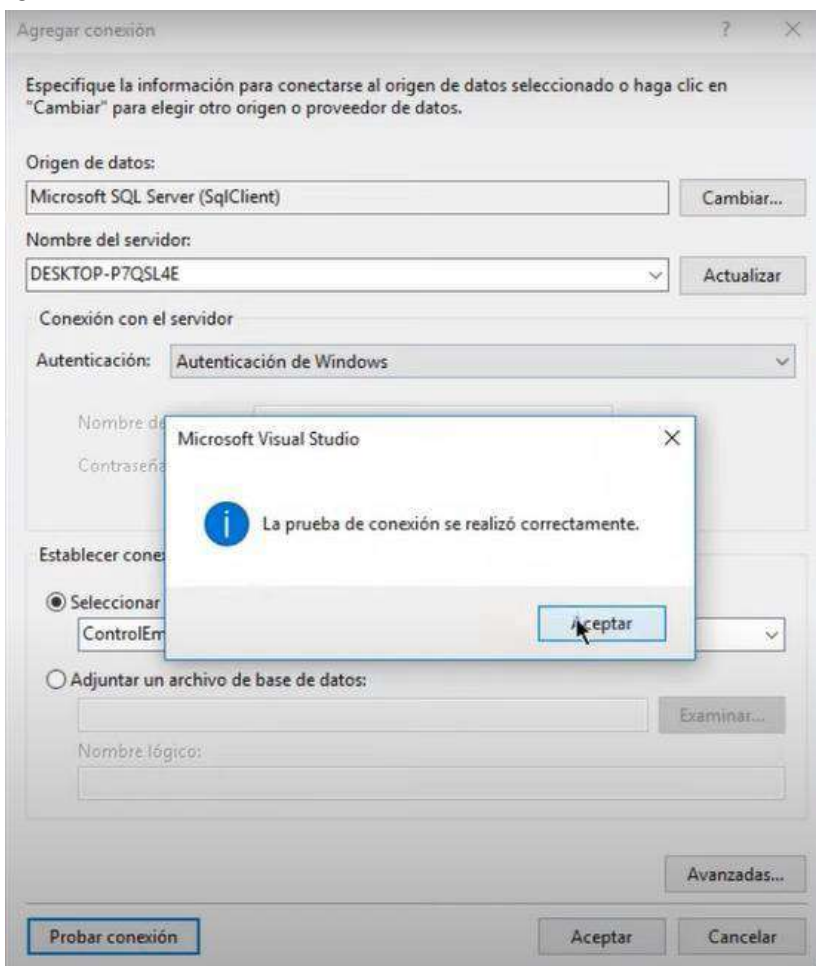


Imagen: Ejemplo de cómo conectar SQL Server a Visual Basic
 Fuente: <https://www.youtube.com/watch?v=CmWleAx4CFw&t=202s>

- Seguido de eso simplemente vamos a seleccionar nuestra base de datos y daremos click en siguiente.

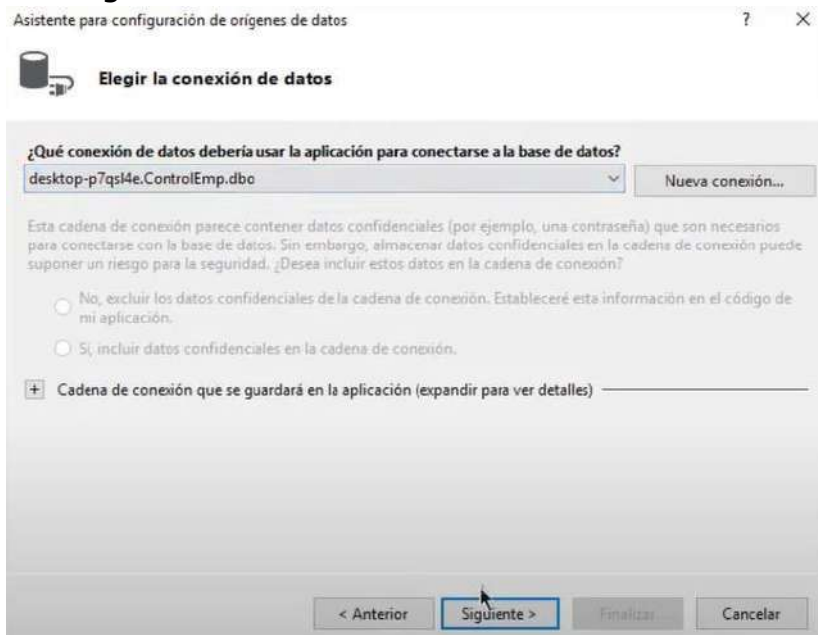


Imagen: Ejemplo de como conectar SQL Server a Visual Basic
 Fuente: <https://www.youtube.com/watch?v=CmWleAx4CFw&t=202s>

- Por último, solo seleccionaremos nuestras tablas de la base de datos para completar el proceso.

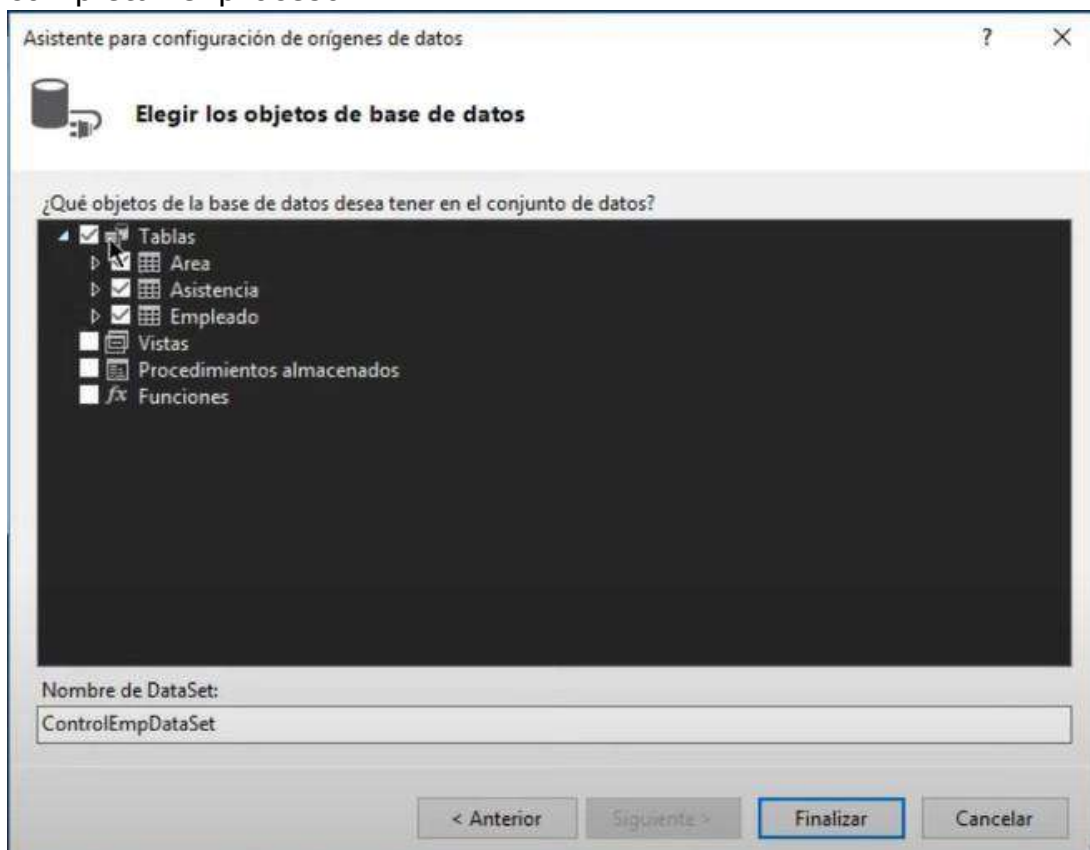


Imagen: Ejemplo de como conectar SQL Server a Visual Basic
 Fuente: <https://www.youtube.com/watch?v=CmWleAx4CFw&t=202s>

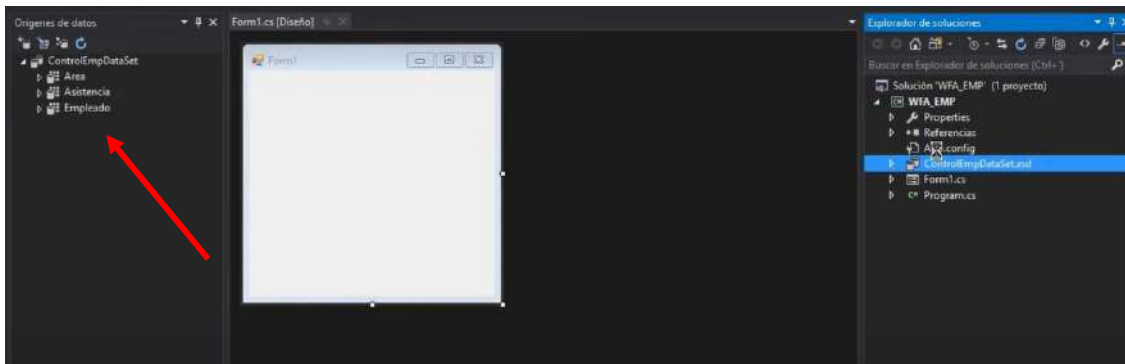


Imagen: Ejemplo de como conectar SQL Server a Visual Basic
 Fuente: <https://www.youtube.com/watch?v=CmWleAx4CFw&t=202s>

1.9 Diseño de Reportes

El diseño de reportes es una parte muy importante para nuestro reporte final, ya que definirá como será su presentación y como se adecuará a las necesidades diversas que se tengan en ese informe.

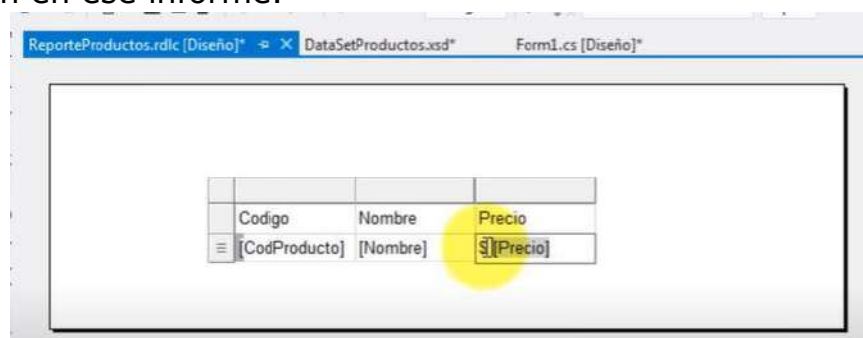


Imagen: Ejemplo de cómo diseñar un Reporte en Visual Studio
 Fuente: <https://www.youtube.com/watch?v=0oCXR5VNzPY>

1.9.1 Herramientas para el diseño

Para el diseño de un reporte tenemos diversas herramientas entre las más utilizadas están:

- **Tabla:** Sirve para colocar los datos de nuestra Base de Datos.
- **Lista:** Para generar una lista ordenada de diversos datos.
- **Cuadro de texto:** Sirve para crear títulos o diversos textos que le darán forma a nuestro reporte.
- **Imagen:** Sirve para darle un diseño más personalizado a nuestro reporte.



Imagen: Ejemplo de cómo diseñar un Reporte en Visual Studio
 Fuente: <https://www.youtube.com/watch?v=0oCXR5VNzPY>

1.10 Campos calculados.

Un campo calculado es un campo que no se almacena físicamente en la tabla. SQL Server emplea una fórmula que detalla el usuario al definir dicho campo para calcular el valor según otros campos de la misma tabla.

1.10.1 Como crear un campo calculado:

Para el siguiente ejemplo vamos a definir un campo calculado que en este caso cuanto se tiene que sumar a el empleado por la cantidad abonada por los hijos (100 por cada hijo).

```
create table empleados(
  documento char(8),
  nombre varchar(10),
  domicilio varchar(30),
  sueldobasico decimal(6,2),
  cantidadhijos tinyint default 0,
  sueldototal as sueldobasico + (cantidadhijos*100)
);
```

Imagen: Ejemplo de cómo realizar un campo calculado con SQL

Fuente: <https://www.tutorialesprogramacionya.com/sqlserverya/temarios/descripcion.php?cod=92>

En este caso crea una operación que definirá el sueldo total del empleado con la cantidad que se sumará. Esto es muy útil para realizar operaciones entre campos por si quieres determinar ciertos datos, en su mayoría contables

1.1 VISUAL STUDIO .NET POO. Y REPORTERIA DE DATOS .NET

Por: David Nehemías Bautista Patzán

1.1.1. Instalación de ReportViewer.

1. Descarga el paquete de instalación de Microsoft ReportViewer desde el sitio web oficial de Microsoft.
2. Ejecuta el archivo de instalación descargado y selecciona la opción "Instalar" para comenzar la instalación.
3. Asegúrate de que la opción "ReportViewer" está seleccionada en la lista de componentes que se instalarán y haz clic en el botón "Siguiente".
4. Acepta los términos del acuerdo de licencia y haz clic en "Siguiente".
5. Selecciona la carpeta donde deseas instalar ReportViewer y haz clic en "Siguiente".
6. Haz clic en "Instalar" para comenzar la instalación.
7. Una vez que la instalación esté completa, abre Visual Studio .NET y crea un nuevo proyecto.
8. Haz clic con el botón derecho en el Explorador de soluciones y selecciona "Agregar nuevo elemento".
9. En el cuadro de diálogo "Agregar nuevo elemento", selecciona "Informe" y haz clic en "Agregar".
10. Selecciona el tipo de informe que deseas crear y haz clic en "Agregar".
11. Se abrirá la ventana de diseño de informes de ReportViewer, donde podrás diseñar tu informe.
12. Cuando hayas terminado de diseñar tu informe, guarda los cambios y compila el proyecto.

Ahora deberías ser capaz de ver y utilizar ReportViewer en tu proyecto de Visual Studio .NET.

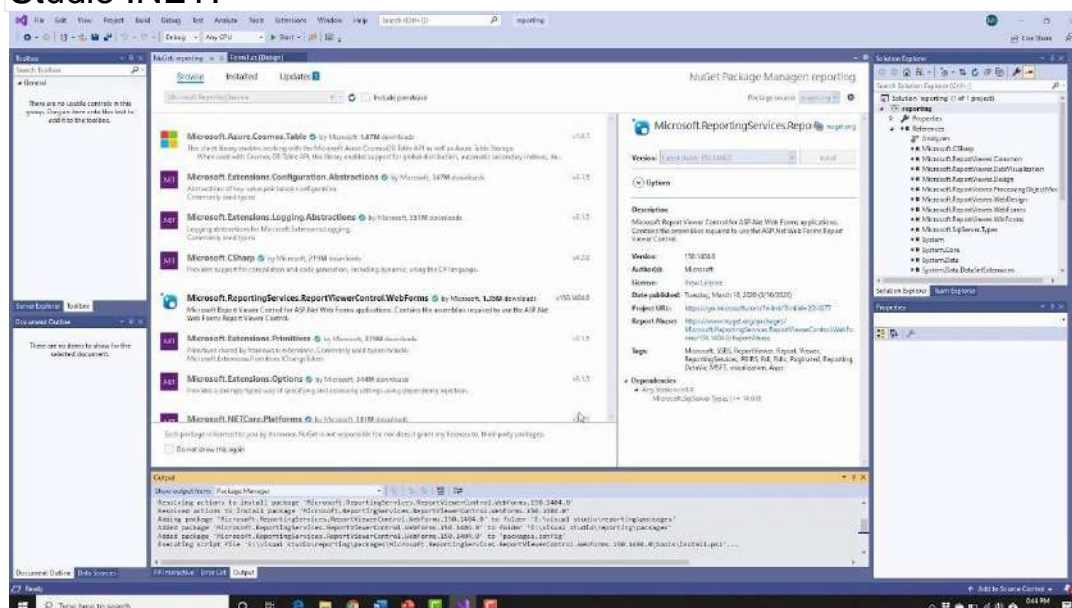


Imagen: Instalando ReportViewer en visual studio.NET
 Fuente: <https://i.ytimg.com/vi/UpKGA8iFCnk/maxresdefault.jpg>

1.1.2. Instalación de Crystal Report.

1. Descarga la última versión de Crystal Reports para Visual Studio desde el sitio web oficial de SAP:
<https://www.sap.com/products/crystal-reports/developer-downloads.html>
2. Ejecuta el archivo descargado y selecciona la opción "Instalar Crystal Reports para Visual Studio".
3. Acepta los términos de la licencia y selecciona la ubicación donde deseas instalar Crystal Reports.
4. Selecciona los componentes que deseas instalar y haz clic en "Siguiente".
5. Espera a que se complete la instalación y haz clic en "Finalizar".
6. Abre Visual Studio y crea un nuevo proyecto.
7. Haz clic con el botón derecho del ratón en la carpeta "Referencias" y selecciona "Agregar referencia".
8. Selecciona "Crystal Reports" en la pestaña "Examinar" y haz clic en "Aceptar".
9. Ahora puedes utilizar Crystal Reports en tu proyecto de Visual Studio.

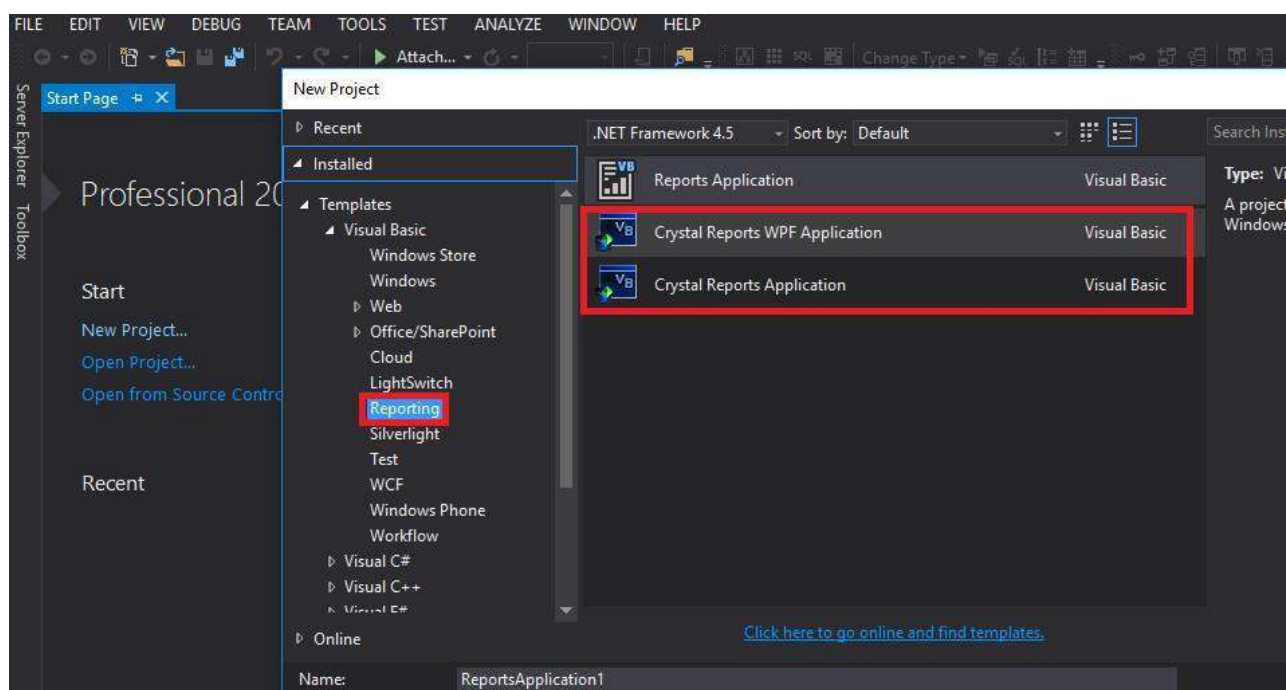


Imagen: Instalando Crystal Reports en visual studio.NET

Fuente: <https://2.bp.blogspot.com/->

[EDWNH2I39rM/WzplRi8NkBI/AAAAAAAAAFQ/rU9P4U2odiAUgyzI9MCCBPpEDR9E37dPACLcBGAs/s1600/cr5.jpg](https://2.bp.blogspot.com/-EDWNH2I39rM/WzplRi8NkBI/AAAAAAAAAFQ/rU9P4U2odiAUgyzI9MCCBPpEDR9E37dPACLcBGAs/s1600/cr5.jpg)

1.1.3 ReportViewer

ReportViewer es un control de interfaz de usuario en Visual Studio .NET que permite a los desarrolladores crear informes y presentar datos de manera visual. Este control se utiliza normalmente en aplicaciones empresariales y de gestión de datos para mostrar información en forma de tablas, gráficos, diagramas, etc. Es un paradigma de programación que se centra en la creación de objetos que interactúan entre sí para realizar una tarea. En el contexto de Visual Studio .NET, la POO se refiere al uso de clases y objetos para construir aplicaciones. La programación orientada a objetos permite crear código modular y reutilizable, lo que facilita el mantenimiento y la extensión de las aplicaciones a medida que evolucionan con el tiempo.

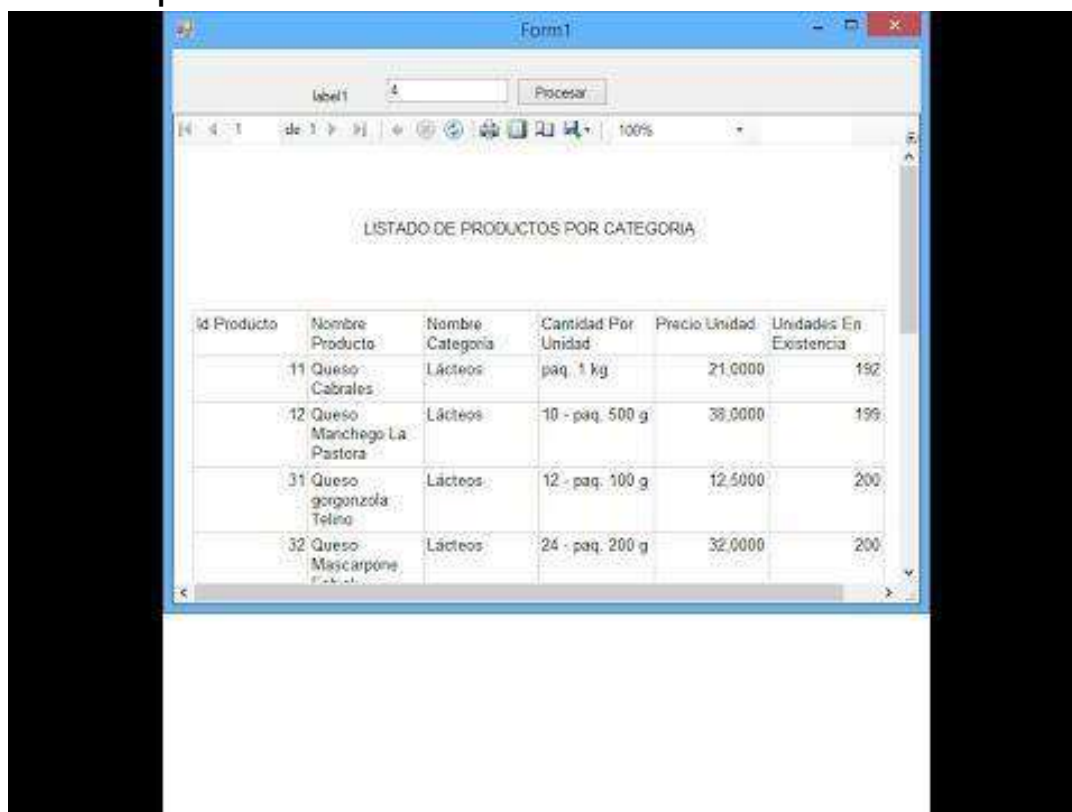


Imagen: Ejemplo aplicando ReportViewer

Fuente: <https://i.ytimg.com/vi/DdPTx4ekFp4/hqdefault.jpg>

1.1.4 Relacionar y Diseñar.

En resumen, relacionar y diseñar sobre el tema de Visual Studio .NET implica entender cómo esta plataforma de desarrollo de software permite a los desarrolladores crear aplicaciones a través de diferentes lenguajes de programación, ofreciendo una interfaz gráfica intuitiva, Los conjuntos de datos que contienen tablas de datos relacionadas usan objetos DataRelation para representar una relación primaria/secundaria entre las tablas y devolver registros relacionados entre sí. Al agregar tablas relacionadas a conjuntos de datos mediante el Asistente para configuración del origen de datos o el Diseñador de DataSet, se crea y configura el objeto DataRelation automáticamente.

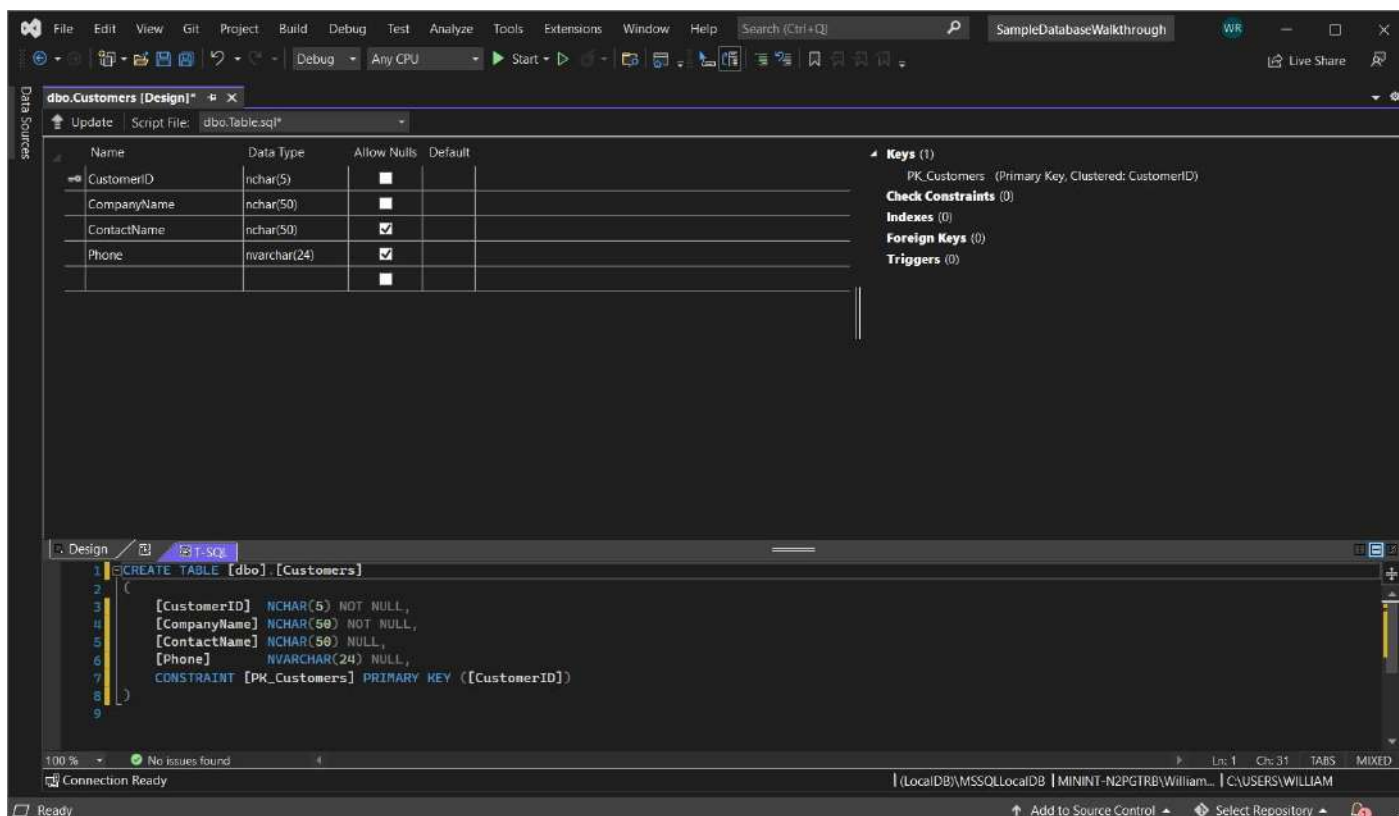


Imagen: Ejemplo del tema relacionar y diseñar

Fuente: <https://learn.microsoft.com/es-es/visualstudio/data-tools/media/vs-2022/table-designer.png?view=vs-2022>

1.1.5 Impresión de Reports.

La impresión de informes o Reports en Visual Studio .NET orientado a objetos (POO) es una funcionalidad que permite generar informes y documentos a partir de datos almacenados en una base de datos o en otras fuentes de datos.

La impresión de informes en Visual Studio .NET se realiza utilizando el componente ReportViewer, el cual permite diseñar y visualizar informes en tiempo de diseño y en tiempo de ejecución. Este componente también proporciona una amplia gama de opciones de formato, como tablas, gráficos, imágenes y textos, para personalizar el informe según las necesidades del usuario.

Para generar un informe en Visual Studio .NET, se debe seguir los siguientes pasos:

1. Crear un nuevo informe: Desde el menú "Archivo", seleccionar "Nuevo" y luego "Informe". Seleccionar el tipo de informe que se desea crear.
2. Configurar el origen de datos: Seleccionar la fuente de datos que se utilizará para generar el informe. Esta puede ser una base de datos, un archivo XML, un servicio web, entre otros.
3. Diseñar el informe: Utilizar el diseñador de informes para agregar elementos al informe, como tablas, gráficos, imágenes y textos. Se puede personalizar el diseño para que se ajuste a las necesidades del usuario.
4. Vista previa del informe: Visualizar el informe para verificar que el diseño y los datos sean correctos.
5. Imprimir el informe: Seleccionar la opción de impresión en el visor de informes para imprimir el informe.

En resumen, la impresión de informes en Visual Studio .NET POO es una funcionalidad muy útil para generar informes y documentos personalizados a partir de datos almacenados en diferentes fuentes de datos.

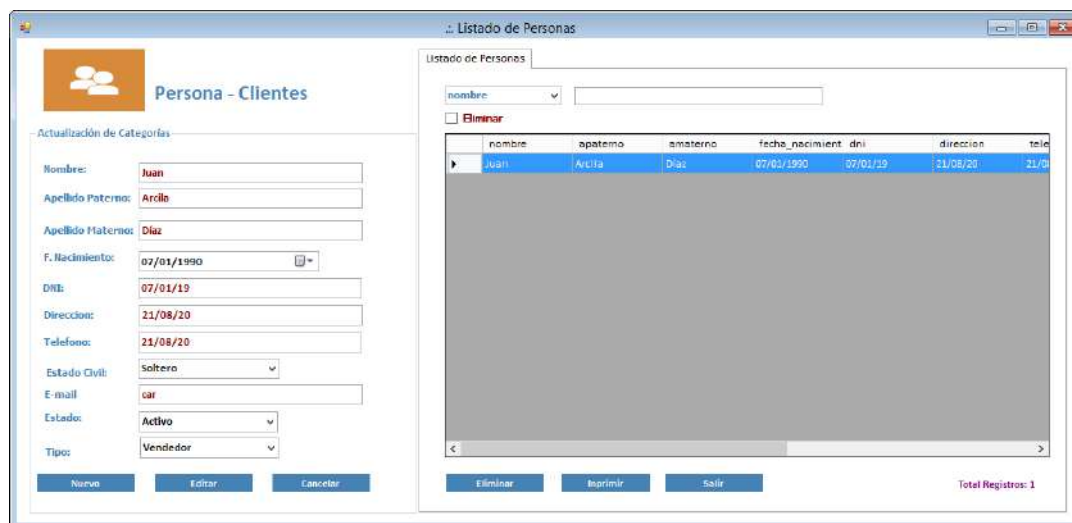


Imagen: Un ejemplo de cómo quedaría la impresión de informes

Fuente: <https://2.bp.blogspot.com/>

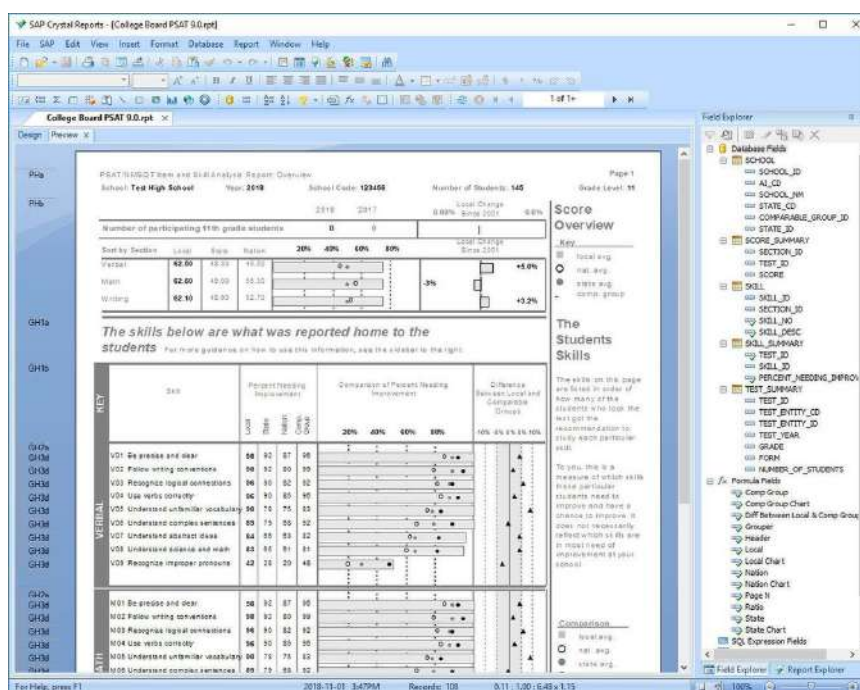
[90oXyuPknP8/U5pjF2USc0I/AAAAAAAAATQ/xWELoCqTjTM/s1600/clientes.png](https://2.bp.blogspot.com/90oXyuPknP8/U5pjF2USc0I/AAAAAAAAATQ/xWELoCqTjTM/s1600/clientes.png)

1.16 Crystal Report

Por: Cristian Alexander Coromac López

Es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos (bases de datos).

Varias aplicaciones, como Microsoft Visual Studio, incluyen una versión OEM de Crystal Reports como una herramienta de propósito general para informes/reportes. Crystal Reports se convirtió en el escritor de informes estándar cuando Microsoft lo liberó con Visual Basic.

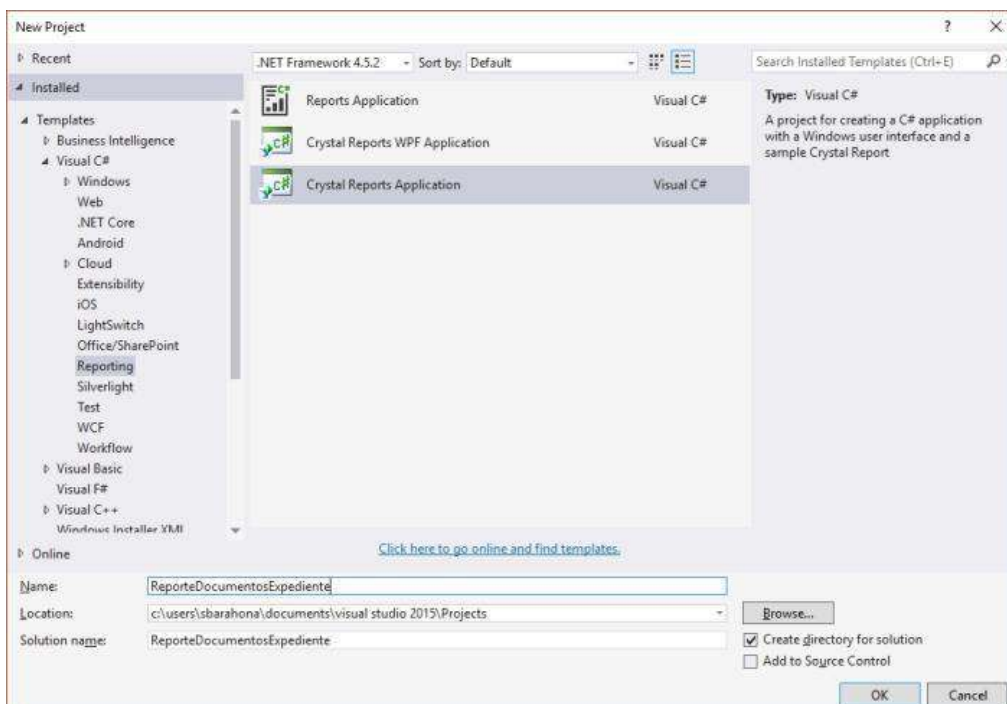


Fuente: [https://es.wikipedia.org/wiki/Crystal Reports](https://es.wikipedia.org/wiki/Crystal_Reports)

1.17 Relacionar y Diseñar

En Visual Studio es necesario crear un proyecto dentro del cual se crea el reporte a utilizar en Evolution (archivo rpt), pero una vez creado el archivo rpt éste se puede modificar directamente con Visual Studio

- Ingresar a Visual Studio y dar File -> New -> Project
- Seleccionar Installed -> Templates -> Visual C# -> Reporting
 - Se podría seleccionar «Visual Basic» pero para efectos prácticos es irrelevante dado que no es necesario modificar la forma que se crea automáticamente
- Seleccionar «Crystal Reports Application»
- Ingresar como nombre del proyecto «ReporteDocumentosExpediente»



Fuente: <https://aseinfoblog.wordpress.com/2017/07/07/guia-para-elaboracion-de-reportes-de-crystal-reports-utilizando-sub-reportes/>

Esta plantilla creará una forma de windows cuyo unico elemento en su cuerpo es un visor de reportes. Tambien creará un archivo de reporte (CrystalReport1.rpt) que será asociado para se muestre en el visor.

1.18 Diferencias entre ReportView y CrystalReport

Report Viewer

Es un control de gama media, ya viene incorporado con visual studio por lo que no hay que pagar por el, es facil e intuitivo.

Report Viewer es de gama media pero su forma de crear es facil e intuitiva, a diferencia de Crystal, no es necesario eliminar todo el reporte para cambiar algo.

El control ReportViewer usa archivos *.rdlc.

ReportViewer pertenece al namespace Microsoft.Reporting.WinForms.

Crystal Reports

Es muy poderoso, pero tiene un precio; es complejo y no siempre hace lo que quieres.

Crystal Reports muy bueno para diseñar, facil, intuitivo, pero para hacer un cambio tenemos que hacer completamente todo.

El archivo de reportes de CrystalReports es *.rpt.

CrystalReportViewer pertenece al namespace CrystalDecisions.Windows.Forms

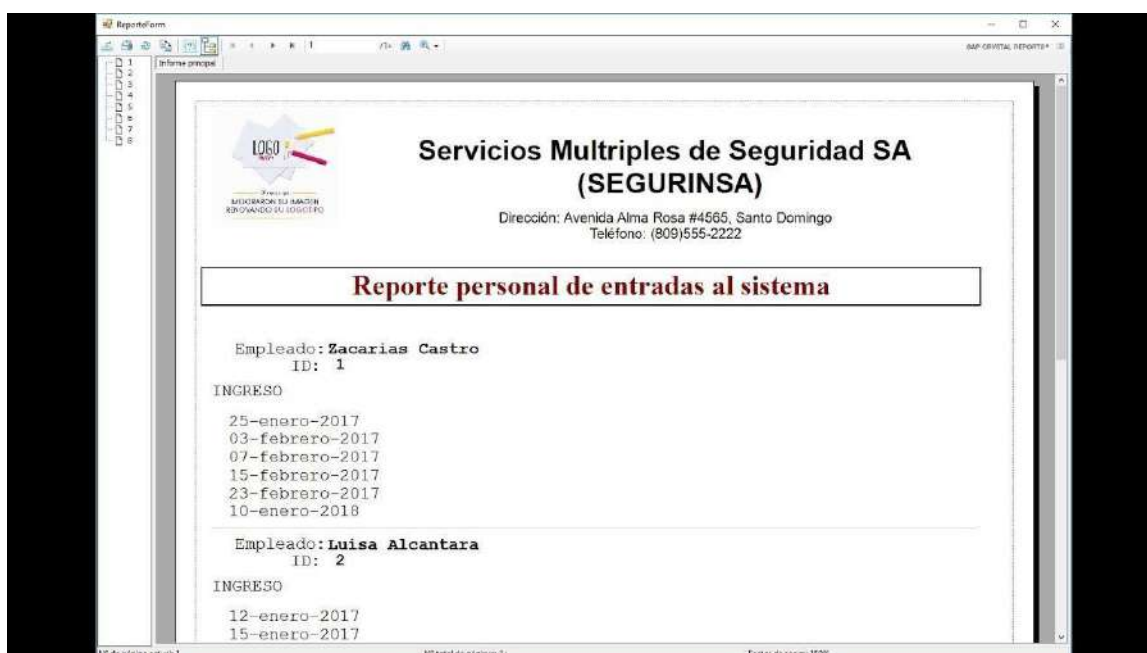
Daniel



Fuente: <https://www.youtube.com/watch?v=N7NGEOhTbEM>

1.19 Impresión de Reportes

Para imprimir un reporte creado en crystal report, tienes que crear un formulario y en el insertar una herramienta que se llama CrystalReportViewer y en la propiedad Report Source asignarle la dirección en la que se encuentra el archivo.rpt que normalmente se crea en la carpeta donde estan haciendo el programa.



Fuente: <https://www.youtube.com/watch?v=I3JM6ai44Mc>

2.1 Entorno de trabajo Ide's

Es una aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo de software.

Normalmente, un IDE consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen auto-completado inteligente de código (*IntelliSense*). Algunos IDE contienen un compilador, un intérprete, o ambos, tales como NetBeans y Eclipse; otros no, tales

como SharpDevelop y Lazarus. El límite entre un IDE y otras partes del entorno de desarrollo de software más amplio, no está bien definido.



Fuente: <https://www.campusmvp.es/recursos/post/Los-mejores-entornos-de-desarrollo-para-Java.aspx>

1.1 Introducción a Java

Por: Willian Estuardo Cuevas Rivas

Java es un lenguaje de programación orientado a objetos que nos permite el desarrollo de aplicaciones básicas, pasando por aplicaciones empresariales hasta aplicación oficiales de móviles.



Imagen: Ejemplo de la representación de Java

Fuente: <http://www.jtech.ua.es/historico/paj/restringido/apuntes/sesion01-apuntes.htm>

1.1.1 ¿Qué es Java?

Por: Willian Estuardo Cuevas Rivas

Java es una plataforma informática de programación que fue creada por Sun Microsystems en 1995. A lo largo de los años, Java se ha ido utilizando en la mayor parte del mundo para la creación de aplicaciones, productos o servicios digitales. Actualmente una gran parte de las aplicaciones que nosotros utilizamos están compuestas por el lenguaje Java, su software es de uso gratuito y puede ser instalado desde cualquier máquina para utilizarlo

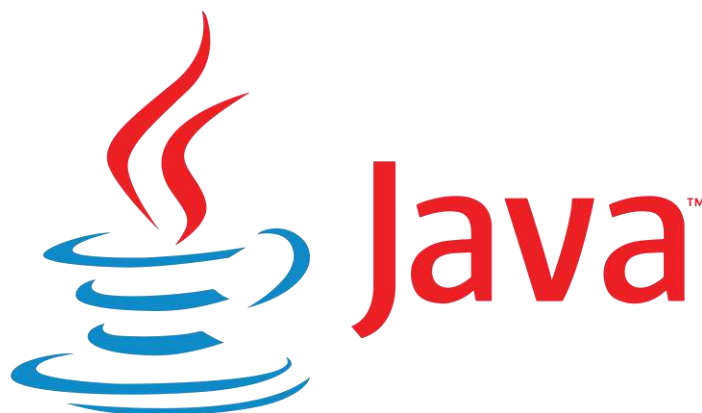


Imagen: Ejemplo del logotipo Java

Fuente: https://www.java.com/es/download/help/whatis_java.html

1.1.2 Instalar Java IDE's

Por: Willian Estuardo Cuevas Rivas

Existen múltiples IDE's para poder ejecutar Java como un lenguaje de programación. Pero el más efectivo o el más utilizado por múltiples usuarios hasta la fecha es "NetBeans", uno de los mejores IDE's gratuitos para trabajar con Java.

- Nos dirigimos al enlace de la [web oficial de NetBeans](#).
- Elegimos el sistema operativo e idioma el cual vamos a trabajar.
- Damos clic en "Download" en la parte izquierda de la primera columna
- Una vez se termine de descargar el archivo lo abrimos como Administrador y damos los pasos de instalación.
- Clic en "Next" y luego marcamos la casilla correspondiente y en "Next"
- Por último esperamos que cargue y clicamos en "Finish"

Con esto ya tendríamos instalado nuestro IDE's para trabajar en Java.

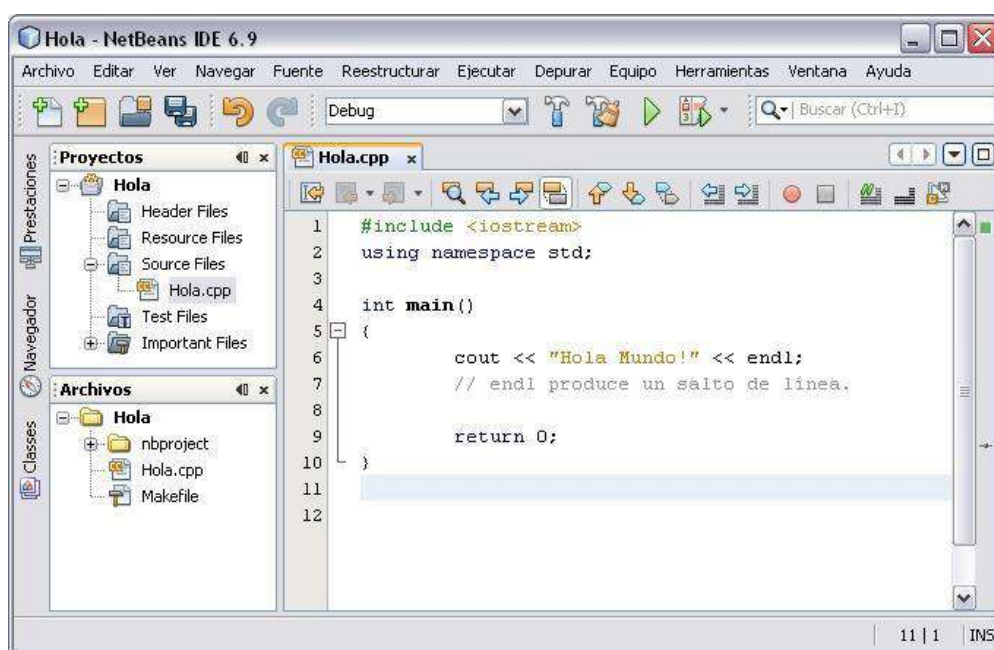


Imagen: Ejemplo de la interfaz de Java

Fuente: <https://unipython.com/instalar-java-netbeans-windows-os-linux/>

1.1.3 Variables de Entorno

Java posee dos variables de entorno que nos permiten configurarlas dentro de Windows. La primera de ellas "Java_Home", la cual es una variable de entorno de sistema que informa al Sistema Operativo sobre la ruta donde se encuentra instalado. La segunda de ellas es la variable "Path", el cual es una variable como la anterior solo que con la diferencia de que esta informa a nuestro ordenador su funcionamiento dentro de la IDE de Java.

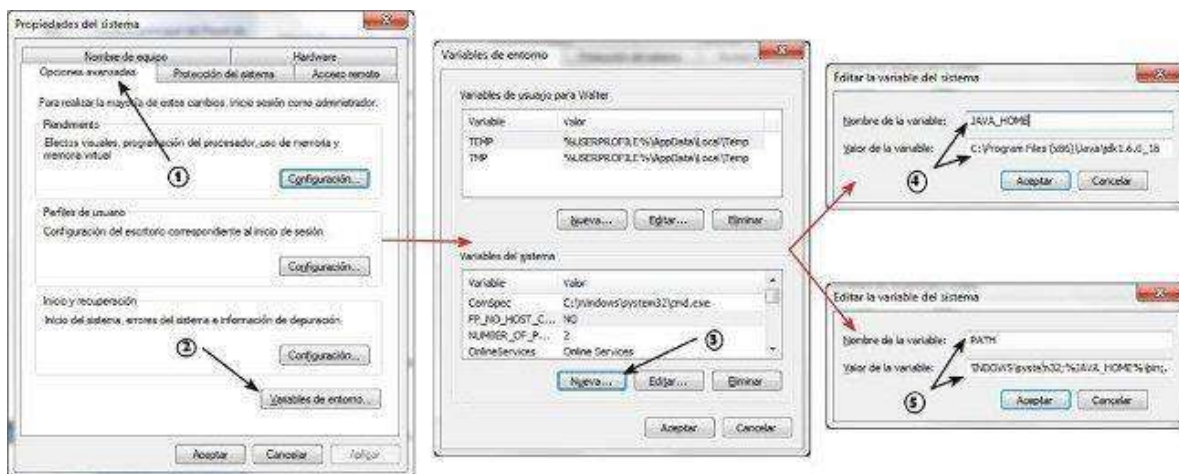


Imagen: Ejemplo de la interfaz de las variables de entorno
 Fuente: <http://www.aprenderaprogramar.com>

1.1.4 Estructura General de un Programa

Java al igual que otros lenguajes de programación utilizan la misma estructura que se compone por datos de entrada que salen en un dispositivo que almacena los diferentes caracteres dentro de una variable para que le demos una acción y de esta manera puedan ser procesados dentro del programa que nosotros vayamos a estar utilizando.

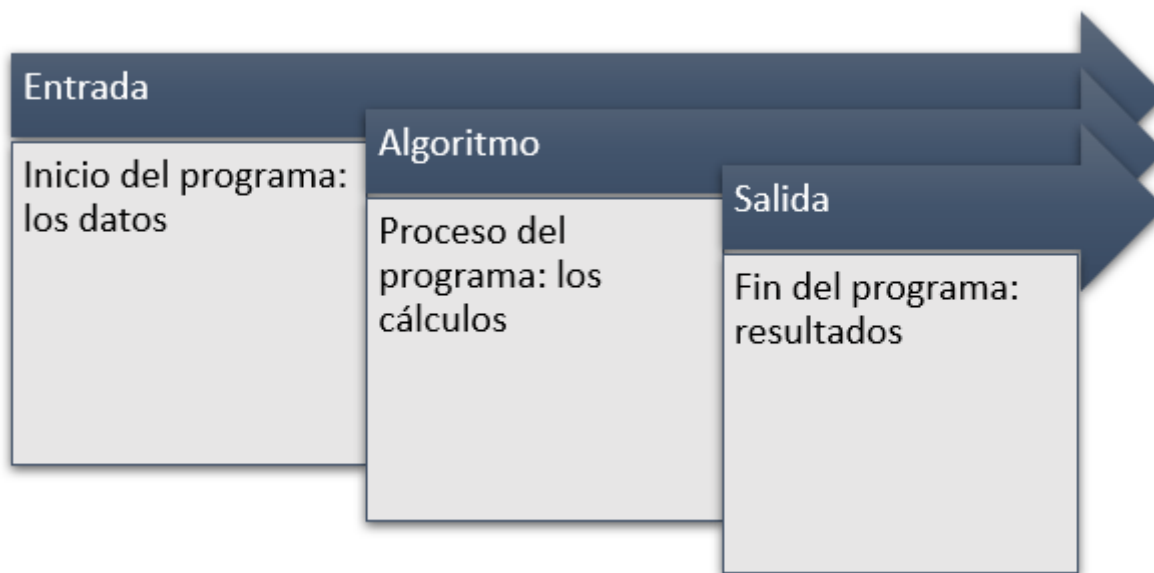


Imagen: Ejemplo de la secuencia de estructura de un programa
 Fuente: <https://i0.wp.com/fernandoarciniega.com/wp-content/uploads/image-169.png?fit=615%2C325&ssl=1>

1.2 Java Consola.

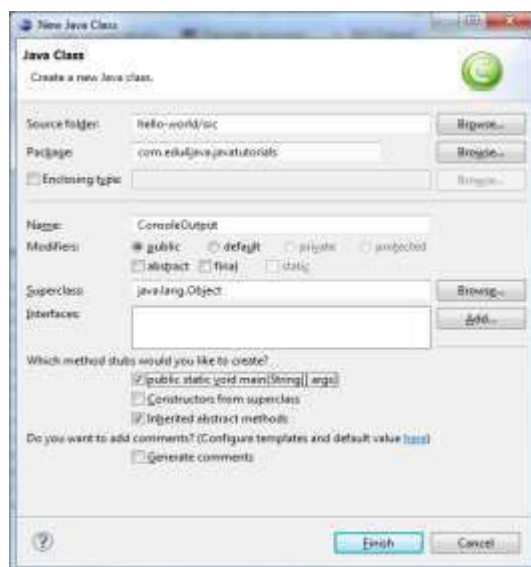
Por: Nohelia Guadalupe Culajay Oxcal

1.2.8 Salida de datos en consola.

Esta comunicación tiene dos sentidos; entrada y salida (input & output):

- **Entrada:** El usuario da información al programa.
- **Salida:** el programa da información al usuario.

Una forma fácil de crear una nueva clase con Eclipse es, botón derecho del ratón sobre src - New - Class



Fuente: <http://www.edu4java.com/es/java/entrada-salida-consola.html#:~:text=Salida%20por%20la%20consola&text=Dar%20el%20nombre%20de%20la,%E2%80%9D%20se%20convirti%C3%B3%20en%20%E2%80%9CSystem.>

1.2.9 Compilador primario de Java.

Un compilador de java es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación. La función de un compilador es convertir un programa escrito un en lenguaje como Java a un lenguaje que la computadora pueda entender.

1.2.10 Variables e Identificadores.

Los identificadores nombran variables, funciones, clases y objetos; cualquier cosa que el programador necesite identificar o usar.

En Java, un identificador comienza con una letra, un subrayado (_) o un símbolo de dólar (\$). Los siguientes caracteres pueden ser letras o dígitos. Se distinguen las mayúsculas de las minúsculas y no hay longitud máxima.

Serían identificadores válidos:

- 2 identificador
- 3 nombre_usuario
- 4 Nombre_Usuario
- 5 _variable_del_sistema
- 6 \$transaccion

y su uso sería, por ejemplo:

```
int contador_principal;
char _lista_de_ficheros;
float $cantidad_en_Ptas;
```

1.2.11 Tipos de datos.

Un dato siempre lleva asociado un tipo de dato, que determina el conjunto de valores que puede tomar.

En Java toda la información que maneja un programa está representada por dos tipos principales de datos:

- Datos de tipo básico o primitivo.
- Referencias a objetos.

Los tipos de datos básicos o primitivos no son objetos y se pueden utilizar directamente en un programa sin necesidad de crear objetos de este tipo. La biblioteca Java proporciona clases asociadas a estos tipos que proporcionan métodos que facilitan su manejo.

Tipo de dato	Representación	Tamaño (Bytes)	Rango de Valores	Valor por defecto	Clase Asociada
byte	Numérico Entero con signo	1	-128 a 127	0	Byte
short	Numérico Entero con signo	2	-32768 a 32767	0	Short
int	Numérico Entero con signo	4	-2147483648 a 2147483647	0	Integer
long	Numérico Entero con signo	8	-9223372036854775808 a 9223372036854775807	0	Long
float	Numérico en Coma flotante de precisión simple Norma IEEE 754	4	$\pm 3.4 \times 10^{-38}$ a $\pm 3.4 \times 10^{38}$	0.0	Float
double	Numérico en Coma flotante de precisión doble Norma IEEE 754	8	$\pm 1.8 \times 10^{-308}$ a $\pm 1.8 \times 10^{308}$	0.0	Double
char	Carácter Unicode	2	\u0000 a \uFFFF	\u0000	Character
boolean	Dato lógico	-	true ó false	false	Boolean
void	-	-	-	-	Void

1.2.12 Constantes.

En un programa, las constantes se utilizan para almacenar datos (valores) –en la memoria de la computadora– que no podrán cambiar durante la ejecución del programa.

Declarar una constante en Java.

En Java, para declarar una constante se utiliza la palabra reservada final.

```
import java.util.Scanner;

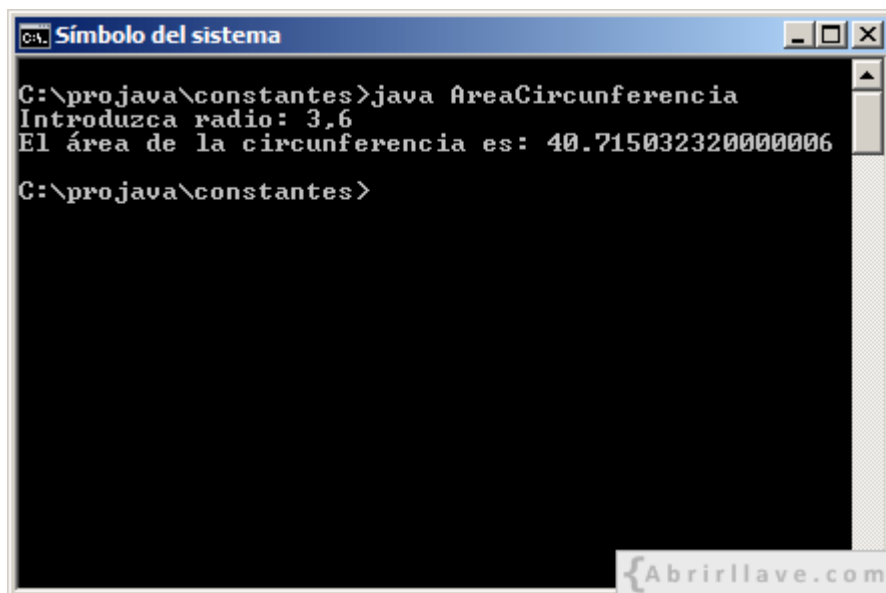
public class AreaCircunferencia
{
    public static void main(String[] args)
    {
        final double PI = 3.141592; // Declaración de una constante
        double area, radio;

        Scanner teclado = new Scanner(System.in);
        System.out.print("Introduzca radio: ");

        radio = teclado.nextDouble();
        area = PI * radio * radio;

        System.out.println("El área de la circunferencia es: " +
area);
    }
}
```

Este programa calcula el área de una circunferencia, cuyo radio es introducido por el usuario.



```
C:\projava\constantes>java AreaCircunferencia
Introduzca radio: 3,6
El área de la circunferencia es: 40.715032320000006
C:\projava\constantes>
```

Fuente: <https://www.abrirllave.com/java/constantes.php>

1.1 Operadores lógicos

Por: Kevin Alexander De León Andrade

¿Si quisiésemos evaluar más de una condición o utilizar operadores distintos en una misma sentencia? ¿Como hacemos esto? Para estos casos, java dispone de los llamados operadores lógicos.

Operadores Lógicos de Java.	
Operador.	Uso.
&&	Operador AND. Para que la condición resultante final sea true ambas condiciones tienen que ser ciertas.
	Operador OR. Para que la condición resultante final sea true, tan solo bastaría con que una fuese cierta.
!	Operador NOT. Dicho muy genéricamente, este operador se puede utilizar para cambiar el resultado de una condición. Es decir si la condición se evaluase como true mediante este operador lo transformaríamos en false y viceversa:.

Imagen: Ejemplo de operadores lógicos

Fuente: <https://codesitio.com/recursos-utiles-para-tu-web-o-blog/cursos/curso-de-java-operadores-de-relacion-y-logicos-de-java/>

1.1.1 Operadores Aritméticos

Los operadores aritméticos en Java son los operadores que nos permiten realizar operaciones matemáticas: suma, resta, multiplicación, división y resto.

Operador	Descripción
+	Operador de Suma. Concatena cadenas para la suma de String
-	Operador de Resta
*	Operador de Multiplicación
/	Operador de División
%	Operador de Resto

Imagen: Ejemplo de operadores aritméticos

Fuente: <https://www.manualweb.net/java/operadores-asignacion-aritmeticos-java/#:~:text=Los%20operadores%20aritm%C3%A9ticos%20en%20Java,%2C%20multiplicaci%C3%B3n%2C%20divisi%C3%B3n%20y%20resto.&text=Los%20operadores%20aritm%C3%A9ticos%20en%20Java%20los%20utiliza%20remos%20entre%20dos%20literales,variable%20o%20bien%20lo%20evaluamos.>

1.1.2 Operadores Relacionales

los operadores relacionales para verificar si se cumple una relación.

Operadores relacionales		
Operador	Utilización	Resultado
>	A > B	verdadero si A es mayor que B
>=	A >= B	verdadero si A es mayor o igual que B
<	A < B	verdadero si A es menor que B
<=	A <= B	verdadero si A es menor o igual que B
==	A == B	verdadero si A es igual a B
!=	A != B	verdadero si A es distinto de B

Imagen: Ejemplo de operadores relacionales

Fuente: https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_Java/Operadores_relacionales

1.1.3 Expresiones

Una expresión es un conjunto de variables, operadores e invocaciones de métodos que se construyen para poder ser evaluadas retornando un resultado.

Ejemplos de expresiones son:

```
int valor = 1;
if (valor 1 > valor2) { ... }
```

Imagen: Ejemplo de expresiones

Fuente: <https://www.manualweb.net/java/expresiones-sentencias-bloques-java/>

Reglas de prioridad

Por: Marta Alicia Del Carmen Diaz Cifuentes

Todos los operadores de comparación tienen la misma prioridad; es decir, se evalúan en el orden en que aparecen, de izquierda a derecha. Cuando la multiplicación y la división aparecen juntas en una expresión, cada operación se evalúa como aparece de izquierda a derecha.

Entrada de datos en consola

Para introducir un valor con Console, usamos `Console.ReadLine()`. Veremos que en la pantalla el puntero se queda parpadeando, a la espera de que introduzcamos datos. Al pulsar Enter, seguirá a la siguiente línea. Para guardar el valor, usamos una variable.

Por defecto es un String, pero podemos almacenar el valor en otro tipo de dato, siempre y cuando lo que introduzcamos sea correcto. Por ejemplo:

```
1 | Dim valor as Integer = Console.ReadLine()
```

Fuente: <https://www.discoduroderoer.es/introduccion-y-salida-de-datos-desde-consola-en-visual-basic-net/>

Si el valor que introducimos no se puede convertir en número, nos saltará un error. Para mostrar un valor, hay que usar `Console.Write(valor_a_mostrar)`. Donde el `valor_a_mostrar` será lo que queremos mostrar. Por ejemplo:

```
1 | Console.Write("Hola mundo")
```

Fuente: <https://www.discoduroderoer.es/introduccion-y-salida-de-datos-desde-consola-en-visual-basic-net/>

Si queremos que dé un salto de línea automáticamente, usamos `Console.WriteLine(valor_a_mostrar)`. Si queremos concatenar algún valor con el texto mostrado, usamos el operador `&` Por ejemplo:

```
1 | Dim nombre as String = "Fernando"
2 |
3 | Console.WriteLine("Hola " & nombre)
```



Hola mundo

Fuente: <https://www.discoduroderoer.es/introduccion-y-salida-de-datos-desde-consola-en-visual-basic-net/>

Clases buffered

Buffer solo afecta a matrices de tipos primitivos; esta clase no se aplica a los objetos. Cada tipo primitivo se trata como una serie de bytes sin tener en cuenta ningún comportamiento o limitación asociado al tipo primitivo.

Buffer proporciona métodos para copiar bytes de una matriz de tipos primitivos a otra matriz de tipos primitivos, obtener un byte de una matriz, establecer un byte en una matriz

y obtener la longitud de una matriz. Esta clase proporciona un mejor rendimiento para manipular tipos primitivos que métodos similares en la System.Array clase .

Bufferes aplicable a los siguientes tipos primitivos: Boolean, Char, Byte, UInt16, Int32, UInt32, Int16, SByte, Int64, IntPtr, UIntPtr, UInt64, Single, Double

Métodos

<code>BlockCopy(Array, Int32, Array, Int32, Int32)</code>	Copia un número especificado de bytes de una matriz de origen a partir de un desplazamiento determinado a una matriz de destino a partir de un desplazamiento determinado.
<code>ByteLength(Array)</code>	Devuelve el número de bytes de la matriz especificada.
<code>GetByte(Array, Int32)</code>	Recupera el byte en la posición especificada de la matriz determinada.
<code>MemoryCopy(Void*, Void*, Int64, Int64)</code>	Copia un número de bytes especificado como un valor de entero long desde una dirección de la memoria en otra. Esta API no es conforme a CLS.
<code>MemoryCopy(Void*, Void*, UInt64, UInt64)</code>	Copia un número de bytes especificado como un valor de entero long sin signo desde una dirección de la memoria a otra. Esta API no es conforme a CLS.
<code>SetByte(Array, Int32, Byte)</code>	Asigna un valor especificado a un byte en una posición determinada de una matriz especificada.

Fuente: <https://learn.microsoft.com/es-es/dotnet/api/system.buffer?view=net-7.04>

Se aplica a

Producto	Versiones
.NET	Core 1.0, Core 1.1, Core 2.0, Core 2.1, Core 2.2, Core 3.0, Core 3.1, 5, 6, 7, 8
.NET Framework	1.1, 2.0, 3.0, 3.5, 4.0, 4.5, 4.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2, 4.7, 4.7.1, 4.7.2, 4.8, 4.8.1
.NET Standard	1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 2.0, 2.1
UWP	10.0
Xamarin.iOS	10.8
Xamarin.Mac	3.0

Fuente: <https://learn.microsoft.com/es-es/dotnet/api/system.buffer?view=net-7.04>

Clase scanner



También debe implementar un analizador que implemente IScanner. Sin embargo, dado que este escáner funciona en línea a través de la Colorizer clase , normalmente es más fácil de implementar. Al principio de cada línea, MPF proporciona a la Colorizer clase un valor que se usará como una variable de estado que se pasa al analizador. Al final de cada línea, el analizador devuelve la variable de estado actualizada. MPF almacena en caché esta información de estado para cada línea para que el analizador pueda empezar a analizar desde cualquier línea sin tener que iniciarse al principio del archivo de origen. Este análisis rápido de una sola línea permite al editor proporcionar comentarios rápidos al usuario.

Sentencias condiciones

Las estructuras condicionales nos permiten ejecutar una serie de instrucciones si cumple una determinada condición que nosotros le indiquemos. Es importante recordar que la condición debe dar un resultado booleano, por lo que lo más normal es usar operadores relacionales y condicionales. Tenemos varios tipos de estructuras condicionales:

if: le indicamos una condición si esta es verdadera se ejecuta, sino no se ejecuta las instrucciones de dentro. Se suele traducir como «Si se cumple esta condición haz esto». Su sintaxis es:

```
1  if condicion then
2
3      'Instrucciones
4
5  end if
```

Por ejemplo:

```
1  Module Module1
2
3      Sub Main()
4
5          Dim precio As Integer = 300
6
7          If precio > 100 Then
8              Console.WriteLine("El precio es mayor que 100")
9          End If
10
11         Console.ReadLine ()
12
13     End Sub
14
15 End Module
```

Fuente: <https://www.discoduroderoer.es/estructuras-condicionales-en-visual-basic-net/>

Donde colocamos la condición, podemos colocar una variable booleana, ya que su valor puede ser true o false.

if – else: es como el anterior solo que después de cerrarse la llave de if, se añade else sin indicar ninguna condición. Esto sirve para que si la condición del if no es verdadera, ejecuta otras instrucciones que estarán dentro de else. Se suele traducir como «Si se cumple esta condición haz esto y sino haz esto».

Veamos un ejemplo:

```

1  Module Module1
2
3      Sub Main()
4
5          Dim precio As Integer = 50
6
7          If precio > 100 Then
8              Console.WriteLine("El precio es mayor que 100")
9          Else
10             Console.WriteLine("El precio es menor que 100")
11         End If
12     End Sub
13
14 End Module

```

En este caso, se ejecutaría lo que hay en `else` ya que no cumple la condición.

Fuente: <https://www.discoduroderoer.es/estructuras-condicionales-en-visual-basic-net/>

if -elseif: esta estructura es como una mezcla de los anteriores, esto nos permite, que si no se cumple la condición podamos indicar otra condición para hacerlo aún más específico. Se suele traducir como «Si se cumple esta condición haz esto y sino si cumple esta condición haz esto».

```

1  Module Module1
2
3      Sub Main()
4
5          Dim precio As Integer = 50
6
7          If precio > 100 Then
8              Console.WriteLine("El precio es mayor que 100")
9          ElseIf precio > 80 Then
10             Console.WriteLine("El precio es mayor que 80")
11         Else
12             Console.WriteLine("El precio es menor que 80")
13         End If
14     End Sub
15
16 End Module

```

El ejemplo anterior es equivalente al siguiente, esto es lo que se conoce como `if anidados`, podemos anidar tantos como queramos:

Fuente: <https://www.discoduroderoer.es/estructuras-condicionales-en-visual-basic-net/>

```

1  Module Module1
2
3      Sub Main()
4
5          Dim precio As Integer = 50
6
7          If precio > 100 Then
8              Console.WriteLine("El precio es mayor que 100")
9          Else
10             If precio > 80 Then
11                 Console.WriteLine("El precio es mayor que 80")
12             Else
13                 Console.WriteLine("El precio es menor que 80")
14             End If
15         End If
16     End Sub
17
18 End Module

```

Fuente: <https://www.discoduroderoer.es/estructuras-condicionales-en-visual-basic-net/>

Select: esta estructura condicional de selección múltiple, le damos un valor (puede ser una variable) y una lista de casos y si cumple alguno de los casos ejecuta las instrucciones asociadas a ella, en caso de que ninguna sea podemos indicarle que hace por defecto, normalmente se usa para indicar de un error con un mensaje. Su sintaxis es:

```

1  Select valor
2      Case caso1
3          'Instrucciones
4      Case caso2
5          'Instrucciones
6      Case casoN
7          'Instrucciones
8      Case Else
9          'Instrucciones
10 End Select

```

Los case de un select se pueden escribir con rangos con **To**, por ejemplo, **1 TO 5**, también podemos usar comas para indicar varios valores. Por ejemplo:

```

1  Module Module1
2
3      Sub Main()
4
5          Dim valor As Integer = 3
6
7          Select Case valor
8              Case 1 To 5
9                  Console.WriteLine("El valor esta entre 1 y 5")
10                 Console.WriteLine("El valor esta es 6 o 8 o 9")
11             Case Else
12                 Console.WriteLine("El valor es otro")
13             End Select
14
15             Console.ReadLine()
16
17         End Sub
18
19     End Module

```

Fuente: <https://www.discoduroderoer.es/estructuras-condicionales-en-visual-basic-net/>

1.1 Simple y Doble

Por: Oscar Echeverría

Las condiciones se utilizan para controlar el flujo de ejecución de un programa en función de si una determinada condición se cumple o no, hay dos tipos de condiciones, las simples y las dobles.

Las condiciones **simples** se utilizan para ejecutar una acción si una condición es verdadera.

Ejemplo

```
if (condición) {
    // código a ejecutar si la condición es verdadera}
```

Las condiciones **dobles** se utilizan para ejecutar una acción si una condición es verdadera y otra acción si la condición es falsa.

Ejemplo

```
if (condición) {
    // código a ejecutar si la condición es verdadera
} else {
    // código a ejecutar si la condición es falsa}
```

1.1.1 Múltiples o anidadas

Las condiciones múltiples o anidadas se utilizan cuando es necesario evaluar varias condiciones a la vez y ejecutar diferentes acciones en función de ellas.

Ejemplo

```
if (condición1) {
    // código a ejecutar si la condición1 es verdadera
} else if (condición2) {
    // código a ejecutar si la condición2 es verdadera
} else if (condición3) {
    // código a ejecutar si la condición3 es verdadera
} else {
    // código a ejecutar si ninguna de las condiciones es verdadera
}
```

1.1.2 Estructuras de repetición

Las estructuras de repetición se utilizan para ejecutar un bloque de código varias veces, hasta que se cumple una determinada condición, Hay tres tipos de estructuras de repetición tenemos: **el bucle for**, **el bucle while** y **el bucle do-while**.

Bucle for: se utiliza cuando se conoce el número exacto de veces que se debe ejecutar el bloque de código.

Ejemplo

```
for (inicialización; condición; expresión de actualización) {
    // código a ejecutar
}
```

Bucle while: se utiliza cuando se desconoce el número exacto de veces que se debe ejecutar el bloque de código, pero se conoce la condición de finalización.

Ejemplo

```
while (condición) {
    // código a ejecutar
}
```

Bucle do-while: se utiliza cuando se desea ejecutar el bloque de código al menos una vez, independientemente de la condición de finalización.

Ejemplo

```
do {
    // código a ejecutar
} while (condición);
```

1.1.3 While

La estructura while es una herramienta poderosa para controlar el flujo de datos y se utiliza comúnmente para leer datos de un archivo, procesar datos de entrada del usuario y para cualquier otra tarea que requiera que un bloque de código se repita mientras se cumpla una condición determinada.

Ejemplo

```
while (condición) {
    // código a ejecutar
}
```

While se utiliza para repetir un bloque de código mientras se cumple una condición determinada.

Switch (palabra) {

```
    Case `casa`:
    Breack;
    Case `hogar`:
    Breack;

    Default:
```

```
}
```

1.1 Visual Basic

1.2 Do While y For repetitivas

Por: Aroldo Escobar

En Visual Basic, las estructuras de control de flujo Do While y For se utilizan para repetir bloques de código varias veces.

La estructura Do While ejecuta un bloque de código mientras se cumple una determinada condición. El formato de la estructura Do While es el siguiente:

Do While condición

'código a ejecutar mientras se cumpla la condición

Loop

Por ejemplo, el siguiente código utiliza una estructura Do While para imprimir los números del 1 al 10:

Dim i As Integer

i = 1

Do While i <= 10

 Console.WriteLine(i)

 i = i + 1

Loop

La estructura For, por otro lado, se utiliza para ejecutar un bloque de código un número determinado de veces. El formato de la estructura For es el siguiente:

For variable = valorInicial To valorFinal Step incremento

'código a ejecutar en cada iteración

Next

Por ejemplo, el siguiente código utiliza una estructura For para imprimir los números del 1 al 10:

For i = 1 To 10

 Console.WriteLine(i)

Next

En este caso, la variable i se inicializa en 1, se ejecuta el bloque de código para cada valor de i desde 1 hasta 10, y se incrementa i en 1 en cada iteración.

1.3 Anidadas

En Visual Basic, se pueden anidar estructuras de control de flujo dentro de otras estructuras de control de flujo. Esto se utiliza cuando se necesita realizar una repetición dentro de otra repetición o una condición dentro de otra condición.

Por ejemplo, el siguiente código utiliza una estructura Do While anidada dentro de una estructura For para imprimir los números del 1 al 5 para cada letra del alfabeto:

```
For letra = Asc("A") To Asc("E")
    Dim i As Integer
    i = 1
    Do While i <= 5
        Console.WriteLine(Chr(letra) & i)
        i = i + 1
    Loop
Next
```

En este caso, la estructura For itera a través de las letras A a E utilizando la función Asc() para obtener el código ASCII de la letra y la función Chr() para obtener la letra correspondiente. Dentro de la estructura For, se utiliza una estructura Do While para imprimir los números del 1 al 5 para cada letra del alfabeto.

También se pueden anidar estructuras If dentro de otras estructuras If para realizar una comprobación más compleja. Por ejemplo:

```
If condición1 Then
    'código a ejecutar si se cumple la condición1
    If condición2 Then
        'código a ejecutar si se cumple la condición2
    Else
        'código a ejecutar si no se cumple la condición2
    End If
Else
    'código a ejecutar si no se cumple la condición1
End If
```

En este caso, se evalúa la condición1 y, si se cumple, se evalúa la condición2 utilizando una estructura If anidada. Si la condición2 se cumple, se ejecuta el código dentro del primer bloque If, de lo contrario se ejecuta el código dentro del bloque Else de la estructura If anidada. Si la condición1 no se cumple, se ejecuta el código dentro del bloque Else de la estructura If principal.

1.4 Vectores y matrices

En Visual Basic, un vector es una estructura de datos que almacena una colección de valores del mismo tipo. Para declarar un vector, se utiliza la siguiente sintaxis:

```
Dim nombreVector(tamaño) As tipoDeDato
```

Donde "nombreVector" es el nombre que se le da al vector, "tamaño" es el número de elementos que se desean almacenar en el vector y "tipoDeDato" es el tipo de dato que se desea almacenar. Por ejemplo:

```
Dim miVector(4) As Integer
```

Esto declara un vector llamado "miVector" con 5 elementos de tipo Integer (enteros) numerados desde 0 hasta 4. Para asignar valores a un vector, se utiliza el siguiente formato:

```
nombreVector(indice) = valor
```

Donde "indice" es la posición del elemento en el vector y "valor" es el valor que se desea asignar a ese elemento. Por ejemplo:

```
miVector(0) = 10
```

```
miVector(1) = 20
```

```
miVector(2) = 30
```

```
miVector(3) = 40
```

```
miVector(4) = 50
```

Esto asigna los valores 10, 20, 30, 40 y 50 a los elementos del vector "miVector".

Una matriz es una estructura de datos que almacena una colección de valores en una o más dimensiones. Para declarar una matriz, se utiliza la siguiente sintaxis:

```
Dim nombreMatriz(filas, columnas) As tipoDeDato
```

Donde "nombreMatriz" es el nombre que se le da a la matriz, "filas" es el número de filas que se desean almacenar en la matriz y "columnas" es el número de columnas que se desean almacenar en la matriz y "tipoDeDato" es el tipo de dato que se desea almacenar. Por ejemplo:

```
Dim miMatriz(2, 2) As Integer
```

Esto declara una matriz llamada "miMatriz" con 3 filas y 3 columnas de tipo Integer (enteros). Para asignar valores a una matriz, se utiliza el siguiente formato:

```
nombreMatriz(fila, columna) = valor
```

Donde "fila" es el número de la fila y "columna" es el número de la columna y "valor" es el valor que se desea asignar a esa posición en la matriz. Por ejemplo:

```
miMatriz(0, 0) = 10
```

```
miMatriz(0, 1) = 20
```

```
miMatriz(0, 2) = 30
```

```
miMatriz(1, 0) = 40
```

```
miMatriz(1, 1) = 50
```

```
miMatriz(1, 2) = 60
```

```
miMatriz(2, 0) = 70
```

```
miMatriz(2, 1) = 80
```

`miMatriz(2, 2) = 90`

Esto asigna los valores a las posiciones correspondientes de la matriz "miMatriz".

1.5 Arreglos

En Visual Basic, un arreglo es una colección de elementos del mismo tipo que se almacenan en memoria contigua. Los arreglos son útiles para almacenar y manipular datos de manera eficiente. Hay varios tipos de arreglos disponibles en Visual Basic, entre los que se incluyen:

1.- Arreglos unidimensionales: son arreglos que tienen una sola dimensión. Se declaran utilizando la siguiente sintaxis:

Dim `nombreArreglo`(tamaño) As tipoDeDato

Donde "nombreArreglo" es el nombre que se le da al arreglo, "tamaño" es el número de elementos que se desean almacenar en el arreglo y "tipoDeDato" es el tipo de dato que se desea almacenar.

2.- Arreglos multidimensionales: son arreglos que tienen dos o más dimensiones. Se declaran utilizando la siguiente sintaxis:

Dim `nombreArreglo`(filas, columnas) As tipoDeDato

Donde "nombreArreglo" es el nombre que se le da al arreglo, "filas" es el número de filas que se desean almacenar en el arreglo y "columnas" es el número de columnas que se desean almacenar en el arreglo y "tipoDeDato" es el tipo de dato que se desea almacenar.

3.- Arreglos dinámicos: son arreglos cuyo tamaño se puede cambiar en tiempo de ejecución. Se declaran utilizando la siguiente sintaxis:

Dim `nombreArreglo`() As tipoDeDato

Donde "nombreArreglo" es el nombre que se le da al arreglo y "tipoDeDato" es el tipo de dato que se desea almacenar. El tamaño del arreglo se puede cambiar utilizando la función ReDim, por ejemplo:

ReDim `nombreArreglo`(tamaño)

Donde "tamaño" es el nuevo tamaño del arreglo.

Para acceder a los elementos de un arreglo, se utiliza la sintaxis:

`nombreArreglo`(indice)

Donde "indice" es el índice del elemento que se desea acceder. Para asignar un valor a un elemento del arreglo, se utiliza la sintaxis:

`nombreArreglo`(indice) = valor

Donde "valor" es el valor que se desea asignar al elemento del arreglo.

1.6 Unidimensionales

En Visual Basic, los arreglos unidimensionales son arreglos que tienen una sola dimensión. Se utilizan para almacenar y manipular una lista de valores del mismo tipo.

Para declarar un arreglo unidimensional, se utiliza la siguiente sintaxis:

```
Dim nombreArreglo(tamaño) As tipoDeDato
```

Donde "nombreArreglo" es el nombre que se le da al arreglo, "tamaño" es el número de elementos que se desean almacenar en el arreglo y "tipoDeDato" es el tipo de dato que se desea almacenar. Por ejemplo, para declarar un arreglo unidimensional de enteros con 10 elementos, se utiliza la siguiente línea de código:

```
Dim numeros(9) As Integer
```

Los elementos del arreglo se pueden acceder utilizando el índice del elemento correspondiente. El índice de un arreglo unidimensional comienza en cero y termina en el número de elementos menos uno. Por ejemplo, para acceder al primer elemento del arreglo "numeros", se utiliza el índice cero de la siguiente manera:

```
numeros(0) = 5
```

Para recorrer todos los elementos del arreglo, se puede utilizar un ciclo For o un ciclo For Each. Por ejemplo, para imprimir todos los elementos del arreglo "numeros", se puede utilizar el siguiente código:

```
For i = 0 To 9
```

```
    Console.WriteLine(numeros(i))
```

```
Next
```

También es posible utilizar la función Split para convertir una cadena de texto en un arreglo de cadenas. Por ejemplo, para dividir una cadena en palabras utilizando el espacio como separador, se puede utilizar el siguiente código:

```
Dim texto As String = "Hola mundo"
```

```
Dim palabras() As String = texto.Split(" ")
```

```
For Each palabra In palabras
```

```
    Console.WriteLine(palabra)
```

```
Next
```

Este código imprimirá las palabras "Hola" y "mundo".

1.1 Bidimensionales.

Por: Thania Paola García Cos

Los arreglos bidimensionales o también conocidos como matrices en JAVA, son un conjunto de datos del mismo tipo. En este conjunto los datos se organizan en filas y columnas. Se dice que son bidimensionales porque para acceder a una posición de una matriz se requieren de dos índices, el índice de la fila y el índice de la columna. Para crear una matriz en JAVA se utiliza el tipo y corchetes cuadrados dobles.

1.1.1 Métodos.

Un método es un bloque de código que contiene una serie de instrucciones. Un programa hace que se ejecuten las instrucciones al llamar al método y especificando los argumentos de método necesarios. En C#, todas las instrucciones ejecutadas se realizan en el contexto de un método. El método Main es el punto de entrada para cada aplicación de C# y se llama mediante Common Language Runtime (CLR) cuando se inicia el programa. En una aplicación que usa instrucciones de nivel superior, el compilador genera el método Main y contiene todas las instrucciones de nivel superior.

1.1.2 Procedimientos.

Uno de los métodos fundamentales para resolver un problema es dividirlo en problemas más pequeños, llamados subproblemas. Estos problemas pueden a su vez dividirse repetidamente en problemas más pequeños hasta que los problemas sean de fácil solución. Divide y vencerás ... Cada subproblema es deseable que sea independiente de los demás y se denomina módulo. El problema original se resuelve con un programa principal (llamado también driver o main), y los subproblemas (módulos) mediante subprogramas: procedimientos y funciones.

1.1.3 Funciones.

Una función es un bloque de código que realiza alguna operación. Una función puede definir opcionalmente parámetros de entrada que permiten a los llamadores pasar argumentos a la función. Una función también puede devolver un valor como salida. Las funciones son útiles para encapsular las operaciones comunes en un solo bloque reusable, idealmente con un nombre que describa claramente lo que hace la función. La siguiente función acepta dos enteros de un autor de llamada y devuelve su suma; a y b son parámetros de tipoint.

```
C++

int sum(int a, int b)
{
    return a + b;
}
```

Imagen: Funciones

Fuente: <https://learn.microsoft.com/es-es/cpp/cpp/functions-cpp?view=msvc-170>

1.1.4 Elementos de una declaración de función

Una declaración de función mínima consta del tipo de valor devuelto, el nombre de la función y la lista de parámetros (que pueden estar vacíos), junto con palabras clave opcionales que proporcionan más instrucciones al compilador. El siguiente ejemplo es una declaración de función:

```
int sum(int a, int b);
```

Imagen: Elementos de una declaración de función

Fuente: <https://learn.microsoft.com/es-es/cpp/cpp/functions-cpp?view=msvc-170>

1.1.5 JAVA POO SWING

La Java FoundationClasses (JFC) se compone del AWT, Swing y Java 2D. Juntas, proporcionan una interfaz de usuario consistente para programas Java, tanto si el sistema de interfaz de usuario es Windows, Mac Oso Linux. También suministra gestores de disposición básicos, un paquete de transferencia de datos para uso con el bloc denotas, arrastrar y soltar y la interfaz para los dispositivos de entrada comunes como el ratón y el teclado. Java Swinges un paquete que contienen conjunto de librerías que provee herramientas o facilidades para la construcción degusto Interfaces Graficas de Usuario (GraphicalUser Interface).

1.1 PROGRAMACIÓN ORIENTADA A OBJETOS.

Por: Daniel de Jesús Gerónimo Vaquiáx

La programación orientada a objetos (POO) es un paradigma de programación que se basa en la creación de objetos que interactúan entre sí para realizar una tarea.

1.1.1 Encapsulamiento

Se refiere a la capacidad de ocultar los detalles de implementación de un objeto, lo que significa que solo los métodos públicos de la clase pueden ser utilizados por otros objetos.

1.1.2 Herencia

Permite que una clase herede atributos y métodos de otra clase, lo que permite la reutilización del código y la organización jerárquica de los objetos.

1.1.3 Polimorfismo

Se refiere a la capacidad de objetos de diferentes clases para responder de manera diferente a los mismos mensajes. Esto permite la creación de interfaces comunes para objetos de diferentes clases.



Imagen: PROGRAMACIÓN ORIENTADA A OBJETOS

Fuente: <https://ed.team/cursos/poo>

2.1 ESTRUCTURA DE UNA CLASE.

Por: Daniel de Jesús Gerónimo Vaquiax

La estructura de una clase en programación orientada a objetos (POO) depende del lenguaje de programación que se esté utilizando, pero en términos generales, una clase consta de los siguientes elementos:

2.1.1 Nombre de la clase:

Es el nombre que se le da a la clase y debe ser representativo del objeto que se está modelando

2.1.2 Atributos

Son las características o propiedades que definen el objeto de la clase. Los atributos pueden ser públicos (accesibles desde cualquier parte del código), privados (accesibles solo desde dentro de la clase) o protegidos (accesibles desde la clase y sus subclases).

2.1.3 Métodos

Son las acciones o comportamientos que puede realizar el objeto de la clase. Los métodos pueden ser públicos, privados o protegidos y se definen dentro de la clase.



Imagen: Estructura de la clase

Fuente: <https://pt.slideshare.net/johnnikhriz/estructura-de-la-clase-28748361>

2.1.4 Constructores

Son métodos especiales que se utilizan para inicializar los objetos de la clase. Un constructor tiene el mismo nombre que la clase y se ejecuta automáticamente cuando se crea un objeto de la clase.

2.1.5 Destructor:

Es un método especial que se utiliza para liberar los recursos utilizados por un objeto de la clase cuando ya no se necesitan. El destructor tiene el mismo nombre que la clase, pero con el prefijo "~".

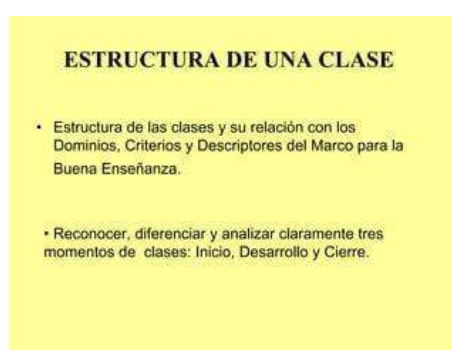


Imagen: Estructura de una clase

Fuente: <https://es.slideshare.net/esmico/estructura-de-una-clase-61671813>

3,1 ATRIBUTOS Y ÁMBITOS DE VARIABLES.

Por: Daniel de Jesús Gerónimo Vaquiáx

En programación, un atributo es una propiedad o característica que tiene un objeto y que se define en la clase a la que pertenece.

3.3.1 Atributos públicos:

Son aquellos atributos que solo se pueden acceder y modificar desde dentro de la misma clase. Se definen con la palabra clave "private" en la declaración de la clase.

3.3.2 Atributos privados:

Son las características o propiedades que definen el objeto de la clase. Los atributos pueden ser públicos (accesibles desde cualquier parte del código), privados (accesibles solo desde dentro de la clase) o protegidos (accesibles desde la clase y sus subclases).

3.3.3 Atributos protegidos:

Son aquellos atributos que solo se pueden acceder y modificar desde dentro de la misma clase y de sus subclases. Se definen con la palabra clave "protected" en la declaración de la clase.

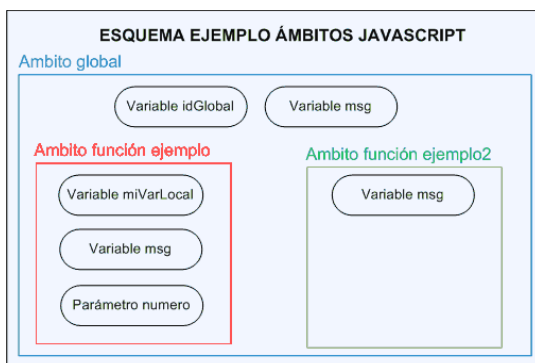


Imagen: Atributos

Fuente: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=852:typeof-javascript-saber-el-tipo-de-variable-global-y-local-ambito-scope-var-ejemplo-cuenta-atras-cu01167&catid=78&Itemid=206

4.1 MÉTODOS ACCESORES (GET Y SET).

Por: Daniel de Jesús Gerónimo Vaquiáx

Los métodos "getters" se utilizan para obtener el valor de un atributo de la clase, mientras que los métodos "setters" se utilizan para establecer o modificar el valor de un atributo de la clase. Estos métodos son una forma de encapsulamiento que protege los atributos de la clase y permite una manipulación más segura de los mismos.

Los métodos "getters" se definen con el prefijo "get" seguido del nombre del atributo y no tienen parámetros de entrada. Estos métodos devuelven el valor del atributo solicitado y pueden ser públicos, privados o protegidos.

Por ejemplo, si tenemos una clase "Persona" con un atributo "nombre", el método "getter" para obtener el valor de ese atributo podría ser:

```
public String getNombre() {
    return nombre;
}
```

Los métodos "setters" se definen con el prefijo "set" seguido del nombre del atributo y tienen un parámetro de entrada que corresponde al nuevo valor que se quiere establecer en el atributo. Estos métodos no devuelven ningún valor y pueden ser públicos, privados o protegidos.

```

1  import java.util.Scanner;
2  public class Inicio{
3      public static void main (String[]args){
4          Scanner ingresoSt = new Scanner(System.in);
5          Scanner ingresoEd = new Scanner(System.in);
6          String nombre;
7          int edad;
8
9      } //fin main
10 } //fin class

```

Imagen: Métodos Get y Set

Fuente: <http://codigogx.blogspot.com/2016/08/capitulo-61-java-metodos-get-y-set.html>

5.1 COLABORACIÓN DE CLASES.

Por: Daniel de Jesús Gerónimo Vaquias

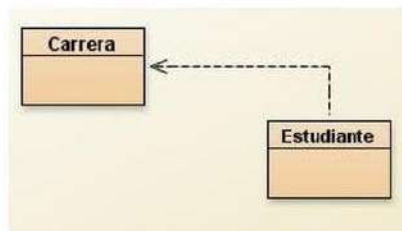
La colaboración de clases es una técnica de programación orientada a objetos que consiste en definir y utilizar clases de forma interdependiente para construir un sistema más complejo. En otras palabras, las clases se comunican entre sí para realizar una tarea específica.

En la colaboración de clases, una clase puede utilizar los servicios de otra clase a través de la creación de objetos y la invocación de métodos. Por lo tanto, es común que una clase tenga atributos que sean objetos de otras clases.

Por ejemplo, si tenemos una clase "Coche" y otra clase "Motor", la clase "Coche" podría tener un atributo que sea un objeto de la clase "Motor" para poder realizar acciones como encender el motor o cambiar de marcha.

Además, la colaboración de clases también implica la definición de interfaces claras y coherentes entre las clases, lo que permite una mayor modularidad y flexibilidad en el diseño del sistema.

POO-2 Colaboración Entre Clases



J. Mancilla M.

POO - COLABORACION

Imagen: Poo 2 colab, presentación que muestra implementa colaboración entre clases

Fuente: <https://es.slideshare.net/jlmanmons/poo-2-colab>

1.1 Herencia

Por: Patrick Leonardo

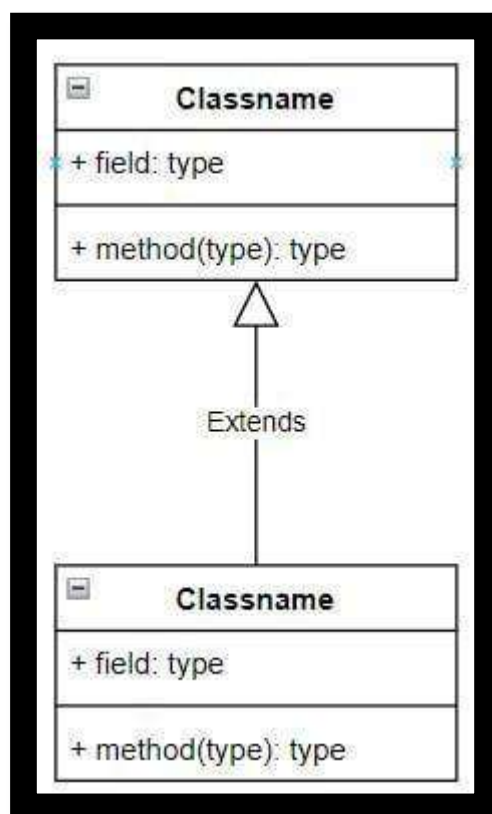
La herencia en Java implica que una superclase o clase base hereda sus funciones y atributos a una subclase o clase derivada. La palabra reservada que nos permite realizar herencia entre clases es extends.

Cabe mencionar que en Java los constructores de las clases no se heredan; sin embargo, pueden ejecutarse en la clase hija con el método super(). Considera que las funciones o atributos del tipo private no podrán heredarse.

1.1.1 4 tipos de herencia en JAVA

1.2 Herencia simple o única

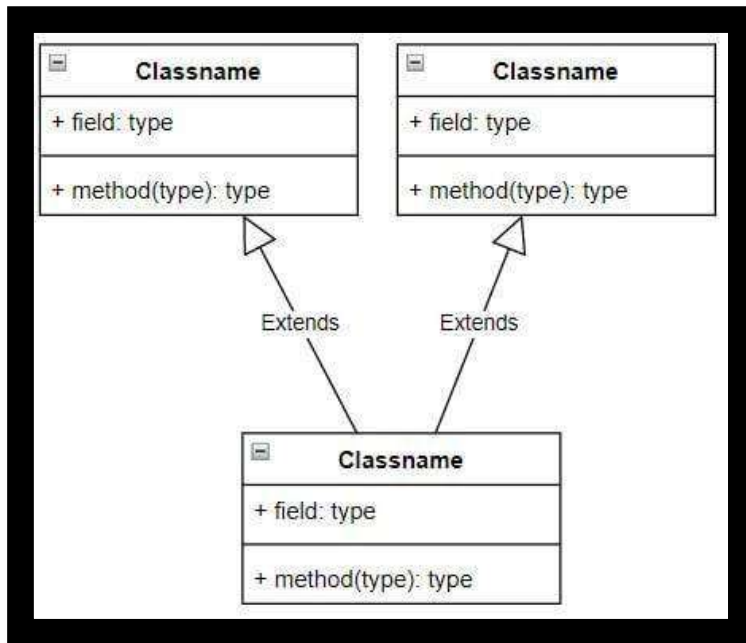
Una clase hija hereda de una clase padre.



Fuente: <https://blog.hubspot.es/website/que-es-herencia-java#:~:text=%C2%BFQu%C3%A9%20es%20la%20herencia%20en,herencia%20entre%20clases%20es%20extends.>

1.2.1 Herencia múltiple

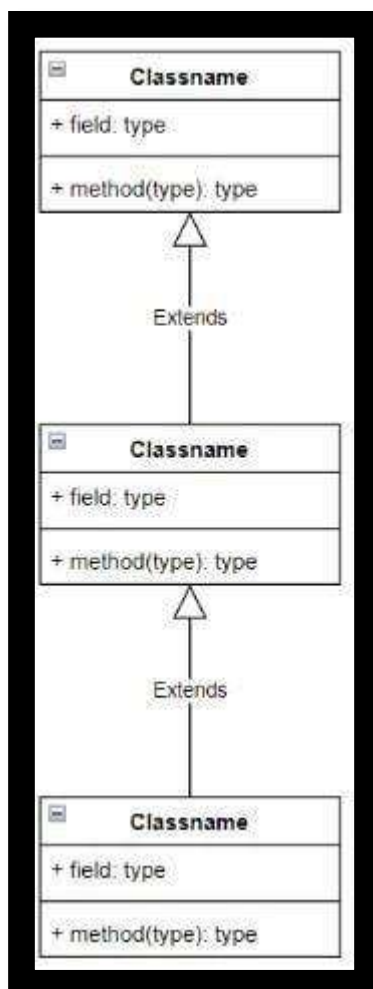
Una clase hija hereda de dos o más clases padre.



Fuente: <https://blog.hubspot.es/website/que-es-herencia-java#:~:text=%C2%BFQu%C3%A9%20es%20la%20herencia%20en,herencia%20entre%20clases%20es%20extends.>

1.2.2 Herencia multinivel

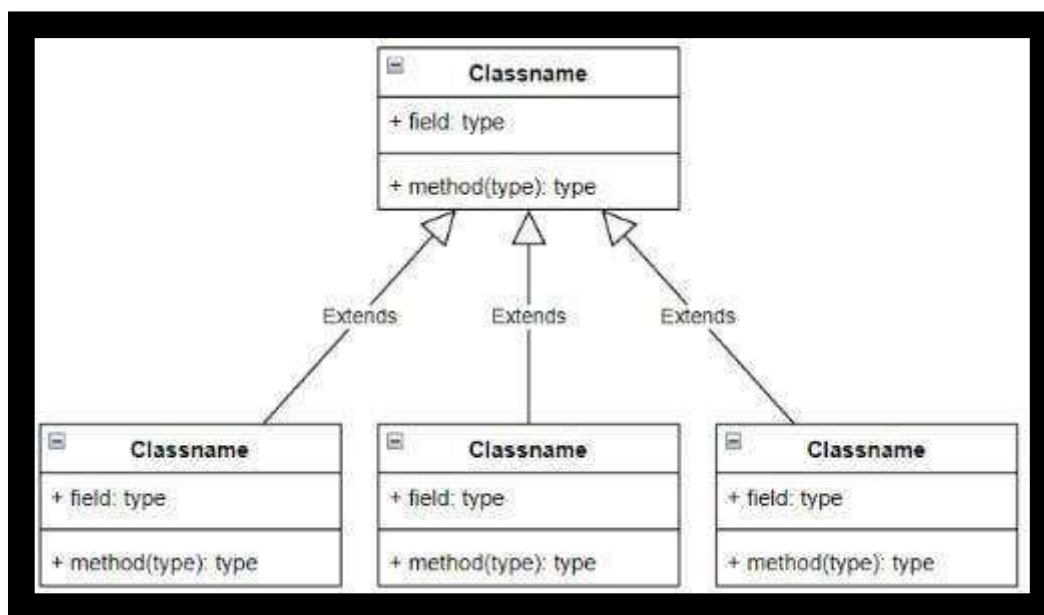
Una clase hija hereda de una clase padre, que a su vez es clase hija de otra clase padre.



Fuente: <https://blog.hubspot.es/website/que-es-herencia-java#:~:text=%C2%BFQu%C3%A9%20es%20la%20herencia%20en,herencia%20entre%20clases%20es%20extends.>

1.2.3 Herencia jerárquica

Una clase padre hereda a dos o más clases hijas.



Fuente: <https://blog.hubspot.es/website/que-es-herencia-java#:~:text=%C2%BFQu%C3%A9%20es%20la%20herencia%20en,herencia%20entre%20clases%20es%20extends.>

1.3 Herencia

1.3.1 Para qué sirve el polimorfismo en java?

A grandes rasgos, el polimorfismo permite que nombres dos acciones del mismo modo dentro de tu código, pero que cada una de ellas acepte diferentes parámetros. Esto facilita que identifiques objetos nominalmente, pero que cada uno de ellos desempeñe una función en particular.

Pensemos, por ejemplo, que quieres llamar `Multiply()` a dos funciones. Una de ellas será utilizada para calcular el producto de dos números enteros, mientras que la otra tendrá como función calcular el producto de dos decimales. Tu código deberá verse así:

```
public class Main {
    public static void main(String[ ] args) {
        // usando la primera función
        Multiplier.Multiply(3,5);
        // usando la segunda función
        Multiplier.Multiply(3.5,5.1);
    }
}
```

}

}

Cuando el código sea ejecutado, el programa deberá arrojar:

- Producto de los enteros: 15
- Producto de los decimales: 17,849999999999998

Pero ¿por qué no simplemente utilizar un nombre diferente? La respuesta radica en que en programación siempre debe promoverse un sistema abierto-cerrado para tener un código limpio y claro. Esto significa que un código debe estar abierto a la extensión, pero cerrado a la modificación. Si, por ejemplo, quieres ampliar las capacidades de un código, no deberás crear una nueva clase, sino extender sus capacidades agregando nuevas funcionalidades. Es siempre preferible hacerlo mediante herencia y polimorfismo.

1.3.1 Diferencias entre herencia y polimorfismo

Es un error común confundir herencia con polimorfismo pues ambos sirven para funciones similares. Sin embargo, es importante recalcar que son sustancialmente diferentes.

En la herencia, la clase que está anidada dentro de otra adquiere los métodos y atributos de la clase madre. O, lo que es igual, sus funciones. Mira este ejemplo:

```
class Forma {
// métodos de la clase Forma
}

class Cuadrado extends Forma {
// gracias a que la clase Cuadrado se encuentra anidada dentro de la clase Forma, esta
última es accesible en cualquier parte del código

// métodos de la clase Cuadrado
}
```

Cuando tenemos herencia dentro de un código, la clase hija hereda los mismos atributos y métodos de la clase en la que está anidada, sin modificar su funcionalidad. Por el contrario, en el caso del polimorfismo, la clase anidada recibe los mismos atributos y métodos que la clase madre, pero cuenta con su propia implementación para cada método.

Podríamos decir que la herencia es una forma de reutilizar código, mientras que el polimorfismo es una forma de dinamizarlo y simplificar el uso de diferentes funciones. Esto se logra porque el programa reconocerá qué función debe cumplir dependiendo de los parámetros que se ingresen en la clase.

1.3.1 Tipos de polimorfismo

1.3.1 Polimorfismo estático

Este tipo de polimorfismo, también conocido como de compilación, se utiliza para crear múltiples métodos con el mismo nombre en la misma clase, que contengan diferentes números de parámetros o bien parámetros de distintos tipos.

En el ejemplo que veremos a continuación, podemos advertir que existen 3 métodos con el mismo nombre, pero que aceptan diferentes parámetros:

```
package com.hubspot;

class Multiplier {
    static void Multiply(int a, int b)
    {
        System.out.println("Multiplicación de enteros, Resultado = "+ a*b);
    }

    // Método 1
    static void Multiply(double a, double b)
    {
        System.out.println("Multiplicación de decimales, Resultado = "+ a*b);
    }

    // Método 2
    static void Multiply(double a, double b, double c)
    {
        System.out.println("Tres parámetros, Multiplicación de decimales, Resultado = "+ a*b*c);
    }
}

public class Main {
    public static void main(String[] args) {
        // usando el primer método
        Multiplier.Multiply(3,5);
        // usando el segundo método
        Multiplier.Multiply(3.5,5.1);
        // usando el tercer método Multiplier.Multiply(3.6,5.2, 6.3);
    }
}
```

Al correr el código obtendremos como resultado:

Multiplicación de enteros, Resultado = 15

Multiplicación de decimales, Resultado = 17,849999999999998

Tres parámetros, Multiplicación de decimales, Resultado = 117.936

El primer método tomará 2 enteros; el segundo 2 decimales, y el tercero 3 valores con diferentes parámetros. ¿Notas que no hicimos ninguna modificación al ingresar la información? Los tres llamados son iguales, pero actúan de diferente manera al ingresar distintos tipos de parámetros.

Fuente: <https://blog.hubspot.es/website/polimorfismo-java#:~:text=El%20polimorfismo%20hace%20referencia%20a,nombre%20para%20ejecutar%20funciones%20diferentes.>

1.3.1 Polimorfismo dinámico

El polimorfismo dinámico o de ejecución es aquel en el que la clase hija tiene una definición propia, pero que depende de la clase en la que está anidada.

Veamos un ejemplo:

```
package com.hupspot;

// Super clase
class Figura{
protected double length;
Figura(double length){
this.length = length;
} void area(){
}
}

// clase hija
class Cuadrado extends Forma{
//constructor
Cuadrado(double side){
super(side); // llamado a la clase del constructor
}

//Overriding area() method
void area(){
System.out.println("Área del cuadrado = " + length*length);
}
}

// clase hija
class Círculo extends Figura{
```

```
//constructor
Círculo(double radius){
super(radius); // llamado a la clase del constructor
}
//Overriding area() method
void area(){
System.out.println("Área del Círculo " + 3.14*length*length);
}
}

public class Main {
public static void main(String[] args){
Figura shape = new Cuadrado(5.0);
// llamado al método de la clase Cuadrado shape.area();
shape = new Circle(5.0); // upcasting // llamado al método de la clase Círculo
shape.area();
}
}
```

El resultado de este código será:

- Área del Cuadrado = 25,0
- Área del Círculo = 75,0

Como puedes ver, el uso del mismo nombre para una clase nos ha permitido extender sus funciones sin modificar el llamado Figura.

Fuente: <https://blog.hubspot.es/website/polimorfismo-java#:~:text=El%20polimorfismo%20hace%20referencia%20a,nombre%20para%20ejecutar%20funciones%20diferentes.>

1.4 Sobrecarga de métodos

La sobrecarga de métodos es la creación de varios métodos con el mismo nombre pero con diferente lista de tipos de parámetros. Java utiliza el número y tipo de parámetros para seleccionar cuál definición de método ejecutar.

Java diferencia los métodos sobrecargados con base en el número y tipo de parámetros o argumentos que tiene el método y no por el tipo que devuelve.

También existe la sobrecarga de constructores: Cuando en una clase existen constructores múltiples, se dice que hay sobrecarga de constructores.

```

Ejemplo

/* Métodos sobrecargados */
int calculaSuma(int x, int y, int z){
    ...
}
int calculaSuma(double x, double y, double z){
    ...
}

/* Error: estos métodos no están sobrecargados */
int calculaSuma(int x, int y, int z){
    ...
}
double calculaSuma(int x, int y, int z){
    ...
}

```

Fuente: <http://profesores.fi->

[b.unam.mx/carlos/java/java_basico4_6.html#:~:text=La%20sobrecarga%20de%20m%C3%A9todos%20es,cu%C3%A1l%20definici%C3%B3n%20de%20m%C3%A9todo%20ejecutar.](http://profesores.fi-b.unam.mx/carlos/java/java_basico4_6.html#:~:text=La%20sobrecarga%20de%20m%C3%A9todos%20es,cu%C3%A1l%20definici%C3%B3n%20de%20m%C3%A9todo%20ejecutar.)

1.5 Arreglos de objetos

Un arreglo es una estructura de datos utilizada para almacenar datos del mismo tipo. Los arreglos almacenan sus elementos en ubicaciones de memoria contiguas. En Java, los arreglos son objetos. Todos los métodos se pueden ser invocados en un arreglo.

Como hemos visto en los arrays, debemos definir el tipo de dato que debe contener, pues también puede contener objetos de una clase. Con las posiciones, podemos hacer igual que si fuera un objeto, como invocar métodos.

```

1  public class EmpleadoApp {
2
3      public static void main(String[] args) {
4
5          //Creamos un array de objetos de la clase empleados
6          Empleado arrayObjetos[]=new Empleado[3];
7
8          //Creamos objetos en cada posición
9          arrayObjetos[0]=new Empleado("Fernando", "Ureña", 23, 1000);
10         arrayObjetos[1]=new Empleado("Epi", "Dermis", 30, 1500);
11         arrayObjetos[2]=new Empleado("Blas", "Femia", 25, 1200);
12
13         //Recorremos el array para calcular la suma de salarios
14
15         int suma=0;
16         for (int i=0;i<arrayObjetos.length;i++){
17             suma+=arrayObjetos[i].getSalario();
18         }
19         System.out.println("La suma de salarios es "+suma);
20     }
21 }
22 }

```

Fuente: <https://www.discoduroderoer.es/arrays-de-objetos-en-java/>

1.6 Swing introduction

El tutorial de Java Swing es una parte de Java Foundation Classes (JFC) que se utiliza para crear aplicaciones basadas en ventanas. Está construido sobre la API AWT (Abstract Windowing Toolkit) y está completamente escrito en java. A diferencia de AWT, Java Swing proporciona componentes livianos e independientes de la plataforma.

No.	AWT de Java	Columpio Java
1)	Los componentes AWT dependen de la plataforma .	Los componentes swing de Java son independientes de la plataforma .
2)	Los componentes AWT son pesados .	Los componentes del columpio son livianos .
3)	AWT no es compatible con la apariencia conectable .	Swing es compatible con la apariencia y la sensación conectables .
4)	AWT proporciona menos componentes que Swing.	Swing proporciona componentes más potentes , como tablas, listas, paneles de desplazamiento, selector de color, panel con pestañas, etc.
5)	AWT no sigue MVC (controlador de vista de modelo), donde el modelo representa datos, la vista representa la presentación y el controlador actúa como una interfaz entre el modelo y la vista.	Swing sigue a MVC .

Fuente: <https://www.javatpoint.com/java-swing>

1.1 Java Poo Swing

Por: Joshua Gonzalez

1.1.1 JButton

JButton es una clase en Java que representa un botón de interfaz de usuario. Es parte del paquete javax.swing y se utiliza para agregar interactividad en una aplicación de escritorio Java. Los botones se pueden hacer clic, lo que permite a los usuarios interactuar con la aplicación. Los desarrolladores pueden agregar acciones que se activan cuando un usuario hace clic en un botón y también pueden personalizar el aspecto y la sensación de los botones mediante la configuración de propiedades como el texto, la imagen, el color y la fuente. En resumen, JButton es una herramienta importante para la creación de interfaces de usuario interactivas en aplicaciones de escritorio Java.

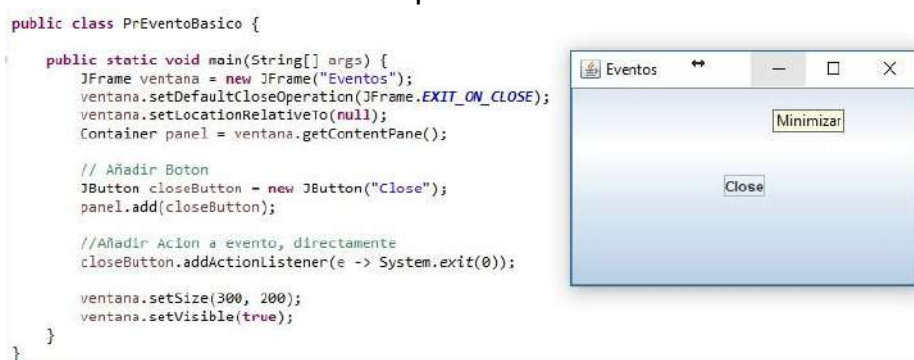


Imagen: Ejemplo del JButton

Fuente: <http://recursosformacion.com/wp-content/uploads/2018/05/eventosswingClose.png>

1.1.2 JLabel

En Java, JLabel es una clase que pertenece a la biblioteca gráfica Swing y que se utiliza para crear un componente de interfaz de usuario que muestra texto o una imagen. Un objeto JLabel se puede agregar a un contenedor, como un panel o un marco, para mostrar información en la pantalla. También se puede utilizar para etiquetar otros componentes de la interfaz de usuario. Un objeto JLabel puede tener diferentes propiedades, como el texto que muestra, la fuente y el color del texto, y la imagen que muestra. Estas propiedades se pueden establecer mediante métodos de la clase JLabel.

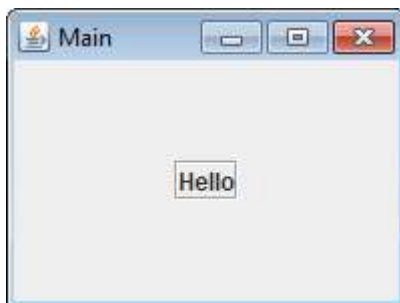


Imagen: Ejemplo del JLabel

Fuente: <https://www.buscaminegocio.com/img/jlabel.png>

1.1.3 JPasswordField

JTextField es un componente de la interfaz gráfica de usuario de Java Swing que permite al usuario ingresar texto de una sola línea. Es una caja de texto editable que se puede utilizar para recibir entradas de usuario en una aplicación de escritorio. El JTextField también permite que el desarrollador establezca un valor predeterminado, un tamaño máximo y un número máximo de caracteres que se pueden ingresar en el campo de texto. Además, el JTextField admite eventos de acción y de cambio, lo que permite al desarrollador controlar el comportamiento del campo de texto cuando se ingresan datos. En resumen, el JTextField es una herramienta importante para permitir la entrada de datos del usuario en una aplicación de escritorio de Java.

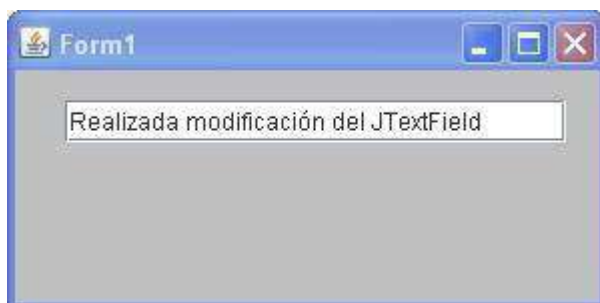


Imagen: Ejemplo del JTextField

Fuente: https://www.aprenderaprogramar.com/images/stories/Cursos/CU009/CU00928C_2.jpg

1.1.4 JTextArea

En Java, JTextArea es una clase que permite crear un área de texto en una interfaz gráfica de usuario (GUI, por sus siglas en inglés). Es similar a un cuadro de texto (JTextField), pero en lugar de un solo línea, JTextArea puede mostrar múltiples líneas de texto. Además, JTextArea admite la edición y selección de texto, así como la configuración de diferentes fuentes, estilos y alineaciones para el texto dentro del área. JTextArea también se puede utilizar para mostrar grandes cantidades de información en una ventana de la GUI.

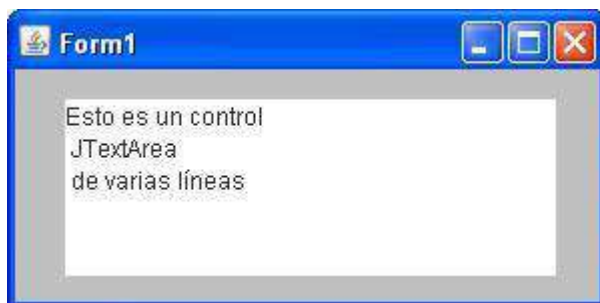


Imagen: Ejemplo del JTextArea

Fuente: https://www.aprenderaprogramar.com/images/stories/Cursos/CU009/CU00929C_2.jpg

1.1.5 JPasswordField

En Java, el JPasswordField es un componente de la interfaz gráfica de usuario (GUI) que se utiliza para ingresar contraseñas de manera segura. Este componente es similar al JTextField, pero oculta los caracteres que se ingresan en la pantalla y muestra un carácter de reemplazo, como asteriscos o puntos, para proteger la privacidad de la contraseña. El JPasswordField utiliza una matriz de caracteres para almacenar la contraseña en lugar de una cadena de caracteres, lo que proporciona una capa adicional de seguridad, ya que las matrices de caracteres son inmutables y no pueden ser compartidas ni recopiladas por el recolector de basura de Java.

Además, el JPasswordField proporciona métodos para acceder y manipular la contraseña, como getPassword() para obtener la contraseña como una matriz de caracteres y setPassword(char[]) para establecer la contraseña.



Imagen: Ejemplo del JTextArea

Fuente: [https://1.bp.blogspot.com/-](https://1.bp.blogspot.com/-oAINrikn31I/U_868b8dtkI/AAAAAAAAADZo/t9MAQ1d0QuI/s1600/UsoJPasswordField03.png)

[oAINrikn31I/U_868b8dtkI/AAAAAAAAADZo/t9MAQ1d0QuI/s1600/UsoJPasswordField03.png](https://1.bp.blogspot.com/-oAINrikn31I/U_868b8dtkI/AAAAAAAAADZo/t9MAQ1d0QuI/s1600/UsoJPasswordField03.png)

1.1 Java JCheckBox

Por: Brandon David Grijalva Marroquín

El JCheckBox Java posee una comprobación de tipo booleano donde al pulsar en una determinada caja, la forma de selección se pasa en dos opciones, la primera es true si la caja se ha seleccionado o false en caso contrario.

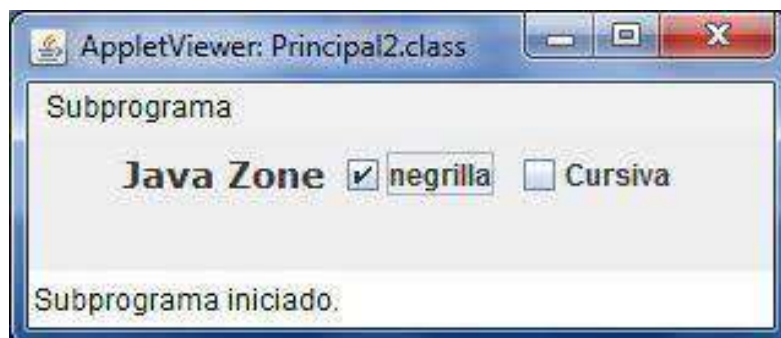


Imagen: Ejemplo de JCheckBox en Java

Fuente: https://www.google.com/url?sa=i&url=http%3A%2F%2Fwww.myjavazone.com%2F2010%2F06%2Fcheckbox.html&psig=AOvVaw1nTEpez8C_XouNMbLv23SD&ust=1680131905500000&source=images&cd=vfe&ved=0CA0QjRxgFwoTCliT_bvh_0CFQAAAAAdAAAAABAD

1.1.1 Java JRadioButton

Normalmente se muestran un conjunto de ellos y permiten la selección simultánea de uno. Se deben agrupar para que actúen en conjunto, es decir cuando se selecciona uno automáticamente se deben seleccionar los otros.

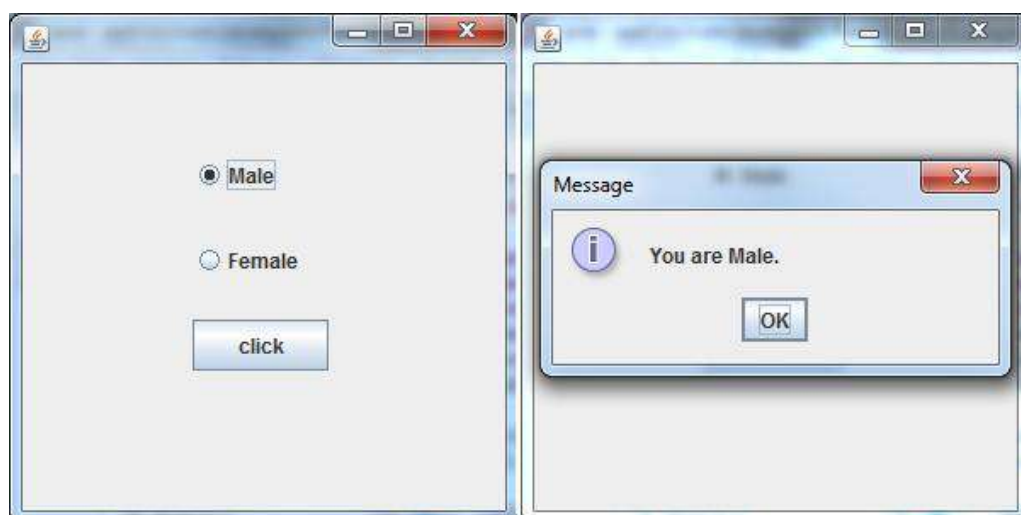


Imagen: Ejemplo de JRadioButton en Java

Fuente: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.javatpoint.com%2Fjava-jradiobutton&psig=AOvVaw0eP1Of16cZ7Yxw2xCuEox&ust=1680132282795000&source=images&cd=vfe&ved=0CA0QjRxgFwoTCMiYi4vk_0CFQAAAAAdAAAAABAD

1.1.2 Java JComboBox

Es una lista desplegable que almacena elementos y podemos agregar elementos desde una base de datos para su manipulación.



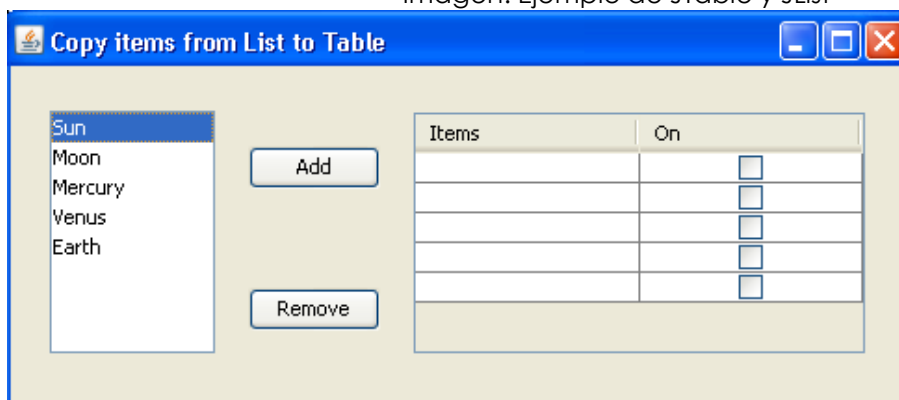
Imagen: Ejemplo de JComboBox en java

Fuente: https://www.google.com/url?sa=i&url=http%3A%2F%2Fcodejavu.blogspot.com%2F2016%2F04%2Fuso-basico-de-icomboBox-en-java.html&psig=AOvVaw2JigPwgRfyprHME6NyUzPg&ust=1680132514778000&source=images&cd=vfe&ved=0CA0QjRxaFwoTCLiUiOjj_0CFQAAAAAdAAAAABAD

1.1.3 Java JTable Java JList

Para el manejo de datos con JList y JTable debemos hacer uso de modelos. En el caso de las listas vamos a necesitar hacer uso de la clase DefaultListModel y para las tablas usaremos DefaultTableModel.

Imagen: Ejemplo de JTable y JList



Fuente: https://www.google.com/url?sa=i&url=https%3A%2F%2Fstackoverflow.com%2Fquestions%2F11101284%2Fcopy-items-from-jlist-to-jtable&psig=AOvVaw1n1i9RbgaQ2oLSCaT5y2p1&ust=1680132895268000&source=images&cd=vfe&ved=0CA0QjRxaFwoTCLiklPvk_0CFQAAAAAdAAAAABAH

1.1.4 Java JOptionPane

JOptionPane es una clase de java que muestra cuadros de diálogo y que podemos utilizar como alternativa a la clase Scanner.

```
//Archivo: PruebaJOption.java
import javax.swing.JOptionPane;

public class PruebaJOption{

    public static void main(String args[]){

        String nombre;
        nombre = JOptionPane.showInputDialog("¿Cual es tu nombre?");

        System.out.println("Hola "+nombre);
    }
}
```

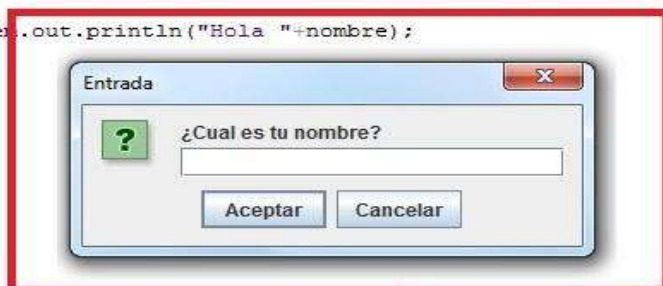


Imagen: Ejemplo de JOptionPane

Fuente: <https://www.google.com/url?sa=t&url=http%3A%2F%2Fbuendaintencions.blogspot.com%2F2011%2F10%2Fla-clase-joptionpane.html&psig=AVvGyGw1C8sG639iutG100bPzJQx&ust=10013143740000&source=images&cd=vfe&ved=0CA0QikxatwoTCMv7-T0CFQAAAAAAdAAAAABAD>

1.1 Java

Por: Braulio José Alexander Guarca Lima

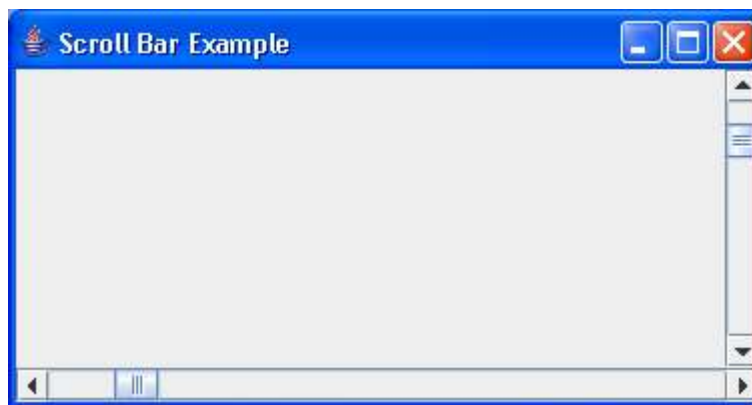
1.1.1 Java JScrollBar

Java JScrollBar es un componente de la interfaz gráfica de usuario (GUI) de Java que proporciona una barra de desplazamiento para desplazar el contenido de una ventana o un panel.

En otras palabras, JScrollBar permite a los usuarios desplazarse hacia arriba y hacia abajo (o hacia la izquierda y hacia la derecha, dependiendo de la orientación) en un área de contenido que es más grande que la ventana o el panel visible.

Este componente es muy útil para la visualización de grandes conjuntos de datos, como por ejemplo, en un editor de texto o en una tabla con muchas filas y columnas. JScrollBar se puede personalizar con diferentes opciones de estilo y comportamiento, como la orientación, el tamaño, el estilo de la barra y la posición de los botones de desplazamiento.

Imagen: Ejemplo de función de ScrollBar.



<http://www.java2s.com/Code/Java/Swing-JFC/AquickdemonstrationofJScrollBarbothverticalandhorizontal.htm>

1.1.2 Java JMenuItem & JMenu

JMenuItem y JMenu son componentes de la interfaz gráfica de usuario (GUI) de Java que se utilizan para crear menús desplegables en una aplicación.

JMenu es un contenedor que puede contener un conjunto de JMenuItem y otros submenús. En otras palabras, JMenu es un objeto que representa una opción de menú

principal, que se puede seleccionar para desplegar un conjunto de opciones secundarias.

Por otro lado, JMenuItem es un objeto que representa una opción de menú individual que se puede seleccionar. Por lo general, se coloca dentro de un JMenu y se utiliza para representar una acción que el usuario puede realizar en la aplicación, como guardar un archivo, imprimir un documento o salir de la aplicación.

Estos componentes son muy útiles para organizar y presentar opciones de acciones en una aplicación de manera clara y organizada. Además, se pueden personalizar con diferentes opciones de estilo y comportamiento, como la apariencia, el color, la fuente y el tamaño del texto.



Imagen: Visualización de JMenu.

<http://www.myjavazone.com/2010/06/imenubar-imenu-y-imenuitem.html>

1.1.3 Java JPopupMenu.

JPopupMenu es un componente de la interfaz gráfica de usuario (GUI) de Java que se utiliza para crear menús contextuales. Un menú contextual es un menú que aparece cuando el usuario hace clic con el botón derecho del ratón en un componente específico de la GUI, como un botón o un área de texto.

JPopupMenu se utiliza para proporcionar opciones de acciones adicionales que se relacionan con el objeto seleccionado en la GUI. Por ejemplo, en un editor de texto, se puede mostrar un menú contextual que ofrezca opciones como "copiar", "pegar", "cortar", etc. cuando el usuario haga clic con el botón derecho en una selección de texto.

Este componente es muy útil para mejorar la interacción del usuario en una aplicación de Java, ya que permite proporcionar opciones de acciones contextuales específicas al objeto seleccionado en la GUI. JPopupMenu se puede personalizar con diferentes opciones de estilo y comportamiento, como la apariencia, el color, la fuente y el tamaño del texto.

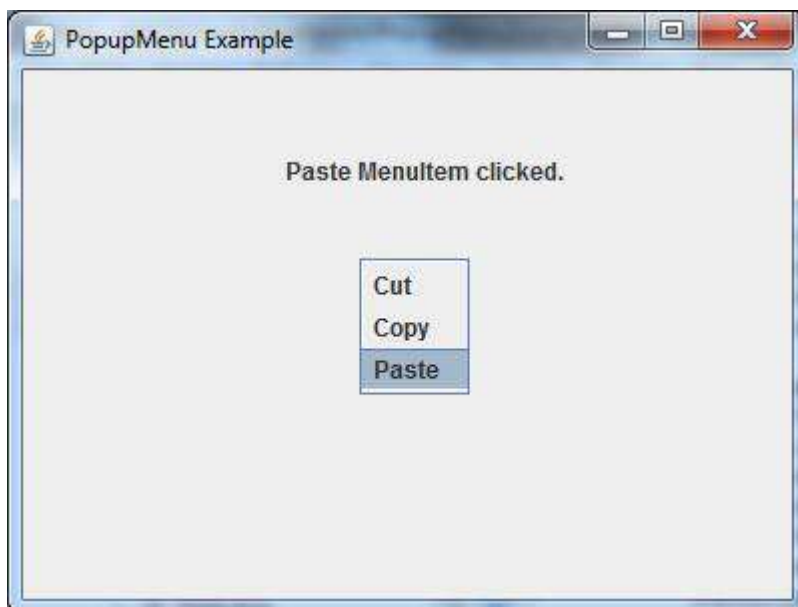


Imagen: Visualización de JPopupMenu.

<https://www.javatpoint.com/java-jpopupmenu>

1.1.4 Java JCheckBoxMenuItem.

JCheckBoxMenuItem es un componente de la interfaz gráfica de usuario (GUI) de Java que se utiliza para crear elementos de menú con casillas de verificación.

Este componente es similar a un JMenuItem, pero incluye una casilla de verificación que permite a los usuarios seleccionar o deseleccionar una opción de menú. JCheckBoxMenuItem se utiliza comúnmente para proporcionar opciones de configuración en una aplicación de Java, donde los usuarios pueden habilitar o deshabilitar una funcionalidad específica.

JCheckBoxMenuItem se puede personalizar con diferentes opciones de estilo y comportamiento, como la apariencia, el color, la fuente y el tamaño del texto, así como el estado de la casilla de verificación.



Imagen: Visualización de JCheckBoxItem.

http://www.java2s.com/Tutorial/Java/0240_Swing/InitializeJCheckBoxMenuItemwithcheckedoption.htm

1.1.5 Java JProgressBar.

Este componente se utiliza comúnmente para indicar al usuario el estado de una tarea que está en progreso, como la carga de un archivo o la descarga de un archivo desde Internet. JProgressBar muestra una barra de progreso que se llena gradualmente a medida que se completa la tarea, proporcionando una indicación visual del progreso de la tarea.

JProgressBar se puede personalizar con diferentes opciones de estilo y comportamiento, como la apariencia, el color y el tamaño de la barra, así como el texto que se muestra junto a la barra para indicar el porcentaje completado de la tarea.

En resumen, JProgressBar es una herramienta importante para mejorar la experiencia del usuario en una aplicación de Java al proporcionar una indicación visual del progreso de una tarea. Esto ayuda a los usuarios a comprender el tiempo estimado que llevará la tarea y proporciona una sensación de satisfacción al completar la tarea exitosamente.

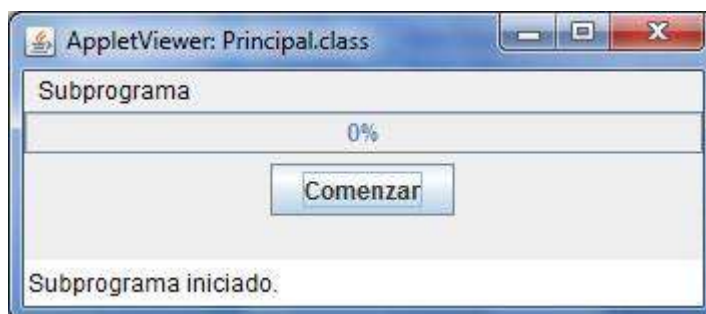


Imagen: Visualización de JProgressBar.

<http://www.myjavazone.com/2010/07/timer-y-jprogressbar.html>

1.1 Java JColorChooser.

Por: Eddy Junior Guerrero Gomez

El componente JColorChooser es un componente integrado en la biblioteca de java swing, y proporciona una interfaz gráfica que ofrece una selección de colores básicos, así como opciones para crear colores personalizados utilizando modelos de color como RGB, HSB y CMYK. Los usuarios pueden ver una vista previa del color seleccionado y ajustar la saturación, brillo y matiz para crear el color deseado.

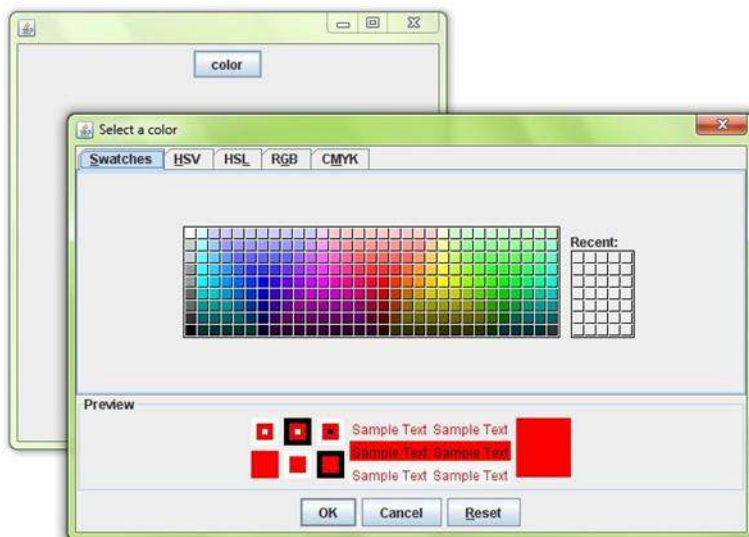


Imagen: vista grafica del componente JColorChooser

Fuente: <https://static.javatpoint.com/java/swing/images/java-icolorchooser1.png>

1.1.1 Java JTabbedPane.

Java JTabbedPane es un componente de interfaz gráfica de usuario. Se utiliza para crear pestañas en una ventana o dialogo y permite al usuario cambiar entre diferentes pestañas para ver diferentes conjuntos de componentes y funcionalidades. Este componente se utiliza comúnmente en aplicaciones de escritorio y son parte del paquete de biblioteca de clases de Java Swing.

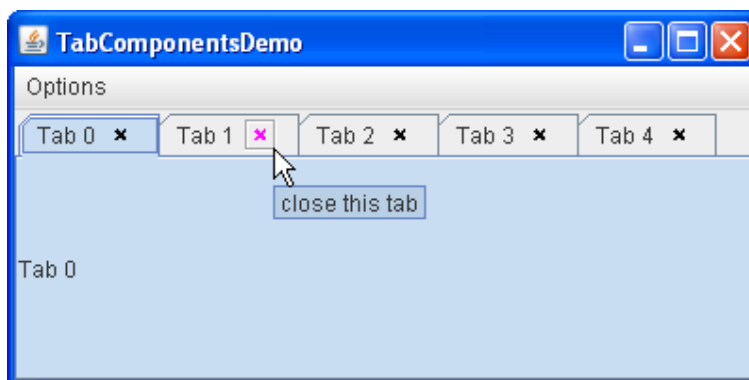


Imagen: vista de como se ven las pestañas con el componente JTabbedPane

Fuente: <https://docs.oracle.com/javase/tutorial/figures/ui/swing/components/TabComponentsDemo.png>

1.1.2 Java JSlider

Java JSlider es un componente de interfaz gráfica de usuario que pertenece a la biblioteca de Java Swing. El JSlider consiste en una barra horizontal o vertical que el usuario puede arrastrar para seleccionar un valor numérico dentro de un rango específico mediante un control deslizante, puede incluir etiquetas o marcas para indicar los valores en los extremos de la barra y los valores seleccionados por el usuario.

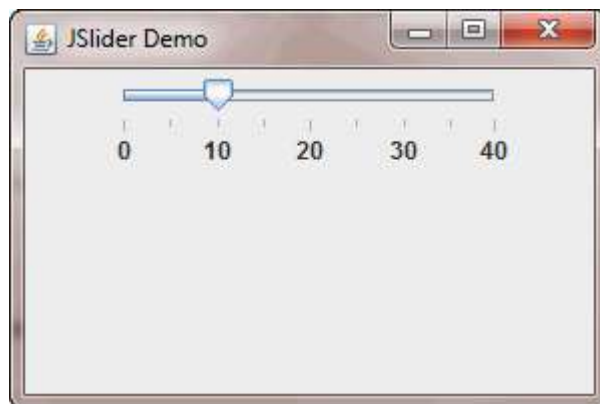


Imagen: Ejemplo de JSlider

Fuente: <https://www.zentut.com/wp-content/uploads/2012/10/JSlider.gif>

1.1.3 Java JSpinner

Java JSpinner es un componente de interfaz gráfica de usuario que pertenece a la biblioteca de Java Swing. Java JSpinner permite al usuario seleccionar un valor numérico o de fecha/hora dentro de un rango específico altamente personalizable, lo que significa que se puede adaptar a las necesidades específicas de la aplicación mediante una serie de botones de incremento y decremento o mediante una interfaz de calendario.



Imagen: Uso de Java JSpinner para guardar valores numéricos

Fuente: <https://i.stack.imgur.com/PwoGU.png>

1.1.4 Java JDialog

Java JDialog es un componente de interfaz gráfica de usuario que pertenece a la biblioteca de Java Swing. Java JDialog es utilizado para mostrar ventanas emergentes en aplicaciones de escritorio de Java y Pueden proporcionar notificaciones importantes, mensajes de confirmación y diálogos de entrada de datos, y se pueden personalizar para ajustarse a la apariencia y el estilo de la aplicación.

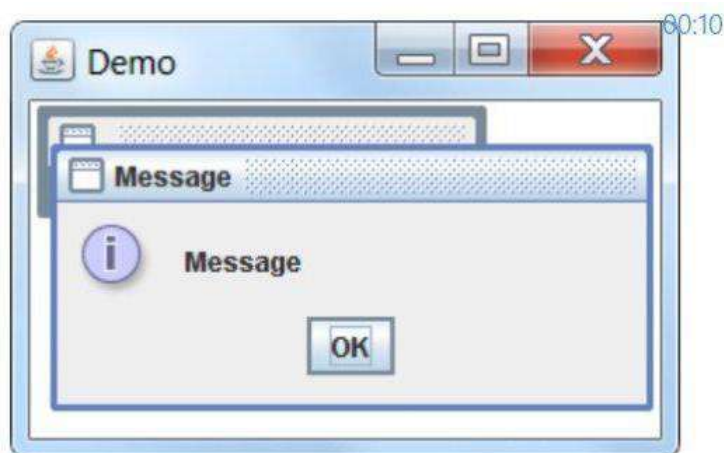


Imagen: Ejemplo de mensajes emergentes con Java JDialog

Fuente: <https://aph.cf2.quoracdn.net/main-qimg-4c2843db92b792d81f4c86605d7cf311-pila>

1.1.5 Java JPanel

Java JPanel es un componente de interfaz gráfica de usuario que pertenece a la biblioteca de Java Swing. Java JPanel es utilizado para organizar y contener otros componentes de la interfaz gráfica de usuario, como botones, campos de texto, etiquetas y otros paneles. Además, los JPanel pueden tener diferentes diseños, colores, fuentes y bordes, lo que permite una mayor personalización y flexibilidad en el diseño de la interfaz de usuario.

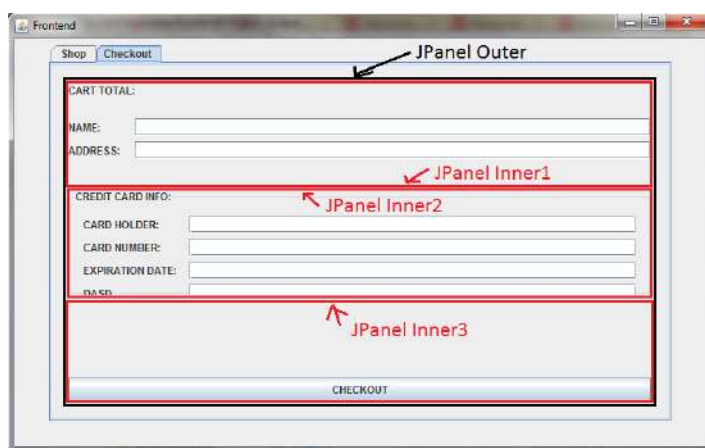


Imagen: Ejemplo de implementación de JPanel
Fuente: <https://i.stack.imgur.com/v8apG.png>

1.1.6 Java JFileChooser

Java JFileChooser es un componente de interfaz gráfica de usuario que pertenece a la biblioteca de Java Swing. Java JFileChooser permite a los usuarios seleccionar archivos o directorios en su sistema de archivos. El componente JFileChooser proporciona una ventana emergente con una interfaz de usuario estándar para que los usuarios elijan un archivo o un directorio.

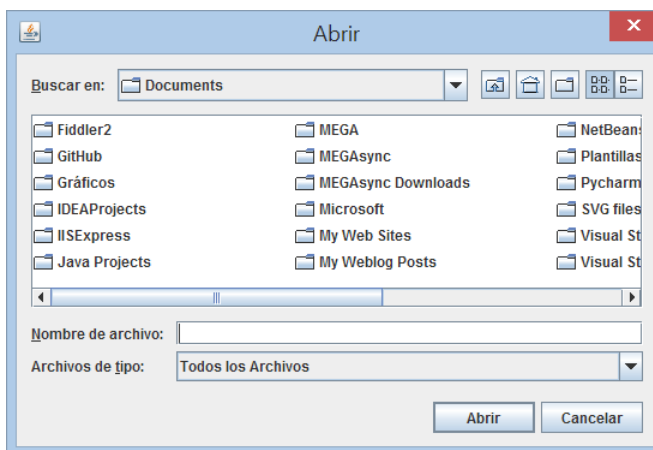


Imagen: Interfaz grafica de Java JFileChooser

Fuente:

https://lh3.googleusercontent.com/--ryd7B0Ntm8/WUU15tZE0xl/AAAAAAAAADGU/hZRg-TEDheonuKbbCGhUgwC_6vQY2GhoACHMYCw/image_thumb1?imgmax=800

1.1 TitJToggleButtonulo

Por: Ricardo Alberto Hernández Bautista

Un ToggleButton en programación es un botón que tiene dos estados (toggle alternar, conmutar, en inglés) «pulsado» o «no pulsado». Se mantiene presionado ~~o~~ se pulsa en él. Se puede utilizar como un conmutador de encendido/apagado.



Imagen: Ejemplo de JToggleButton

Fuente: <https://masaqueprogramar.files.wordpress.com/2019/03/ejemploitogglebutton.png>

1.1.1 Java JToolBar

Un JToolBar es una barra de herramientas, la cual nos facilita el uso de alguna opción de nuestro programa. A esta barra se le puede añadir varios componentes ya sean etiquetas, botones, campos de texto, etc.

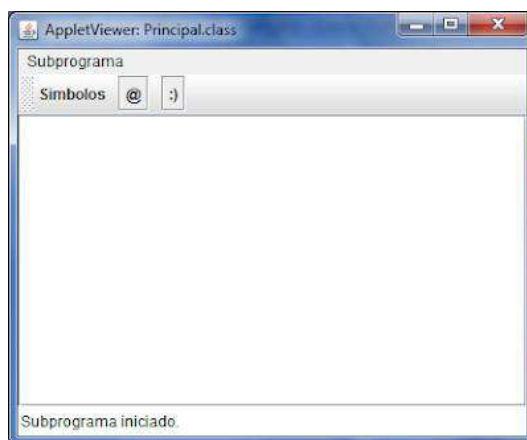


Imagen: Ejemplo de JToolBar.

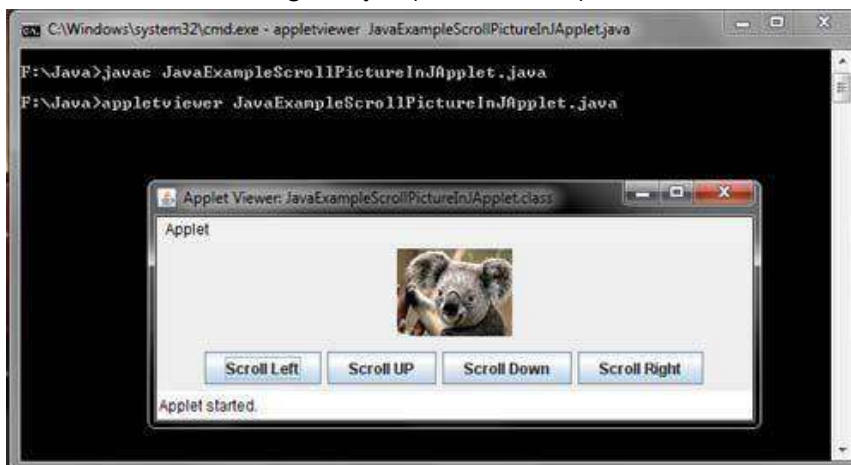
Fuente:

http://2.bp.blogspot.com/_73t-OXs9bAc/TTidYaDHnol/AAAAAAAAALc/0oMiBX4X-3A/s400/toolbar1.png

1.1.2 Java JViewport

La "ventana" o "ojo de buey" a través del cual ve la información subyacente. Cuando desplazas, lo que se mueve es la ventana gráfica. Es como mirar a través del visor de una cámara. Al mover el visor hacia arriba, aparecen cosas nuevas en la parte superior de la imagen y se pierden las que estaban en la parte inferior.

Imagen: Ejemplo de JViewport.



Fuente: <https://ecomputernotes.com/images/JavaExampleScrollPictureInJApplet.ipa>

1.1.3 Java JFrame

JFrame es una clase que se encuentra en el paquete javax Swing (biblioteca gráfica) para generar ventanas o marcos donde se pueden colocar componentes (Contenedores, Controles, etc). La mayoría de las ventanas son instancias de la clase.

- JFrame o subclases de JFrame.
- JFrame es una subclase que extiende de la clase Frame
- JFrame proporciona los atributos y comportamientos básicos de una ventana.

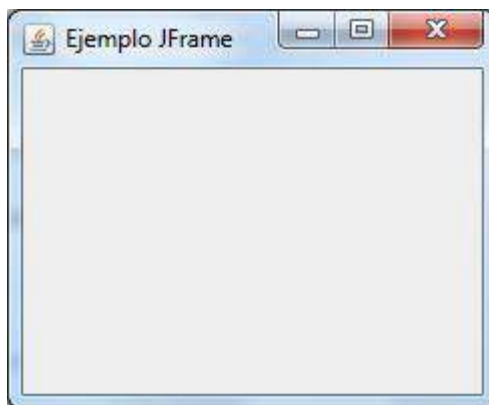


Imagen: Ejemplo de JFrame.

Fuente: <https://es.wikipedia.org/wiki/Jframe>

1.1.4 Java JComponent

Con la excepción de los contenedores de nivel superior, todos los componentes de Swing, cuyos nombres comienzan con "J", descienden de la clase JComponent. Por ejemplo, JPanel, JScrollPane, JButton y JTable heredan de JComponent. Sin embargo, JFrame y JDialog no lo hacen porque implementan contenedores de nivel superior.

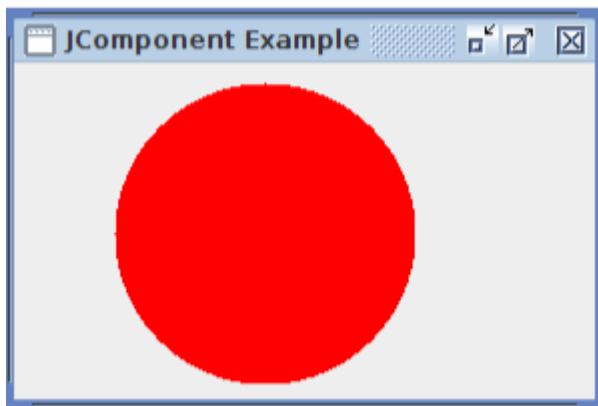


Imagen: Ejemplo de JComponent.

Fuente: <https://www.javacodex.com/Swing/JComponent>

1.1.5 Java JDesktop

JDesktopPane es una clase disponible en JavaFX que se puede usar para generar interfaces multi documentadas, que pueden contener muchas ventanas o páginas de aplicaciones. Se logra haciendo uso de ContentPane y JInternalFrame desde la ventana principal y la ventana interna respectivamente.

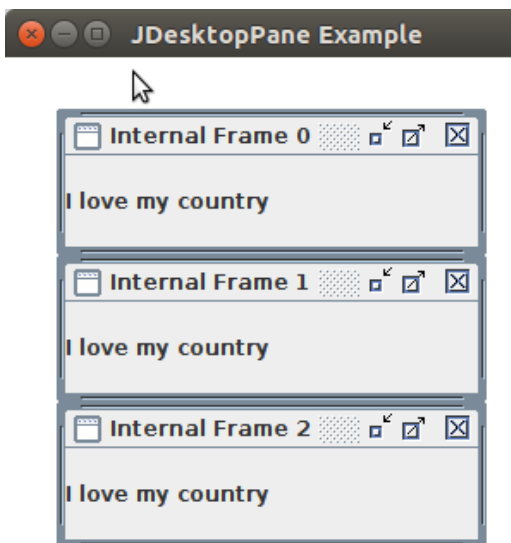


Imagen: Ejemplo de JDesktop.

Fuente: <https://www.javatpoint.com/java-jdesktoppane>

1.1.6 Java JEditorPane

Un JEditorPane es un componente que nos permite editar texto, es mas completo que el JTextField y el JTextArea, ademas acepta formato html, rtf.

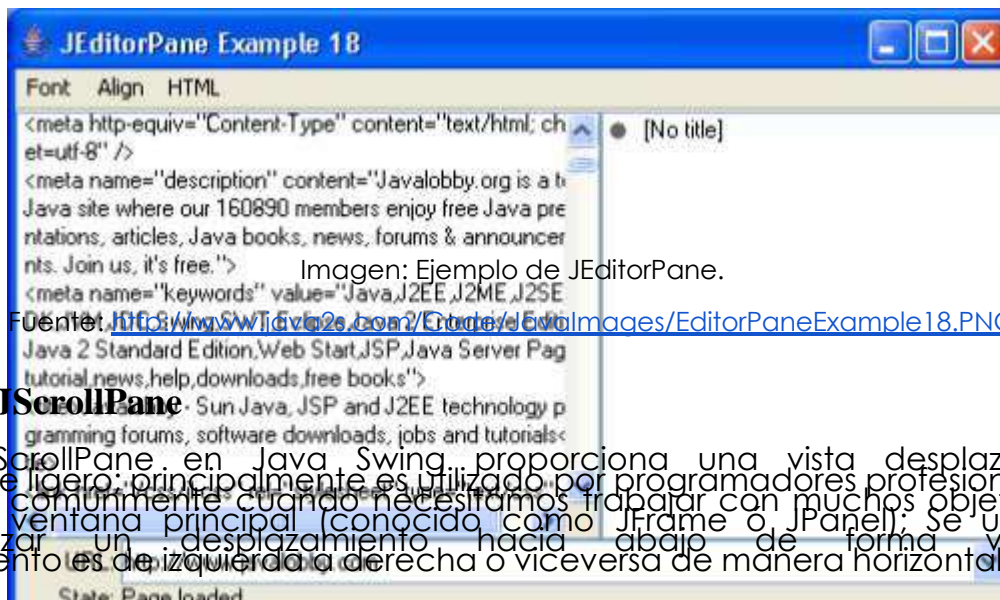


Imagen: Ejemplo de JEditorPane.

Fuente: <https://www.swtjava.com/Content/Images/EditorPaneExample18.PNG>

1.1.7 Java JScrollPane

Un JScrollPane en Java Swing proporciona una vista desplazable de un componente ligero; principalmente es utilizado por programadores profesionales y de nuestra experiencia comúnmente cuando necesitamos trabajar con muchos objetos y se salen de nuestra ventana principal (conocida como JFrame o JPanel); se utiliza el objeto JScrollPane para realizar un desplazamiento hacia abajo de forma vertical o su desplazamiento es de izquierda a derecha o viceversa de manera horizontal.



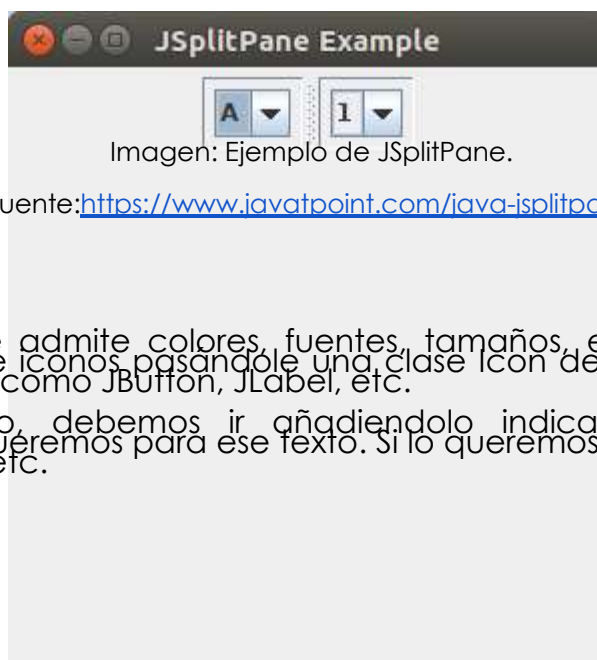
Imagen: Ejemplo de JScrollPane.

Fuente: <https://www.buscaminegocio.com/img/ScrollPane.jpg>

1.1.8 Java JSplitPane

El JSplitPane se usa para dividir el JFrame o el JPanel con una barra divisoria, dicha ventana para poder colocar en ella dos y solo dos componentes, mismos que pueden ser alineados de izquierda a derecha o de arriba hacia abajo.

Un SplitPane se utiliza cuando dos componentes contienen información relacionada y queremos que el usuario pueda cambiar el tamaño de los componentes en relación a uno o a otro. Un uso común de un SplitPane es para contener listas de elecciones y una visión de la elección actual.



1.1.9 Java JTextPane

Es otro editor que admite colores, fuentes, tamaños, etc. Admite incluso que pongamos directamente íconos pasándole una clase Icon de Java o que le añadamos un componente de Java como JButton, JLabel, etc.

Para añadir texto, debemos ir añadiéndolo indicando en cada caso los atributos que queremos para ese texto. Si lo queremos en negrita, con una fuente de letra, tamaño, color, etc.

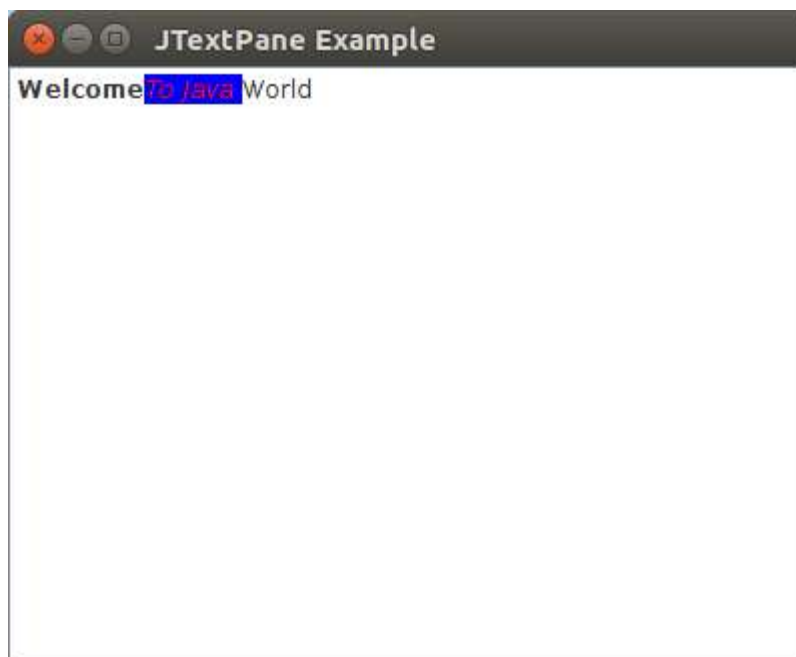


Imagen: Ejemplo de JTextPane.

Fuente: <https://www.javatpoint.com/java-jtextpane>

1.1.10 Sistema Android

Android es un sistema operativo móvil desarrollado por Google y diseñado principalmente para dispositivos móviles, como teléfonos inteligentes y tabletas. Es un sistema operativo de código abierto basado en el núcleo de Linux y se ha convertido en uno de los sistemas operativos móviles más populares en todo el mundo.

Imagen: Ejemplo de icono de sistema android.

Fuente: <https://tecnomari.es/blog/ventajas-inconvenientes-sistema-operativo-android/>



4. Sistema Android

4.1. Historia

Por: José Santiago Herrera Furlán

Android Inc. fue fundada en octubre de 2003 en Palo Alto, California, por Andy Rubin, Rich Miner, Nick Sears y Chris White. En un principio, la idea era desarrollar un sistema operativo avanzado para cámaras digitales, pero luego decidieron enfocarse en crear un sistema operativo para teléfonos móviles que rivalizaría con Symbian y Microsoft Windows Mobile. Andy Rubin describió el proyecto como una oportunidad para desarrollar dispositivos móviles más inteligentes y conscientes de la ubicación y las preferencias del propietario.

Inicialmente, la empresa tuvo dificultades para atraer inversores y se enfrentó al desalojo de su espacio de oficina. Steve Perlman, un amigo cercano de Rubin, le trajo \$10,000 en efectivo en un sobre y poco después transfirió una cantidad no revelada como capital inicial. Perlman rechazó una participación en la empresa y declaró que lo hizo porque creía en la cosa y quería ayudar a Andy.

En abril de 2004, Android presentó su idea a los inversores, pero la empresa decidió que el mercado de las cámaras no era lo suficientemente grande para sus objetivos y desvió sus esfuerzos para crear un sistema operativo para teléfonos móviles. En agosto de 2005, Google compró Android Inc. por una suma no revelada, y el equipo de Android pasó a formar parte de Google.

4.2. Adquisición por parte de Google

Por: José Santiago Herrera Furlán

En julio de 2005, Andy Rubin, el cofundador de Android, se reunió con Larry Page y Sergey Brin, los fundadores de Google, para discutir una posible adquisición. En ese momento, Android era una empresa relativamente pequeña y estaba teniendo dificultades para atraer inversores y hacer crecer su negocio.



android

Imagen: Logo actual de Android

Fuente: <https://commons.wikimedia.org/w/index.php?curid=81546554>

En agosto de 2005, Google anunció que había adquirido Android por una suma no revelada. Se cree que el precio de compra fue de alrededor de \$50 millones de dólares, aunque esto nunca ha sido confirmado oficialmente.

Después de la adquisición, el equipo de Android se unió a Google y comenzó a trabajar en el desarrollo de lo que eventualmente se convertiría en el sistema operativo móvil más popular del mundo.

4.3. Historial de actualizaciones

Por: José Santiago Herrera Furlán

Nombre	Versión	Fecha de lanzamiento
Android 1.0	1.0	23 de septiembre de 2008
Android 1.1	1.1	9 de febrero de 2009
Android Cupcake	1.5	27 de abril de 2009
Android Donut	1.6	15 de septiembre de 2009
Android Eclair	2.0	27 de octubre de 2009
	2.0.1	3 de diciembre de 2009
	2.1	11 de enero de 2010
Android Froyo	2.2 - 2.2.3	20 de mayo de 2010
Android Gingerbread	2.3 - 2.3.2	6 de diciembre de 2010
	2.3.3 - 2.3.7	9 de febrero de 2011
Android Honeycomb	3.0	22 de febrero de 2011
	3.1	10 de mayo de 2011
	3.2 - 3.2.6	15 de julio de 2011

Android Ice Cream Sandwich	4.0 - 4.0.2	18 de octubre de 2011
	4.0.3 - 4.0.4	16 de diciembre de 2011
Android Jelly Bean	4.1 - 4.1.2	9 de julio de 2012
	4.2 - 4.2.2	13 de noviembre de 2012
	4.3 - 4.3.1	24 de julio de 2013
Android KitKat	4.4 - 4.4.4	31 de octubre de 2013
	4.4W - 4.4W.2	25 de junio de 2014
Android Lollipop	5.0 - 5.0.2	4 de noviembre de 2014
	5.1 - 5.1.1	2 de marzo de 2015
Android Marshmallow	6.0 - 6.0.1	2 de octubre 2015
Android Nougat	7.0	22 de agosto de 2016
	7.1 - 7.1.2	4 de octubre de 2016
Android Oreo	8.0	21 de agosto de 2017
	8.1	5 de diciembre de 2017
Android Pie	9	6 de agosto de 2018
Android 10	10	3 de septiembre de 2019
Android 11	11	8 de septiembre de 2020
Android 12	12	4 de octubre de 2021
Android 12L	12.1	7 de marzo de 2022
Android 13	13	15 de agosto de 2022

4.4. Versiones

Por: José Santiago Herrera Furlán

A continuación, se describe de forma muy general las versiones de Android anteriormente mencionadas y sus objetivos.

- **Android 1.0 y 1.1:** Sistema operativo básico para smartphones.
- **Android Cupcake y Donut:** Mejoras en la interfaz de usuario y la navegación.
- **Android Eclair:** Mejoras en la velocidad y rendimiento del sistema.
- **Android Froyo y Gingerbread:** Mejoras en la velocidad y estabilidad del sistema.
- **Android Honeycomb:** Diseñado específicamente para tabletas.
- **Android Ice Cream Sandwich y Jelly Bean:** Mejoras en la interfaz de usuario, la velocidad y la estabilidad.
- **Android KitKat:** Mejoras en la seguridad y la optimización de recursos.
- **Android Lollipop y Marshmallow:** Mejoras en la interfaz de usuario, la seguridad y la duración de la batería.
- **Android Nougat:** Mejoras en la eficiencia y la multitarea.
- **Android Oreo:** Mejoras en la velocidad y la seguridad, incluyendo la optimización para dispositivos con pantalla grande.
- **Android Pie:** Enfoque en la inteligencia artificial y la navegación basada en gestos.
- **Android 10:** Enfoque en la privacidad y la seguridad.
- **Android 11:** Mejoras en la gestión de conversaciones y dispositivos inteligentes.
- **Android 12:** Enfoque en la personalización y la privacidad.
- **Android 12L:** Enfoque en la optimización para pantallas plegables y funciones multitarea.
- **Android 13:** Mejoras en la experiencia de usuario y control de aplicaciones.

4.5. Arquitectura

Por: José Santiago Herrera Furlán

La arquitectura de Android se basa en un modelo de capas que proporciona un marco para el desarrollo de aplicaciones y servicios en el sistema operativo. La arquitectura se compone de cuatro capas principales:

- **Capa de aplicaciones:** Donde se encuentran las aplicaciones que el usuario ve y utiliza. Esta capa también incluye los servicios de los proveedores de contenido.
- **Capa de Framework de Aplicaciones:** Que proporciona una variedad de API para permitir que las aplicaciones interactúen con el sistema operativo y entre sí.
- **Capa de Bibliotecas:** Donde se hayan las bibliotecas escritas en C/C++ utilizadas para el desarrollo de aplicaciones.
- **Runtime de Android:** El sistema de Android utiliza su propia máquina virtual basada en la máquina virtual de Java para la ejecución de los programas.
- **Kernel de Linux:** que proporciona los controladores necesarios para que el sistema operativo se comunique con el hardware, como la pantalla, la cámara y los sensores.

La arquitectura de Android está diseñada para ser modular, lo que significa que las diferentes partes del sistema operativo se pueden actualizar y mejorar sin afectar al resto del sistema. Además, Android está diseñado para ser flexible y personalizable, lo que permite a los fabricantes de dispositivos personalizar el sistema operativo para sus dispositivos específicos.

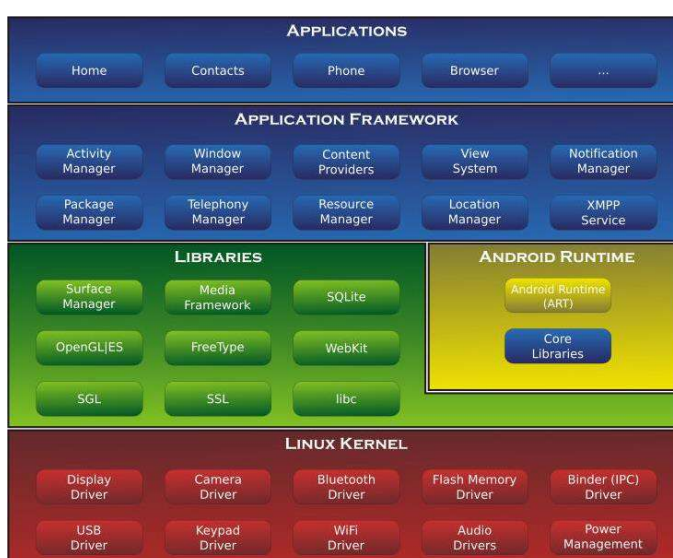


Imagen: Arquitectura del Sistema Android

Fuente: <https://commons.wikimedia.org/wiki/File:Android-System-Architecture.svg>

Seguridad, privacidad y vigilancia.

Por: Jose Jordan

Seguridad en Android: Android cuenta con varias medidas de seguridad para proteger el sistema operativo y los datos del usuario.

Privacidad en Android: Android también tiene varias características para proteger la privacidad del usuario. Por ejemplo, cuenta con controles de privacidad para las aplicaciones que permiten al usuario controlar el acceso a la información personal.

Vigilancia en Android: Aunque Android tiene medidas de seguridad y privacidad, también existe el riesgo de vigilancia, tanto por parte de aplicaciones maliciosas como por parte de gobiernos y otras entidades.

Usos y dispositivos.

Dispositivos:

Smartphones: Android es el sistema operativo móvil más utilizado en smartphones en todo el mundo, con una gran variedad de marcas y modelos disponibles.

Tabletas: Muchas tabletas también utilizan Android como sistema operativo, requieren una experiencia de usuario similar a la de un teléfono inteligente pero en un formato de pantalla más grande.

Smartwatches: Android también se utiliza en smartwatches y otros dispositivos portátiles, lo que permite a los usuarios realizar diversas tareas sin necesidad de sacar su teléfono del bolsillo.

Usos:

Comunicación: Los smartphones con Android son ampliamente utilizados para la comunicación, incluyendo llamadas, mensajes de texto, correo electrónico y aplicaciones de mensajería instantánea.

Entretenimiento: Android también se utiliza para una variedad de actividades de entretenimiento, como ver videos en línea, jugar juegos, escuchar música y leer libros electrónicos.

Productividad: Muchas aplicaciones de productividad están disponibles en Android, incluidas aplicaciones de correo electrónico, calendarios, procesadores de texto, hojas de cálculo y aplicaciones de notas.

Control de dispositivos inteligentes: Los dispositivos Android también se pueden utilizar para controlar otros dispositivos inteligentes.

Diseño y desarrollo del sistema Android

El diseño y desarrollo del sistema operativo Android se basa en una arquitectura de software de código abierto, lo que significa que cualquier desarrollador puede acceder a la fuente de código y modificarlo o mejorar el sistema según sus necesidades.

Mercadotecnia.

la mercadotecnia de Android se basa en la asociación con fabricantes de dispositivos, la promoción de la marca, la publicidad en línea, la promoción de la tienda de aplicaciones y el patrocinio de eventos. Estas estrategias han sido efectivas para posicionar a Android como uno de los sistemas operativos móviles más populares del mundo.



Imagen: la gran batalla entre Android e iOS también sigue su curso

Fuente: <https://marketing4ecommerce.net/android-o-ios-que-sistema-operativo-consigue-mas-conversiones-flat101/>

Entorno Android Studio.

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Este entorno de desarrollo se basa en el popular IDE de desarrollo de software IntelliJ IDEA de JetBrains.

Android Studio proporciona a los desarrolladores un conjunto completo de herramientas para el desarrollo de aplicaciones, incluido un editor de código, un depurador, un emulador de dispositivos y herramientas de diseño visual.

1.1 Instalación de Android Studio

Por: Angel Enrique Juarez Asencio

Paso 1

Descargar el instalador: Ve al sitio web oficial de Android Studio, (<https://developer.android.com/studio>) y haz clic en el botón "Download Android Studio". Selecciona tu sistema operativo (Windows, macOS o Linux) y descarga el archivo de instalación.

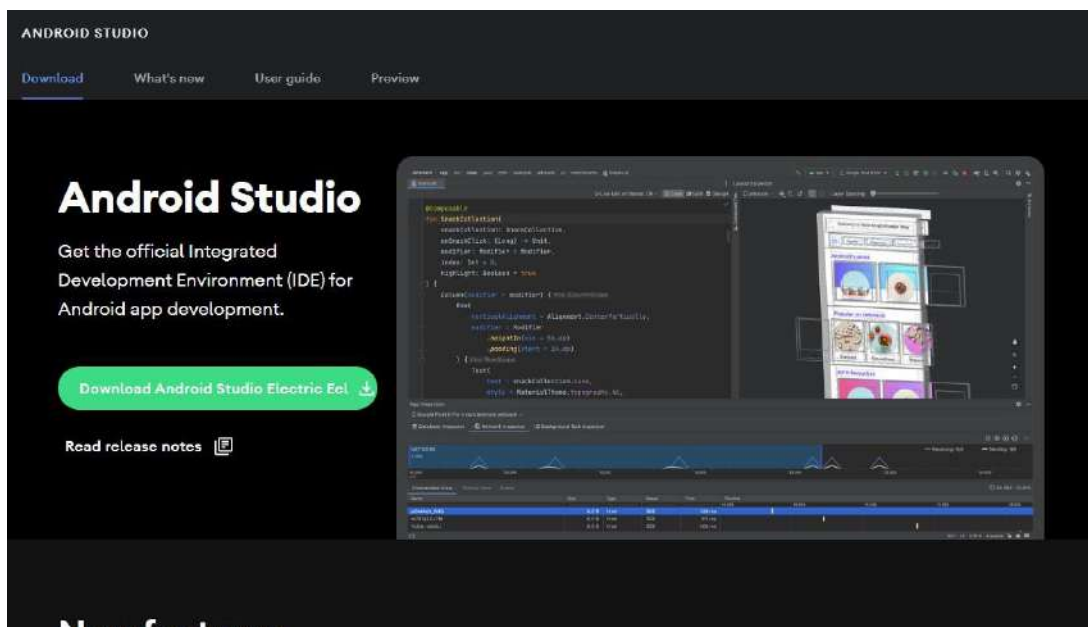


Imagen: pagina para descargar Android studio
Fuente: <https://developer.android.com/studio>

Paso 2

Ejecutar el instalador: Una vez que se haya descargado el archivo de instalación, ábrelo y sigue las instrucciones para instalar Android Studio en tu ordenador.

Paso 3

Aceptar las licencias: Durante la instalación, se te pedirá que aceptes las licencias de Android SDK y Android Studio. Lee los terminos y condiciones y si estas de acuerdo, marca la casilla para aceptarlos.

Paso 4

Seleccionar componentes: En la siguiente pantalla, se te pedirá que selecciones los componentes que deseas instalar. Por la general, se recomienda seleccionar todos los de Android.

Paso 5

Seleccionar la ubicación de instalación: En la siguiente pantalla, se te pedirá que selecciones la ubicación donde deseas instalar Android Studio. Por lo general, se recomienda dejar la ubicación predeterminada.

Paso 6

Esperar a que se complete la instalación: Una vez que hayas seleccionado la ubicación de instalación, haz clic en el botón "Install" y espera a que se complete la instalación.

Paso 7

Configurar Android Studio: Una vez que se haya completado la instalación, inicia Android Studio y sigue las instrucciones para configurar la herramienta. Esto puede incluir la instalación de algunas actualizaciones o componentes adicionales.

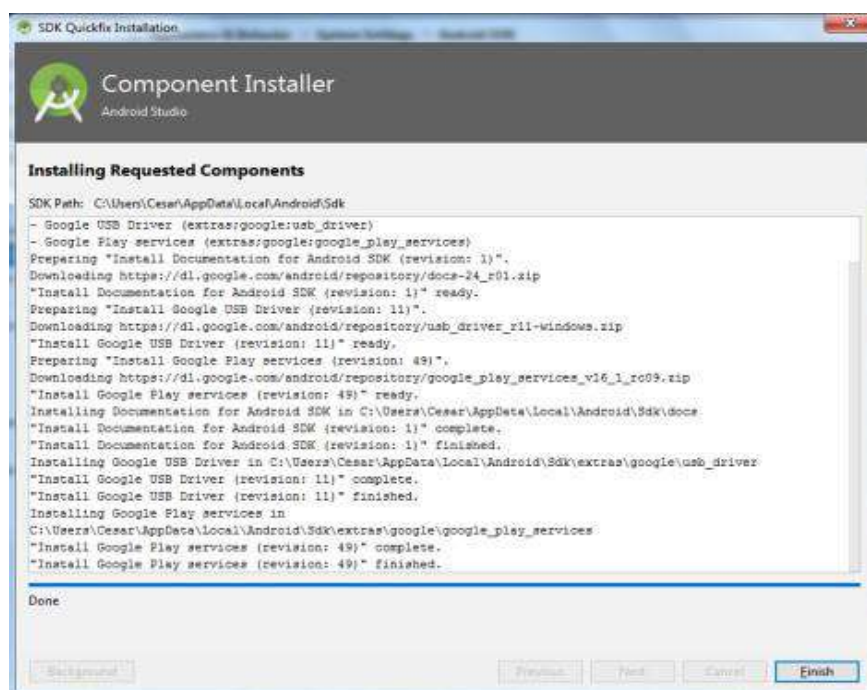


Imagen: último paso de instalación

Fuente: http://lcsi.umh.es/docs/androidTFG/instalacion_android_studio.pdf

1.1.1 Pasos para crear el primer trabajo

Paso 1

Abre Android Studio: Una vez instalado Android Studio, ábrelo desde el menú de aplicaciones o el escritorio de tu ordenador.



Imagen: creación de un nuevo proyecto .

Paso 2

Fuente: <https://developer.android.com/training/basics/firstapp/creating-project?hl=es-419>

Crea un nuevo proyecto: En la pantalla de bienvenida, haz clic en la opción "Start a new Android Studio project" o ve al menú "File" y selecciona "New" > "New Project".

Paso 3

Configura tu proyecto: Se te pedirá que configures algunos detalles de tu proyecto, como el nombre, el paquete de la aplicación, la ubicación de almacenamiento, la versión mínima de Android compatible y la plantilla de actividad inicial. Completa los campos según tus necesidades y haz clic en "Finish" para crear el proyecto.

Paso 4

Espera a que se cree el proyecto: Android Studio creará tu proyecto y puede tardar unos minutos en completarlo. Una vez que se haya creado, verás la pantalla principal de Android Studio, que muestra la estructura de tu proyecto.

Paso 5

Explora tu proyecto: En la ventana de la izquierda, verás la estructura de tu proyecto. Puedes expandir cada carpeta para ver los archivos y recursos que contiene. En la ventana de la derecha, verás la vista previa de la actividad principal de tu aplicación.

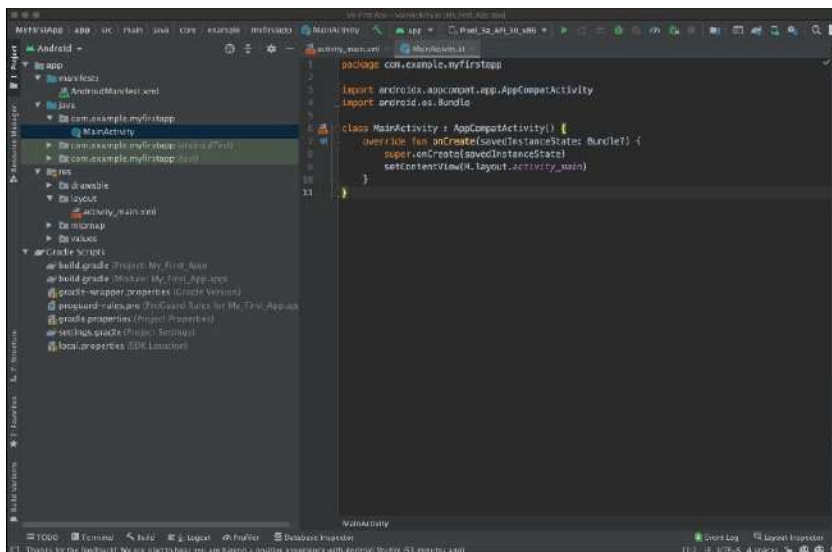


Imagen: exportación de un proyecto.

Fuente: <https://developer.android.com/training/basics/firstapp/creating-project?hl=es-419>

1.1.2 Capturar el click de un botón

Para capturar el click de un botón en Android Studio, debes seguir los siguientes pasos:

En tu archivo XML de diseño, agrega un botón con un ID único:

Code:

```

<Button android:id="@+id/boton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Clickea aquí"/>
    
```

1.1.3 controles de radion group

En Android Studio, el control de RadioGroup se utiliza para agrupar una serie de botones de opción (RadioButtons) en un conjunto exclusivo, es decir, el usuario solo puede seleccionar una opción del conjunto.

Para utilizar un RadioGroup en tu aplicación Android, debes seguir los siguientes pasos:
 En tu archivo XML de diseño, agrega un RadioGroup con un ID único:

Code:

```
<RadioGroup android:id="@+id/radio_group"
  android:layout_width="match_parent"
  android:layout_height="wrap_content">

  <!-- Agrega aquí los RadioButtons que deseas agrupar -->

</RadioGroup>
```

Dentro del RadioGroup, agrega los RadioButtons correspondientes a las opciones que deseas presentar al usuario. Asegurate de que cada RadioButton tenga un ID único y que el atributo "android:checked" este establecido en "true" para la predeterminada.

1.1.3.1 Control checkbox

En Android Studio, un CheckBox es un control que permite al usuario seleccionar una o varias opciones de una lista de opciones. Es similar al control RadioButton, pero en lugar de agrupar opciones exclusivas, CheckBox permite al usuario seleccionar varias opciones.

Para utilizar un CheckBox en tu aplicación Android, debes seguir los siguientes pasos:

En tu archivo XML de diseño, agrega un CheckBox con un ID único y establece el texto que se mostrará junto al control.

Code :

```
<CheckBox android:id="@+id/check_box"
  android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="Opción"/>
```

En el código Java, declara una variable para el CheckBox y úsala para capturar el estado del control

1.1 Control Spinner.

Por: Gustavo Juarez

Android Spinner es solo una lista desplegable similar a lo que se ve en otros lenguajes de programación, como en las páginas HTML. En Android, Spinner se usa para seleccionar un valor de un conjunto de valores. En el estado predeterminado, una ruleta muestra su valor seleccionado actualmente.

1.1.1 Cómo puedo usar más de una ruleta en Android

Puede compartir el adaptador entre diferentes Spinner si adaptaran la misma información. Claramente, cada uno de sus adaptadores necesita adaptar un conjunto diferente de Strings, por lo que necesita crear un ArrayAdapter para cada Spinner. Un solo OnItemSelectedListener funcionará para los 3 Spinners.

1.1.2 Cómo obtengo un valor de ruleta

Recuperar el valor del elemento seleccionado de la lista desplegable Spinner en Android. Spinner es un tipo de menú de lista desplegable y puede seleccionar su elemento tocando en la pantalla del móvil. La función OnItemSelectedListener () se aplica para obtener el valor del elemento seleccionado del menú de la lista desplegable de la ruleta.

1.1.3 Qué es spinner en HTML

Una ruleta le permite aumentar o disminuir un número en un cuadro haciendo clic en las flechas hacia arriba y hacia abajo. Se ven así en el navegador Opera: El código HTML para el cuadro giratorio se ve así:

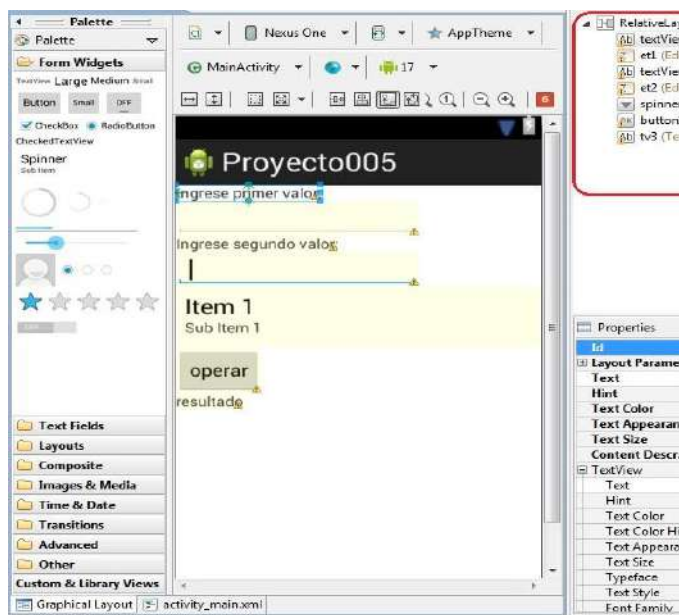


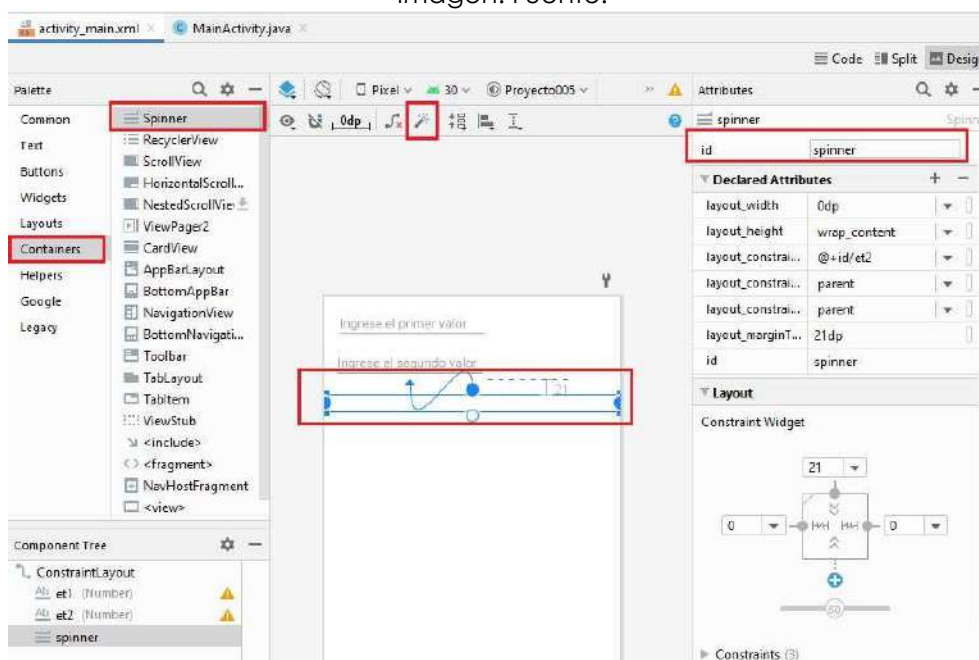
Imagen:

Fuente: <https://www.tutorialesprogramacionya.com/javaya/androidya/imagentema/foto030.jpg>

1.1.4 Cómo podemos enviar valor giratorio de una actividad a otra

La clase pública MySpinnerSelectedListener implementa OnItemSelectedListener {public void onItemSelected(AdapterView parent, View view, int pos, long id) (String selected = parent.getItemAtPosition (pos). Encadenar()); } public void onNothingSelected (AdapterView parent) { // No hacer nada. }}

Imagen: Fuente:



<https://www.tutorialesprogramacionya.com/javaya/androidya/androidstudioya/imagentema/foto029.jpg>

1.2 Control ListView (lista de String)

Por: Gustavo Juarez

El funcionamiento interno de un **ListView** puede ser algo complicado de comprender para un programador sin experiencia. La verdad es que es solo un control complicado de comprender. Sin embargo, no todo está perdido, porque una vez que sabe lo que hace que un **ListView** funcione, literalmente no hay ningún proyecto en el que pueda pensar que no necesite un **ListView**.

1.2.1 flotar

En un elemento de la lista, el puntero del mouse muestra el menú de acciones subyacente. La dirección de la revelación se puede especificar mediante el paso 'left' o el parámetro, o con la propiedad 'right' de objetos flotantes. `.direction` El objeto flotante puede tener una propiedad `timey` una `timeout` para definir el tiempo de animación de revelación y el tiempo de espera de inicio flotante en milisegundos.

1.2.2 Grupos de elementos

Se puede utilizar para definir diferentes acciones y configuraciones para un grupo de elementos. Los elementos se pueden agrupar utilizando el `data-type` atributo del elemento de la lista. Si un artículo no tiene ningún `data-type` atributo, se usará la configuración definida fuera de los grupos de artículos.

1.2.3 navegarOnDrop

Si la lista se puede ordenar y se establece el indicador multinivel, si un elemento se coloca en un elemento principal o en el botón Atrás, el elemento se agregará a la sublista/lista principal. La navegación en la caída es verdadera, después de que se suelta el elemento, la lista se navegará a la lista secundaria o la lista principal.

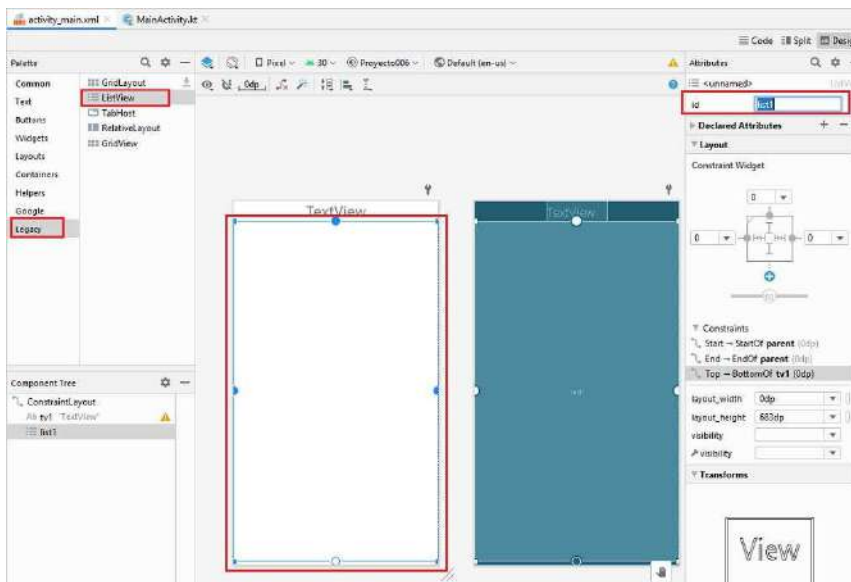


Imagen:

Fuente: <https://www.google.com/search?q=imagen+ejemplo>

1.2.4 ordenable

Si es verdadero, o un objeto, la lista se podrá ordenar. Al pasar un objeto, puede establecer opciones de clasificación adicionales:

handle: si es verdadero o 'derecho', aparecerá un controlador de clasificación en el lado derecho si está 'izquierdo' en el lado izquierdo de los elementos de la lista, lo que permite ordenar si el toque comenzó en el controlador (falso si no se especifica).

grupo: si es verdadero, se permite ordenar entre grupos (verdadero si no se especifica).

multinivel: si es verdadero, se permite ordenar entre niveles de la jerarquía (verdadero si no se especifica).

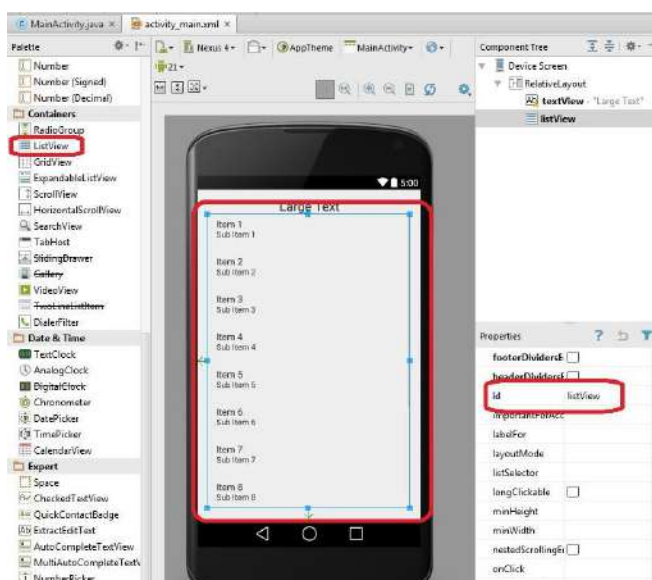


Imagen: Fuente:

<https://titiushko.github.io/Tutoriales-Ya/www.javaya.com.ar/androidva/androidstudiova/imagentema/foto021.jpg>

1.1 Control ImageButton

Por: Gustavo Rafael Juárez Mendoza

Un ImageButton es un AbsoluteLayout que permite especificar la ubicación exacta de sus elementos secundarios. Esto muestra un botón con una imagen (en lugar de texto) que el usuario puede presionar o hacer clic.

1.1.1 Atributos de ImageButton

A continuación, se muestran los atributos importantes relacionados con el control ImageButton. Puede consultar la documentación oficial de Android para obtener una lista completa de atributos y métodos relacionados que puede usar para cambiar estos atributos en tiempo de ejecución.

Fuente: [Android - ImageButton Control \(tutorialspoint.com\)](https://www.tutorialspoint.com/android/android_imagebutton_control.htm)

Por: María Emilia Larios Contreras

1.1.1 Notificaciones sencillas en clase Toast.

Las notificaciones Toast son una forma sencilla de mostrar mensajes breves en la pantalla de un dispositivo Android. Para crear una notificación Toast en una aplicación de Android, se usan estos pasos:

1. Importar la clase Toast en un archivo Java
2. Crear un objeto Toast con el mensaje que se desea mostrar
3. Llamar al método show() para mostrar la notificación Toast en la pantalla

Estos son los pasos básicos para mostrar una notificación Toast en una aplicación de Android. Se puede personalizar la notificación Toast para que se ajuste a diferentes necesidades, como cambiar la posición de la notificación o agregar un icono.



Fuente: [Notificaciones sencillas en clase Toast. - Bing images](#)

Esta sirve para interpretar cada una de sus usos

1.1.2 Control EditText

Un EditText es un elemento de interfaz de usuario (UI) en una aplicación de Android que permite al usuario ingresar y editar texto. Es un tipo de vista de entrada que se utiliza comúnmente en formularios y pantallas de ingreso de datos en aplicaciones móviles.

El EditText es un tipo especial de TextView que tiene la capacidad de ser editado por el usuario. Los usuarios pueden tocar el EditText para enfocarlo y luego ingresar texto a través del teclado virtual en la pantalla. El EditText también puede mostrar texto predeterminado, como instrucciones o sugerencias, para ayudar al usuario a ingresar la información correcta.

El EditText también puede ser personalizado para cumplir con las necesidades de diseño específicas de una aplicación. Por ejemplo, puedes cambiar el tamaño del texto, el color

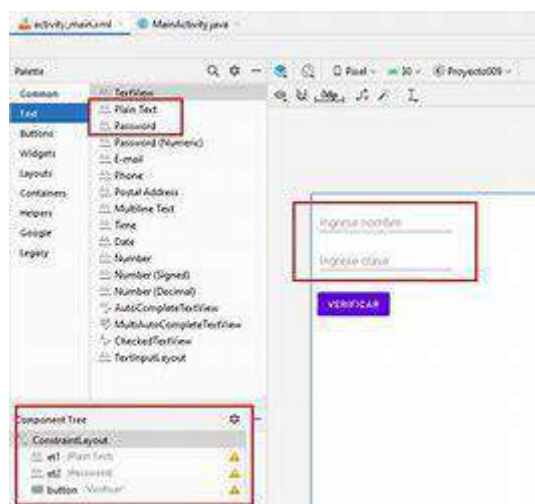
del fondo, la fuente o el estilo de la vista para que se ajuste a la apariencia general de la aplicación.

Para controlar un EditText en una aplicación de Android, se pueden utilizar varios métodos y escuchadores disponibles en la clase EditText. Algunos de los métodos y escuchadores más comunes son:

- `getText()`: Este método te permite obtener el texto actual que se encuentra en el EditText.
- `setText()`: Este método te permite establecer un texto en el EditText.
- `setKeyListener()`: Este método te permite establecer un escuchador de eventos de teclado en el EditText, que se activa cuando el usuario presiona una tecla.
- `setOnFocusChangeListener()`: Este método te permite establecer un escuchador de eventos de foco en el EditText, que se activa cuando el usuario enfoca o desenfoca el EditText.



- `addTextChangedListener()`: Este método te permite establecer un escuchador de eventos de texto en el EditText, que se activa cuando el texto en el EditText cambia.



Fuente: [Control EditText - Bing](#)

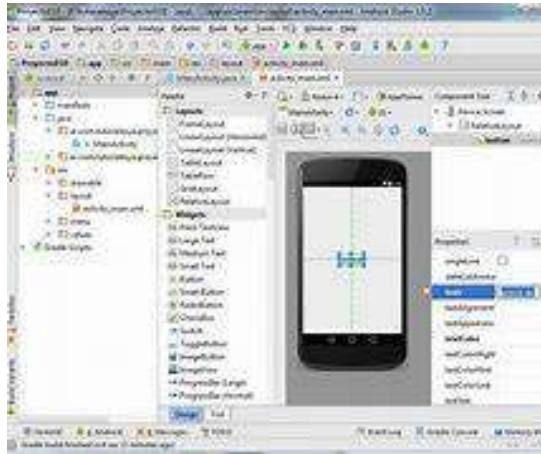
Sirve para la interpretación de cada nivel de usos adecuados.

1.1.3 Lanzar un segundo "Activity"

Para lanzar un segundo Activity en una aplicación de Android, se pueden seguir los siguientes pasos:

1. Crear una nueva clase Java para el segundo Activity y definir su layout en un archivo XML. Por ejemplo, se puede crear una clase llamada "SecondActivity.java" y su layout se puede definir en un archivo llamado "activity_second.xml".
2. En la clase Java del primer Activity, se crea un objeto Intent para lanzar el segundo Activity. El Intent describe el segundo Activity que se desea lanzar.

- Luego, se llama al método `startActivity()` y se le pasa el `Intent` como argumento para iniciar el segundo `Activity`.



Fuente: [Lanzar un segundo "Activity" - Bing images](#)

Esta imagen sirve para crear diseños adecuados a tu diseño.

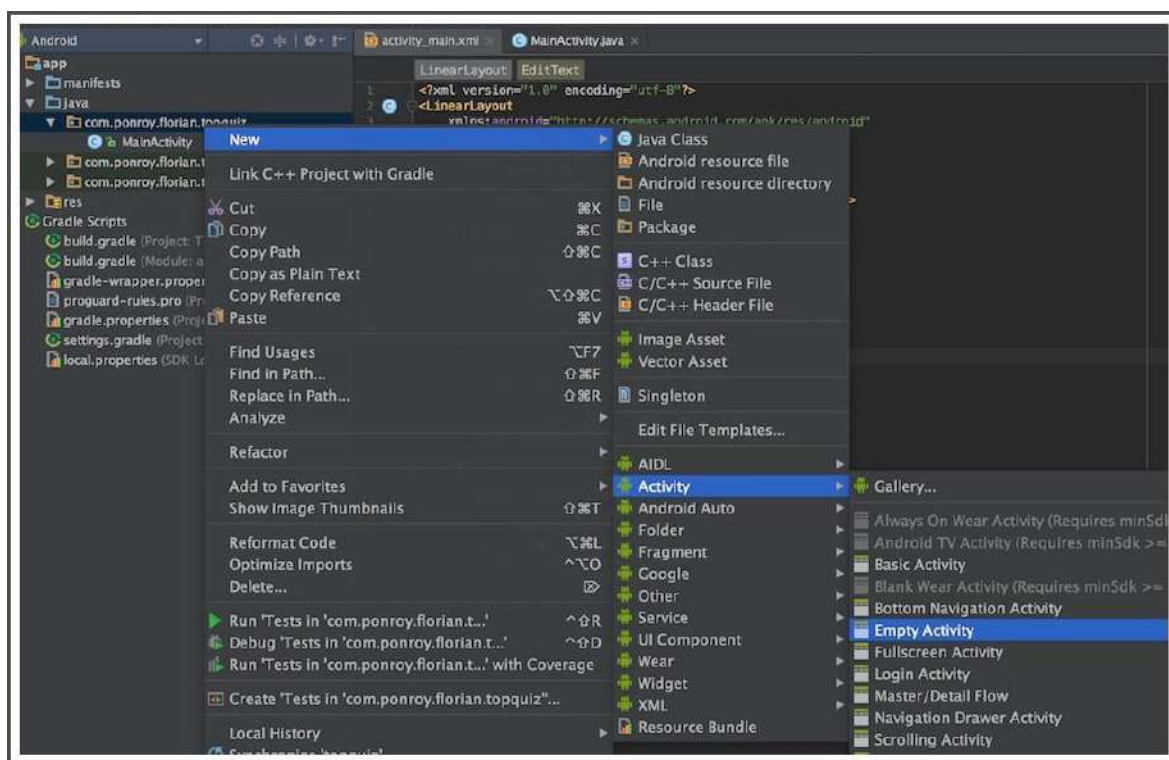
Sistema Android

Por: Glendy Lopez

1.1.1 Lanzar un segundo "Activity"

Para implementar esto en Android debemos plantear otros dos archivos uno xml con la interfaz y otro java con la lógica (tenemos en cuenta que cuando utilizamos Android Studio automáticamente cuando creamos un proyecto nos crea el archivo XML y el código java del primer Activity).

En la ventana Project, haz clic con el botón derecho en la carpeta de la app y selecciona New > Activity > Empty Activity.



Fuente https://user.oc-static.com/upload/2017/07/19/15004713822284_android-studio-create-new-activity.jpeg

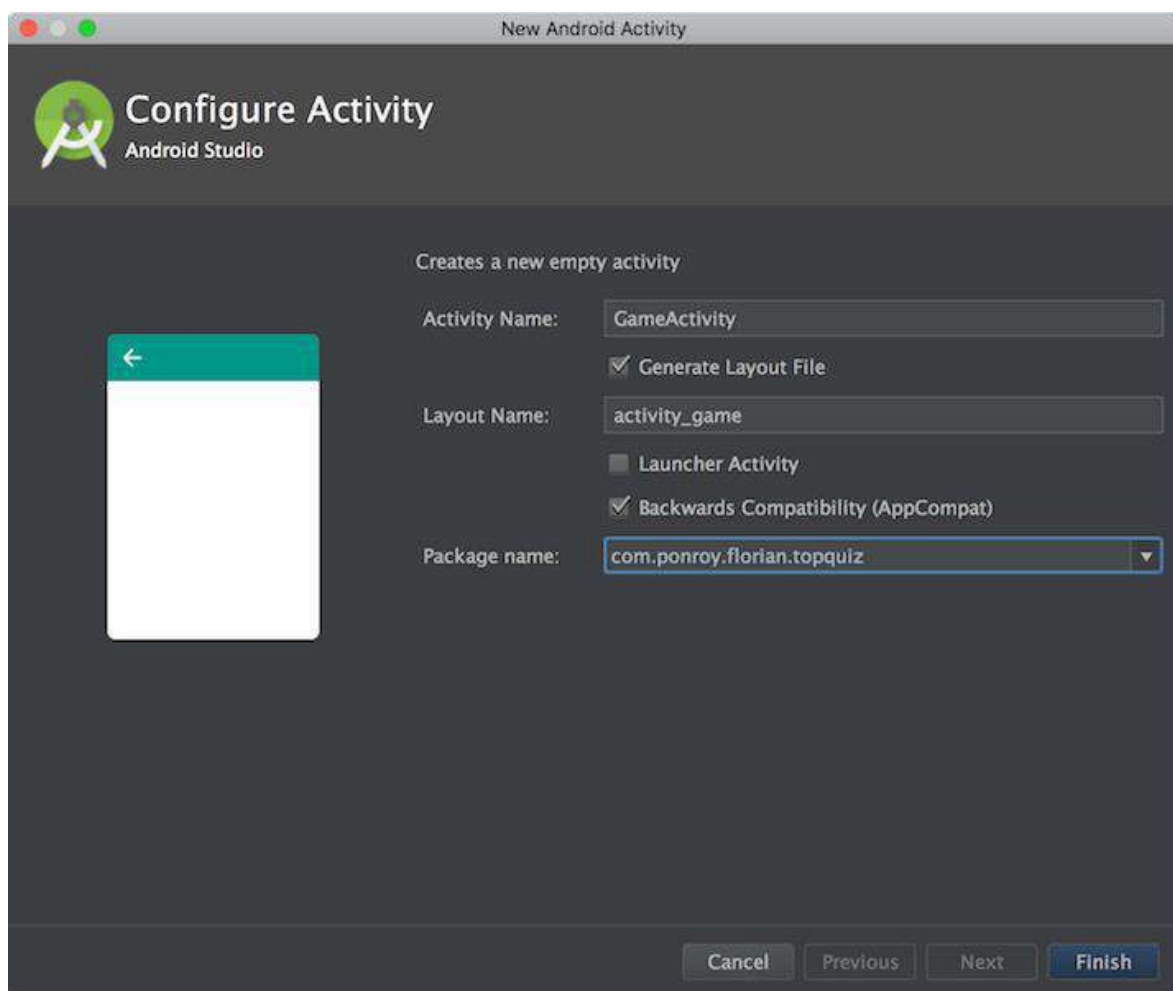
Android Studio muestra una nueva ventana. Esta ventana le permite configurar el nombre de la actividad y su archivo de diseño. Sabemos que esta Actividad manejará la lógica del juego, así que la llamaremos GameActivity . Deje las siguientes configuraciones:

Generar archivo de diseño está seleccionado

actividad_juego como el nombre de archivo generado

Se ha seleccionado Compatibilidad con versiones anteriores (App Compat)

Haga clic en el botón Finalizar.



Fuente https://user.oc-static.com/upload/2017/07/19/15004716158957_android-studio-configure-new-activity.png

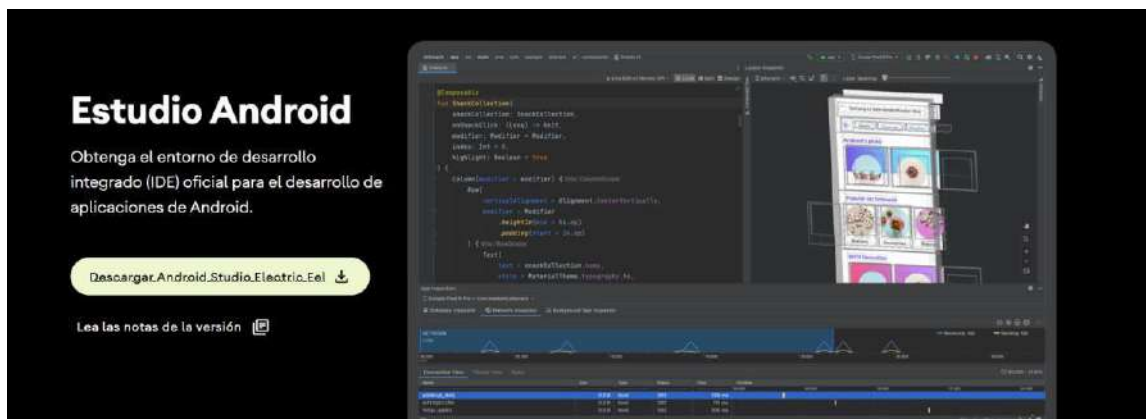
Android Studio realiza tres acciones automáticamente:

- Crea el archivo `DisplayMessageActivity`.
- Crea el archivo de diseño `activity_display_message.xml`, que se corresponde con el archivo `DisplayMessageActivity`.
- Agrega el elemento `<activity>` obligatorio en `AndroidManifest.xml`.
- Si ejecutas la app y presionas el botón en la primera actividad, se iniciará la segunda. Sin embargo, estará vacía porque usa el diseño vacío proporcionado por la plantilla.

1.1.2 Instalación Android en un dispositivo.

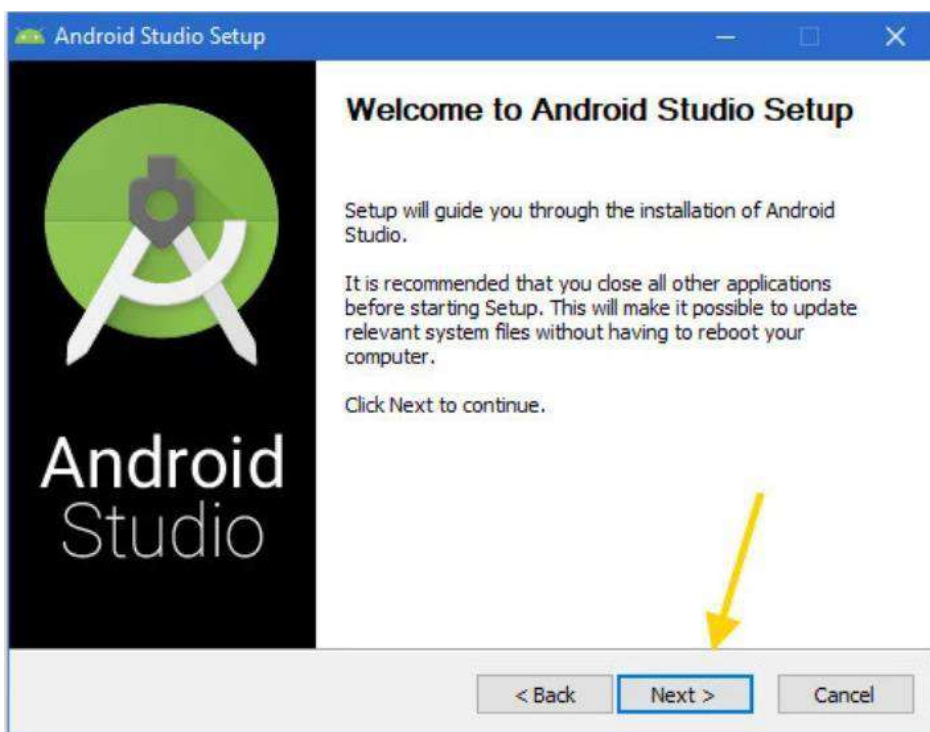
Para instalar Android Studio en Windows, sigue estos pasos:

1. Se descarga el programa en la página oficial.



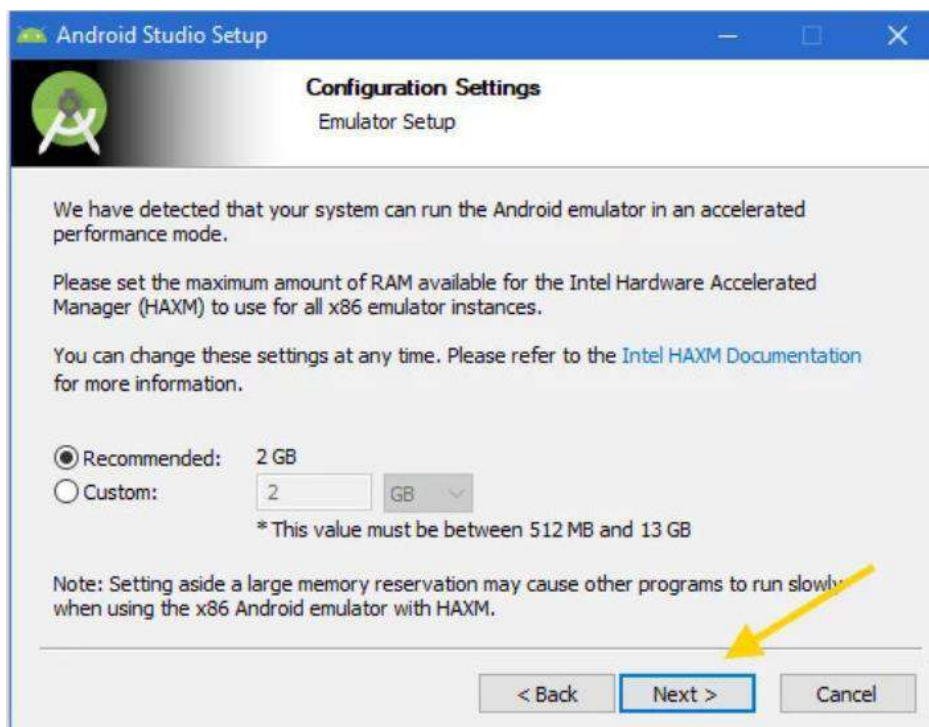
Fuente https://user.oc-static.com/upload/2017/07/19/15004713822284_android-studio-create-new-activity.jpeg

2. Descomprime el .zip.
3. Inicia la instalación



Fuente https://i.blogs.es/26b368/androidstudio11/1366_2000.webp

4. Se configura sobre la memoria RAM máxima que se dedicará al emulador de Android. Lo cierto es que puedes cambiar esto más tarde y la configuración recomendada, así que en esta ventana pulsar de nuevo Next.Century.



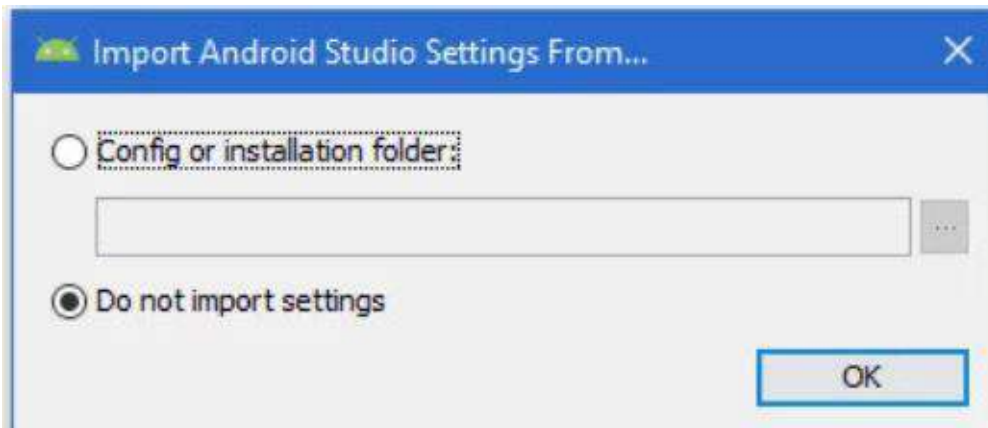
Fuente https://i.blogs.es/26b368/androidstudio11/1366_2000.webp

5. En cuanto termine, pulsa Finish para que se abra Android Studio.



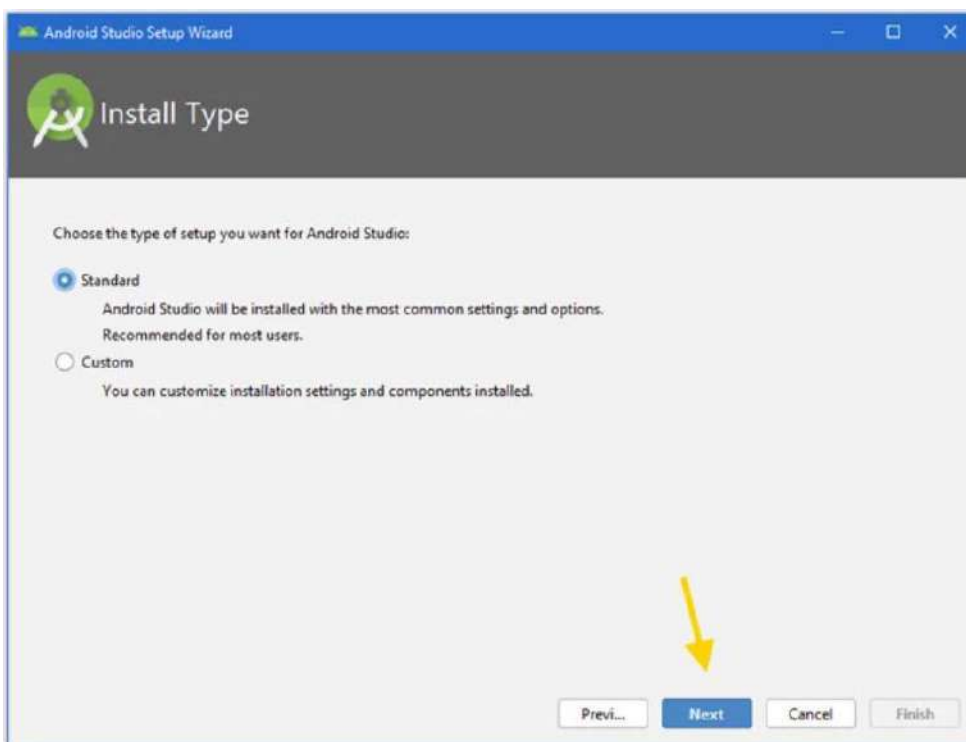
Fuente https://i.blogs.es/26b368/androidstudio11/1366_2000.webp

6. Abrir Android Studio. En ella simplemente se te pregunta si quieres importar la configuración de una versión anterior. Generalmente Android Studio detectará la configuración por sí mismo, y si no tienes ninguna configuración anterior guardada se marcará Do not import settings, o no importar configuración.



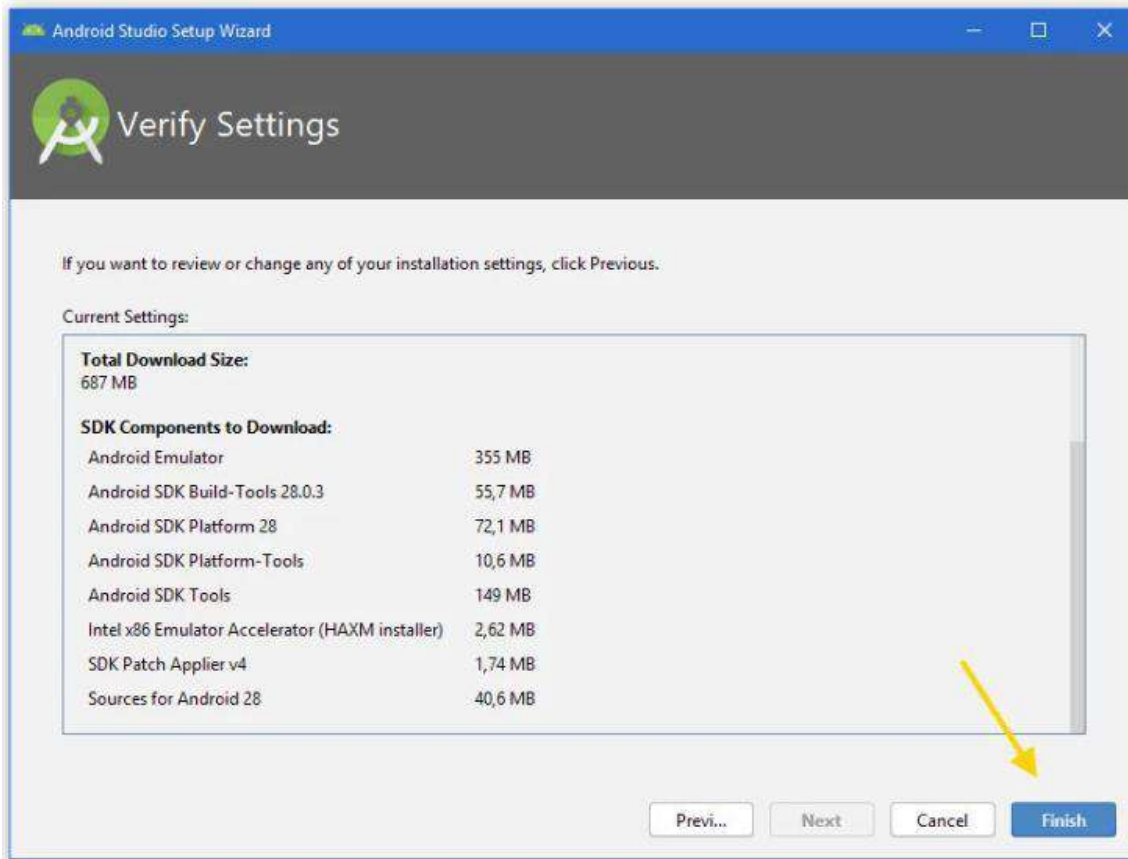
Fuente https://i.blogs.es/26b368/androidstudio11/1366_2000.webp

- 7. Lo primero que se te pregunta es si quieres hacer una instalación estándar o personalizada de Android Studio.



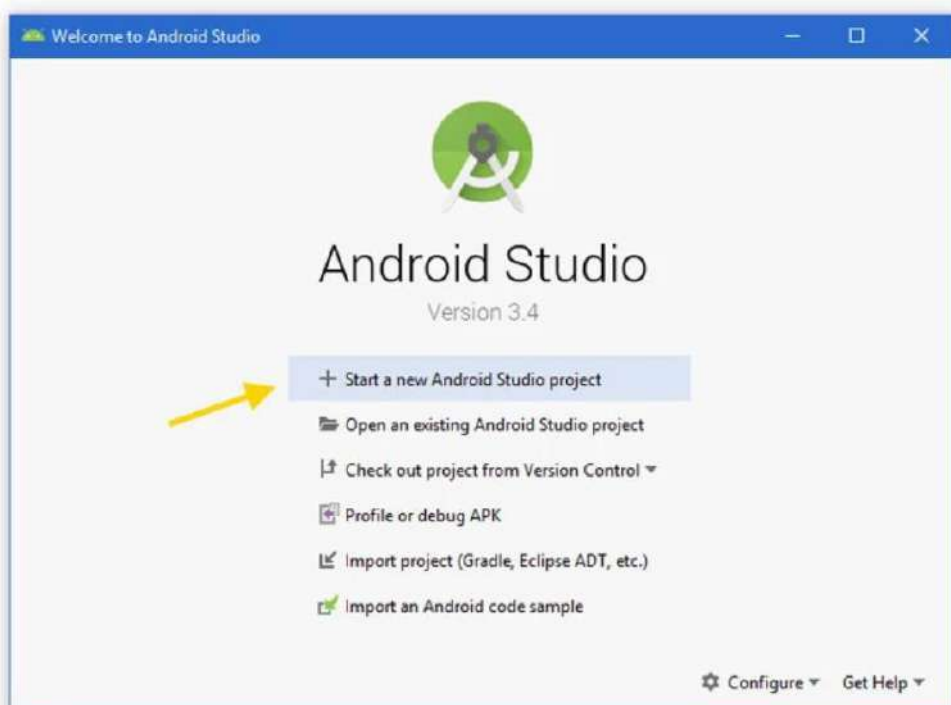
Fuente https://i.blogs.es/26b368/androidstudio11/1366_2000.webp

- 8. Descarga los componentes



Fuente https://i.blogs.es/26b368/androidstudio11/1366_2000.webp

- Una vez más la descarga seguramente supere los 500 MB, así que asegúrate de estar conectado a una red Wi-Fi y de tener suficiente espacio en el disco. Deja que el PC haga el trabajo por ti y espera a que termine la descarga e instalación. Cuando termine, pulsa Finish.



Fuente https://i.blogs.es/26b368/androidstudio11/1366_2000.webp

- Android Studio ya está instalado en tu sistema y te mostrará la pantalla de bienvenida.

1.1.6 Linear Layout

Es un ViewGroup que alinea a todos los niños en una sola dirección, vertical u horizontalmente

Todos los elementos secundarios de a LinearLayout se apilan uno tras otro, por lo que una lista vertical solo tendrá un elemento secundario por fila, sin importar qué tan anchos sean, y una lista horizontal solo tendrá una fila de altura (la altura del elemento secundario más alto, más el relleno).

Peso del diseño

Linear Layout también admite la asignación de un peso a niños individuales con el android:layout_weight atributo. Este atributo asigna un valor de "importancia" a una vista en términos de cuánto espacio debe ocupar en la pantalla. Un valor de peso mayor le permite expandirse para llenar cualquier espacio restante en la vista principal.

Distribución equitativa

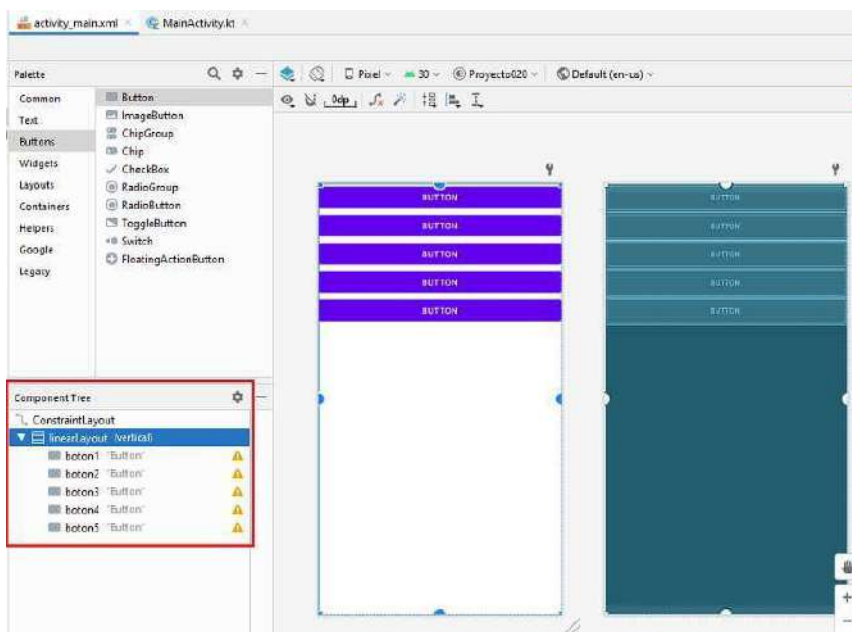
Para crear un diseño lineal en el que cada niño utilice la misma cantidad de espacio en la pantalla, establezca el android:layout_height de cada vista en "0dp" (para un diseño vertical) o el android:layout_width de cada vista en "0dp" (para un diseño horizontal). A continuación, establezca el valor android:layout_weight de cada vista en "1".

Distribución desigual

También puede crear diseños lineales donde los elementos secundarios usen diferentes cantidades de espacio en la pantalla:

Si hay tres campos de texto y dos de ellos declaran un peso de 1, mientras que el otro no tiene peso, el tercer campo de texto sin peso no crece. En cambio, este tercer campo de texto ocupa solo el área requerida por su contenido. Los otros dos campos de texto, por otro lado, se expanden por igual para llenar el espacio restante después de medir los tres campos.

Si hay tres campos de texto y dos de ellos declaran un peso de 1, mientras que al tercer campo se le asigna un peso de 2 (en lugar de 0), ahora se declara más importante que los otros, por lo que obtiene la mitad del total. espacio restante, mientras que los dos primeros comparten el resto a partes iguales.



Fuente <https://www.tutorialesprogramacionya.com/kotlinparaandroidya/imagentema/foto060b.jpg>

1.1.7 Linear Layout

TableLayout posiciona sus elementos secundarios en filas y columnas. Los contenedores de TableLayout no muestran líneas de bordes para las filas, columnas o celdas. La tabla tendrá la misma cantidad de columnas que la fila con la mayor cantidad de celdas. Una tabla puede dejar celdas vacías. Las celdas pueden abarcar varias columnas, igual que en HTML. Puedes abarcar columnas si usas el campo `span` de la clase `TableRow.LayoutParams`.

Los objetos `TableRow` son las vistas secundarias de `TableLayout` (cada objeto `TableRow` define una sola fila de la tabla). Cada fila tiene cero o más celdas, y cada una se define por cualquier tipo de objeto `View`.

Atributos de TableLayout

Androide: identificación

Este es el ID que identifica de forma única el diseño.

Android: colapsar columnas

Esto especifica el índice de base cero de las columnas que se contraerán. Los índices de las columnas deben estar separados por una coma: 1, 2, 5.

Android: reducir las columnas

El índice de base cero de las columnas que se van a reducir. Los índices de las columnas deben estar separados por una coma: 1, 2, 5.

Android: estirar columnas

El índice de base cero de las columnas que se van a estirar. Los índices de las columnas deben estar separados por una coma: 1, 2, 5.

Link

De las páginas

<https://developer.android.com/training/basics/firstapp/starting-activity?hl=es-419#:~:text=informaci%C3%B3n%2C%20consulta%20Navegaci%C3%B3n.-,C%C3%B3mo%20crear%20la%20segunda%20actividad,%22DisplayMessageActivity%22%20en%20Activity%20Name.>

<https://developer.android.com/studio/install?hl=es-419>

<https://www.xatakandroid.com/tutoriales/como-instalar-android-studio-tu-pc-cinco-sencillos-pasos>

<https://openclassrooms.com/en/courses/4661936-develop-your-first-android-application/4679051-create-a-second-activity#:~:text=Right-click%20on%20it%20and,name%20and%20its%20layout%20file.>

<https://tutorialesenlinea.es/75-como-instalar-el-sistema-operativo-android-en-el-movil.html>

<https://developer.android.com/develop/ui/views/layout/linear#:~:text=LinearLayout%20is%20a%20view%20group,buid%20your%20layout%20with%20ConstraintLayout.>

<https://developer.android.com/guide/topics/ui/layout/grid?hl=es-419>

https://www.tutorialspoint.com/android/android_table_layout.htm

10.1 Layout

Por: Didier Lopez

10.1.1 Layout (FrameLayout).

FrameLayout está diseñado para bloquear un área en la pantalla para mostrar un solo elemento. En general, FrameLayout se debe usar para contener una sola vista secundaria, ya que puede ser difícil organizar las vistas secundarias de una manera que sea escalable a diferentes tamaños de pantalla sin que los elementos secundarios se superpongan entre sí.

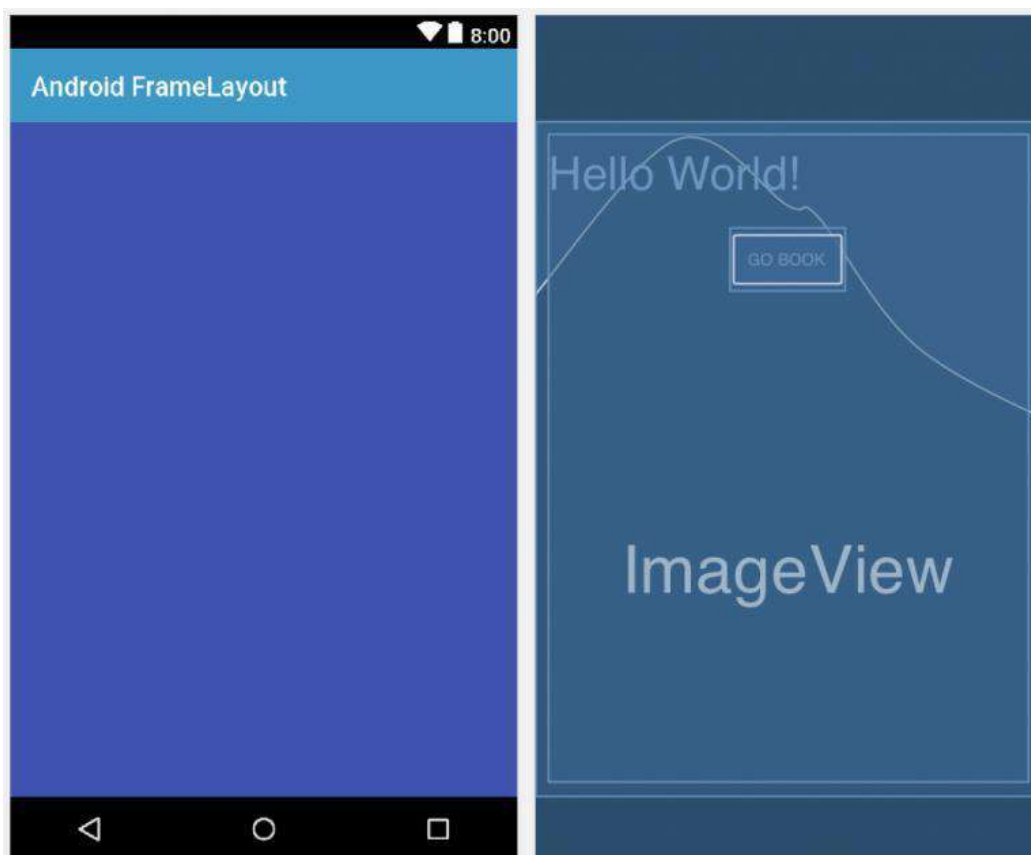


Imagen: Ejemplo de lo que es un FrameLayout.

Fuente: <https://tutorial.eyehunts.com/android/android-framelayout-tutorial-example-kotlin/>

10.1.2 ScrollView

Un grupo de vistas que permite desplazarse por la jerarquía de vistas colocada dentro de él. La vista de desplazamiento puede tener solo un hijo directo colocado dentro de ella.

ScrollView en Android también funciona como una especie de carpeta archivadora que se asegura de que el contenido en cada una de las vistas pensadas llegue de forma efectiva a los usuarios.

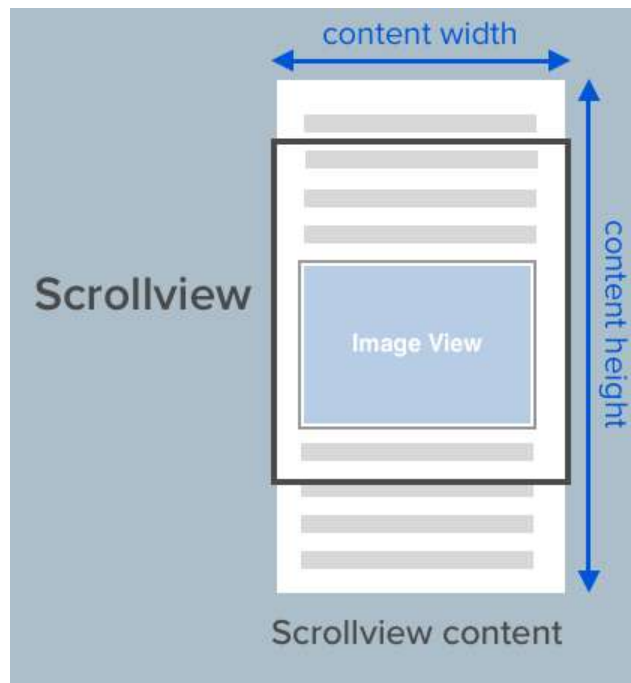


Imagen: Ejemplo de lo que es el ScrollView
 Fuente: <https://fluffy.es/scrollview-storyboard-xcode-11/>

10.1.3 LinearLayout.

LinearLayout es uno de los diseños más simples y más empleado. Simplemente establece los componentes visuales uno junto al otro, ya sea horizontal o verticalmente. Un LinearLayout respeta los márgenes entre los elementos secundarios y la gravedad (alineación a la derecha, centrada o a la izquierda) de cada elemento secundario.

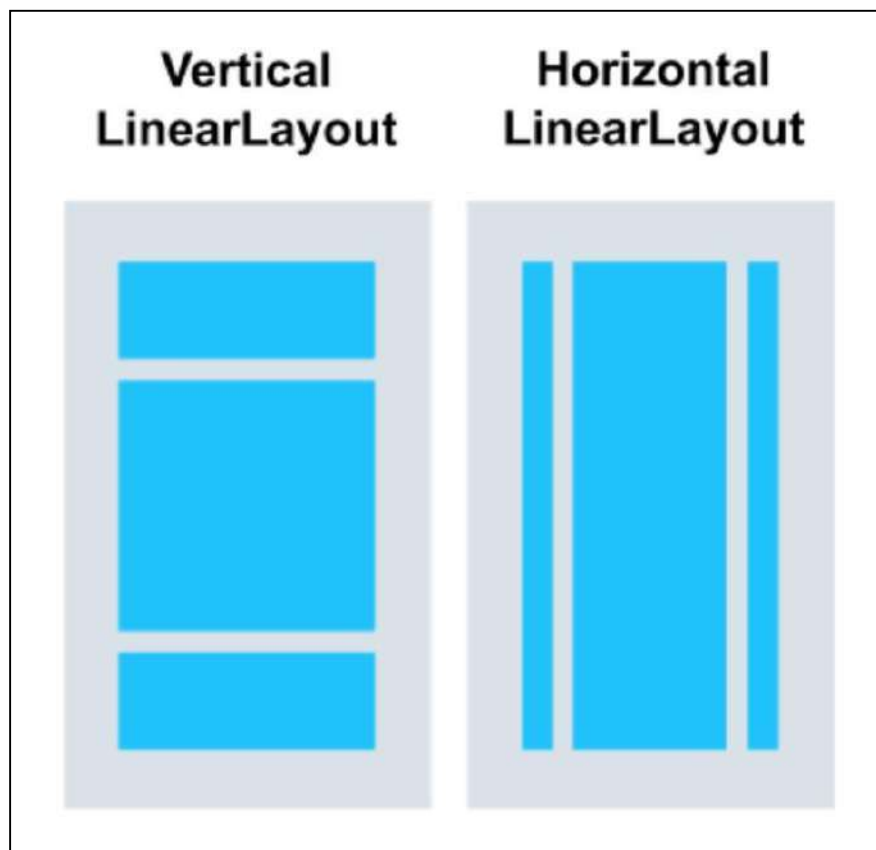


Imagen: LinearLayout en vertical y horizontal.

Fuente: <https://www.topcoder.com/thrive/articles/relativelayout-and-linearlayout-in-android>

1.1.3 Icono de aplicación

El ícono de la app es importante para diferenciarla, ya que le otorga un estilo y una apariencia distintivos. El ícono de la app aparece en diferentes lugares, por ejemplo, en la pantalla principal, en la pantalla Todas las apps y en la app de Configuración.

Para iniciar Image Asset Studio, sigue estos pasos:

1. En la ventana Project, selecciona la vista de Android.
2. Haz clic con el botón derecho en la carpeta res y selecciona New > Image Asset.

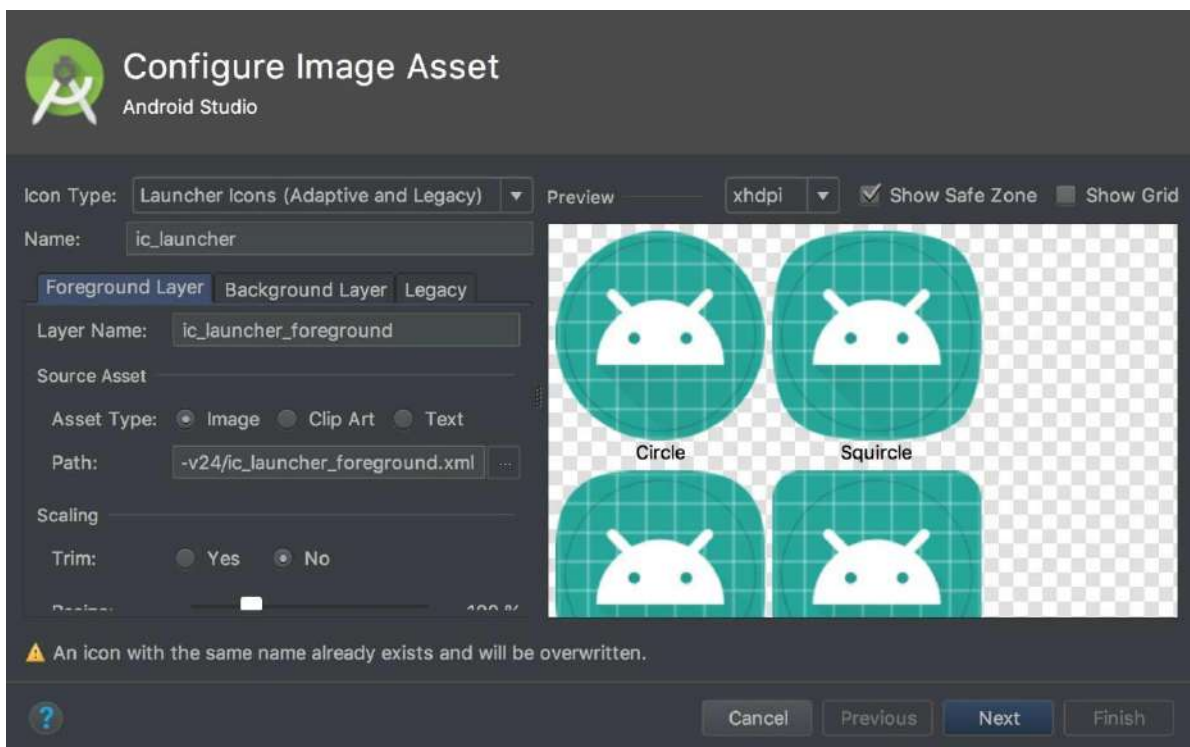


Imagen: Apartado de Image Asset para cambiar icono de la aplicación.

Fuente: <https://naps.com.mx/blog/icono-de-la-aplicacion-en-android/>

1.1 Reproduccion de Audio

Por: Byron Mansilla

La reproducción de audio en Android Studio se puede realizar utilizando la API de Android MediaPlayer. Esta API permite la reproducción de archivos de audio en diferentes formatos, como MP3, WAV, AAC, entre otros.

Para reproducir un archivo de audio con MediaPlayer, primero debes crear una instancia de esta clase utilizando el siguiente código:

```
java
```

```
MediaPlayer mediaPlayer = new MediaPlayer();
```

Imagen: Línea de código de Java utilizando la API de Android Studio MediaPlayer.

Fuente: Captura de Pantalla.

Luego, debes llamar al método `setDataSource()` para establecer la fuente de datos del archivo de audio que deseas reproducir. Por ejemplo:

```
java
```

```
mediaPlayer.setDataSource("ruta_del_archivo_de_audio");
```

Imagen: Línea de código para el llamado del método "setDataSource"

Fuente: Captura de Pantalla.

Después, debes llamar al método `prepare()` para preparar el reproductor para la reproducción del archivo de audio:

```
java
```

```
mediaPlayer.prepare();
```

Imagen: Línea de código para el llamado del método "prepare" para preparar el reproductor para la reproducción del archivo de audio.

Fuente: Captura de Pantalla.

Finalmente, puedes llamar al método `start()` para iniciar la reproducción del archivo de audio:

```
java
```

```
mediaPlayer.start();
```

Imagen: Línea de código para el llamado del método "start" para iniciar la reproducción del archivo de audio.

Fuente: Captura de Pantalla.

También es posible controlar la reproducción del archivo de audio utilizando otros métodos de la clase MediaPlayer, como `pause()` para pausar la reproducción, `stop()` para detener la reproducción, y `seekTo()` para buscar una posición específica en el archivo de audio.

Es importante tener en cuenta que, al finalizar la reproducción del archivo de audio, debes liberar los recursos utilizados por MediaPlayer llamando al método `release()`.

1.1.1 Pausa

Para pausar la reproducción de un archivo de audio en Android Studio utilizando la API de MediaPlayer, debes llamar al método `pause()` de la instancia de MediaPlayer. Este método detendrá temporalmente la reproducción del archivo de audio, manteniendo la posición actual en el archivo.

Por ejemplo, si deseas pausar la reproducción de un archivo de audio llamado "mi_audio.mp3", el código sería el siguiente:

```
java
mediaPlayer.pause();
```

Imagen: Línea de código para el llamado del método "pause" para pausar la reproducción del archivo.

Fuente: Captura de Pantalla

Es importante tener en cuenta que, después de llamar al método `pause()`, la instancia de MediaPlayer aún mantiene los recursos utilizados para la reproducción del archivo de audio, por lo que debes llamar al método `start()` nuevamente para reanudar la reproducción.

Para reanudar la reproducción después de una pausa, puedes llamar al método `start()` de la instancia de MediaPlayer. Por ejemplo:

```
java
mediaPlayer.start();
```

Imagen: Línea de código para el llamado del método "start" para reanudar la reproducción.

Fuente: Captura de Pantalla.

Hay que tomar en cuenta que si llamas al método `start()` después de una pausa, la reproducción del archivo de audio se reanudará desde la posición donde se detuvo al llamar al método `pause()`.

1.1.2 Continuación

Para continuar la reproducción de un archivo de audio desde la posición donde se detuvo la última vez, puedes utilizar el método `seekTo()` de la instancia de

MediaPlayer para establecer la posición deseada en el archivo de audio y luego llamar al método start() para continuar la reproducción.

Por ejemplo, si deseas continuar la reproducción de un archivo de audio llamado "mi_audio.mp3" desde la posición de tiempo de 10 segundos, puedes utilizar el siguiente código:

```
java
mediaPlayer.setDataSource("ruta_del_archivo_de_audio");
mediaPlayer.prepare();
mediaPlayer.seekTo(10000); // 10000 representa la posición
mediaPlayer.start();
```

Imagen: Línea de código para el llamado del método "seekTo".
Fuente: Captura de Pantalla.

1.1.3 Detención de un archivo de audio

La detención de la reproducción de un archivo de audio en Android Studio utilizando la API de MediaPlayer se realiza mediante el método stop() de la instancia de MediaPlayer. Este método detiene por completo la reproducción del archivo de audio y libera los recursos utilizados por MediaPlayer.

Por ejemplo, si se desea detener la reproducción de un archivo de audio llamado "mi_audio.mp3", el código sería el siguiente:

```
java
MediaPlayer.stop();
```

Imagen: Línea de código para el llamado del método "stop" para parar la reproducción de audio.
Fuente: Captura de Pantalla.

Después de llamar al método stop(), la instancia de MediaPlayer debe liberarse utilizando el método release(). Por ejemplo:

```
java
MediaPlayer.release();
```

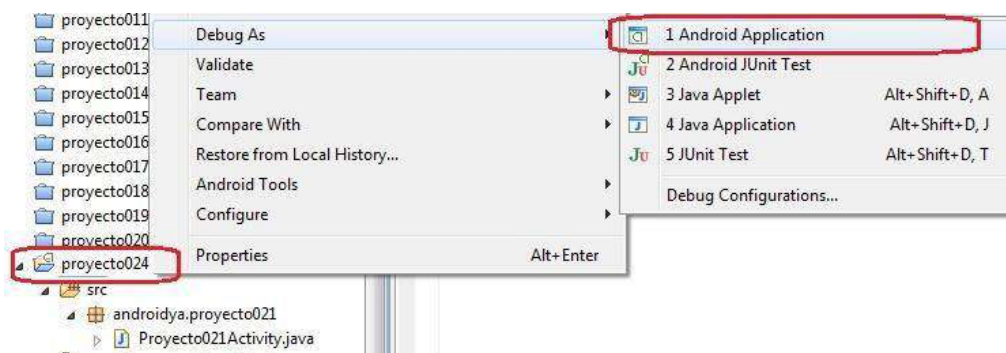
Imagen: Línea de código para el llamado del método "release"
Fuente: Captura de Pantalla.

Es importante tener en cuenta que, al liberar la instancia de MediaPlayer, se pierde el estado actual de reproducción y no se puede reanudar la reproducción desde la posición donde se detuvo. Si deseas continuar la reproducción desde la posición donde se detuvo, debes guardar esta posición en una variable antes de llamar al método stop(). Luego, puedes utilizar la posición guardada para establecer la posición de reproducción utilizando el método seekTo() cuando se reanuda la reproducción.

1. Reproducción de audio utilizando el reproductor propio de Android (v. Intent). Por: German Daniel Marquez Estrada

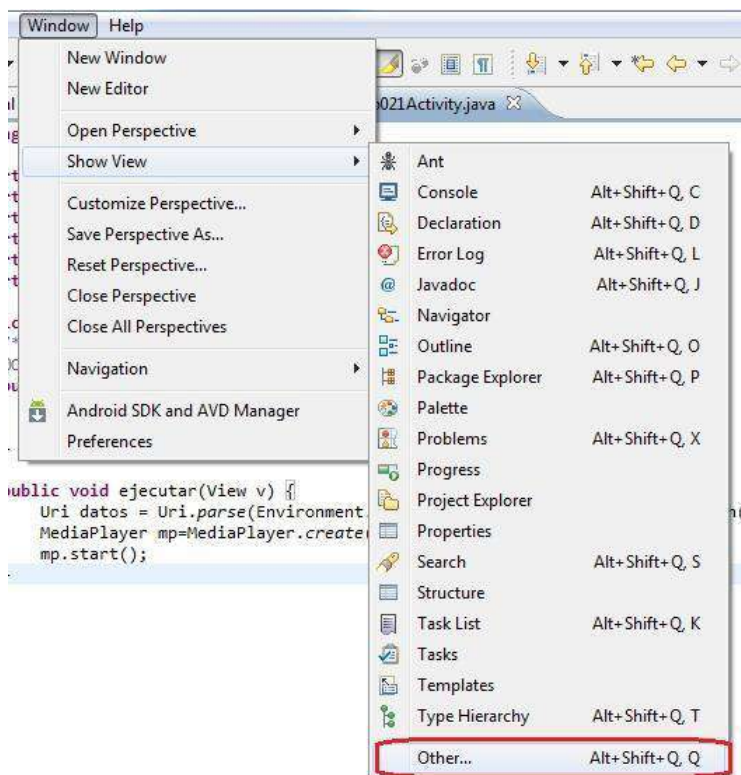
Disponer un botón con la etiqueta: "Ejecutar mp3 con el reproductor propio de Android", luego cuando se presione reproducir el archivo de audio respectivo con el reproductor de Android via Intent. El archivo de sonido almacenarlo en la tarjeta SD.

Luego de crear el proyecto debemos ejecutar la aplicación en modo depuración para poder acceder a la tarjeta SD que crea el emulador Android. Para ejecutar la aplicación en modo "Debug" presionamos el botón derecha del mouse sobre el proyecto y seleccionamos la opción "Debug A" "Android Application":



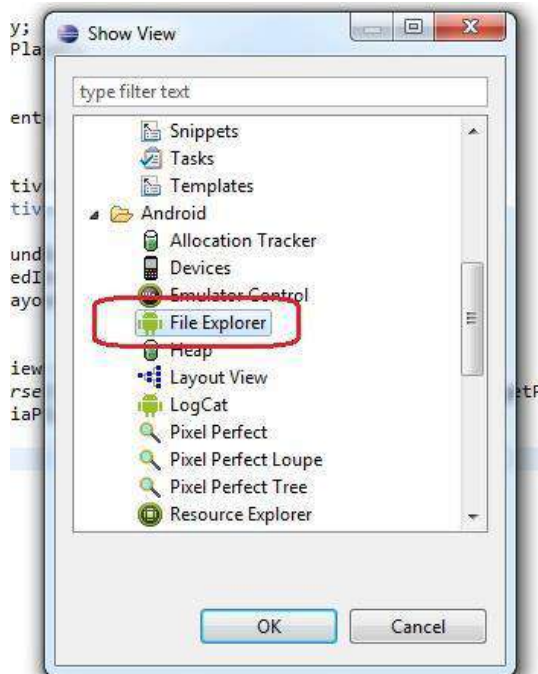
Fuente: <https://www.tutorialesprogramacionya.com/javaya/androidya/detalleconcepto.php?codigo=156&inicio=#:~:text=Disponer%20un%20bot%C3%B3n%20con%20la,almacenarlo%20en%20la%20tarjeta%20SD.>

Una vez que se esté ejecutando la aplicación (no importa que todavía no hemos implementado su funcionalidad) procedemos a abrir la ventana "File Explorer" de Android. Seleccionamos del menú de opciones de Eclipse: Window -> Show View -> Other:



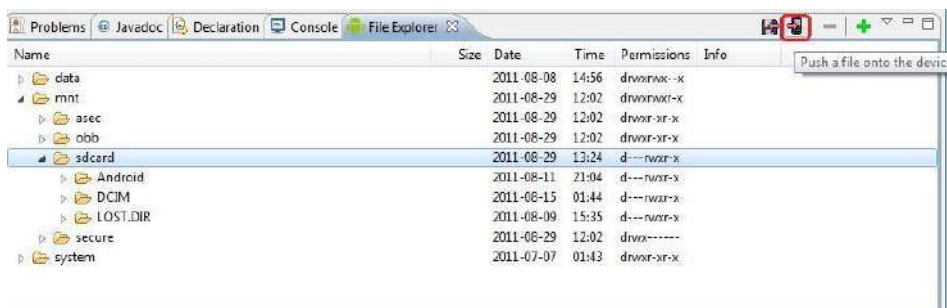
Fuente: <https://www.tutorialesprogramacionya.com/javaya/androidya/detalleconcepto.php?codigo=156&inicio=#:~:text=Disponer%20un%20bot%C3%B3n%20con%20la,almacenarlo%20en%20la%20tarjeta%20SD.>

Y en esta ventana seleccionamos Android -> FileExplorer:



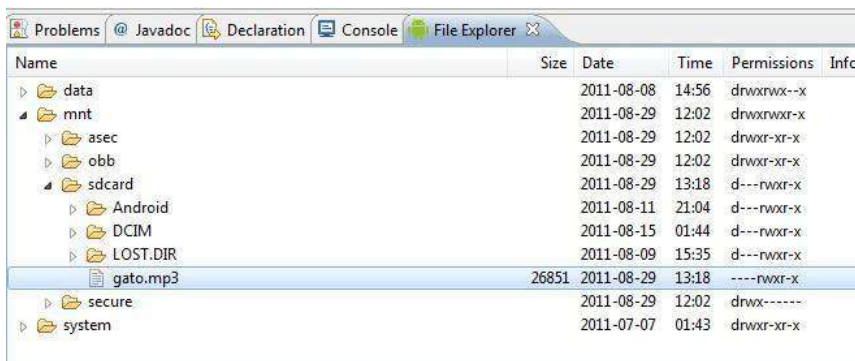
Fuente: <https://www.tutorialesprogramacionya.com/javaya/androidya/detalleconcepto.php?codigo=156&inicio=#:~:text=Disponer%20un%20bot%C3%B3n%20con%20la,almacenarlo%20en%20la%20tarjeta%20SD>.

Ahora podemos ver la ventana donde administramos los archivos contenidos en la tarjeta SD que emula el "Android Virtual Device":



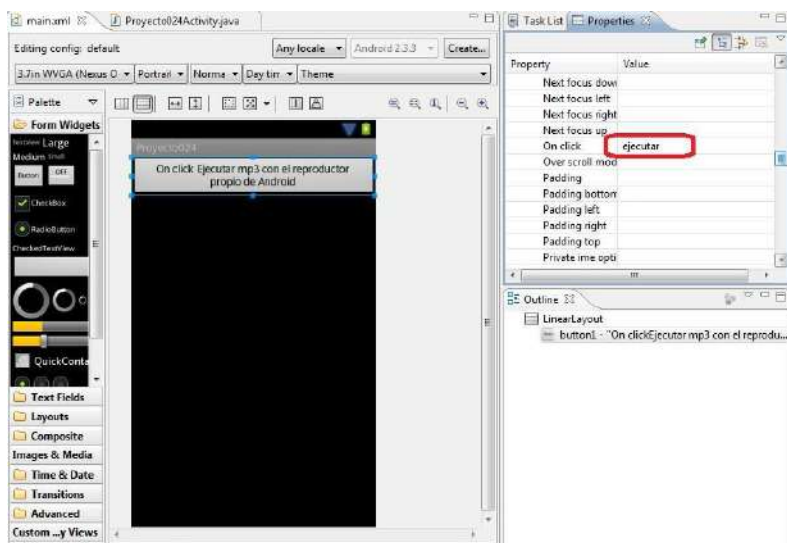
Fuente: <https://www.tutorialesprogramacionya.com/javaya/androidya/detalleconcepto.php?codigo=156&inicio=#:~:text=Disponer%20un%20bot%C3%B3n%20con%20la,almacenarlo%20en%20la%20tarjeta%20SD>.

En la carpeta sdcard debemos subir el archivo mp3 (debemos seleccionar esta carpeta con el mouse). Esto lo hacemos mediante un botón que aparece en la parte derecha de esta ventana "Push a file onto device", luego de esto tenemos el archivo montado en la tarjeta SD:



Fuente: <https://www.tutorialesprogramacionya.com/javaya/androidya/detalleconcepto.php?codigo=156&inicio=#:~:text=Disponer%20un%20bot%C3%B3n%20con%20la,almacenarlo%20en%20la%20tarjeta%20SD>.

Creamos la interfaz con el Button y especificamos el evento On click:



Fuente: <https://www.tutorialesprogramacionya.com/javaya/androidya/detalleconcepto.php?codigo=156&inicio=#:~:text=Disponer%20un%20bot%C3%B3n%20con%20la,almacendarlo%20en%20la%20tarjeta%20SD.>

El código fuente es:

```
package com.androidya.proyecto024;
```

importamos Librerias

```
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        // Inflate the menu; this adds items to the action bar if it is present.
```

```
        getMenuInflater().inflate(R.menu.activity_main, menu);
```

```

return true;
}

public void ejecutar(View v) {
    Intent intent = new Intent(android.content.Intent.ACTION_VIEW);
    Uri data = Uri.parse("file:///sdcard" + "/gato.mp3");
    intent.setDataAndType(data, "audio/mp3");
    startActivity(intent);
}
}
}

```

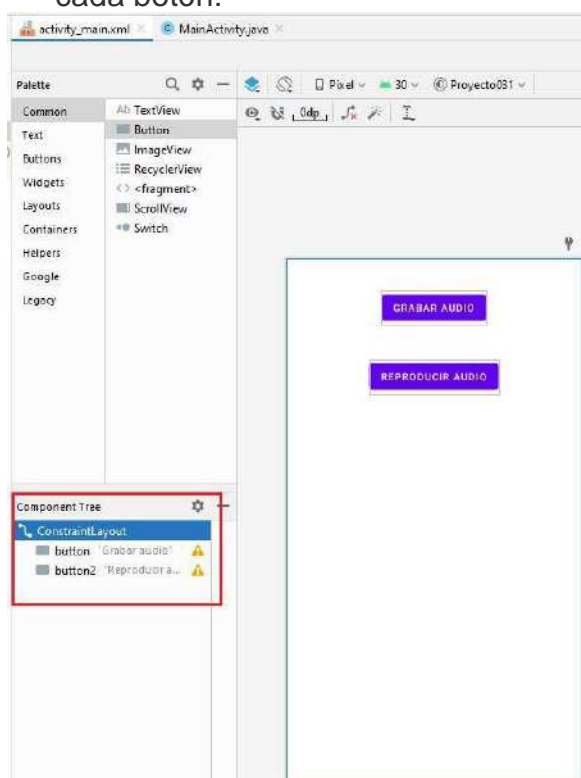
2. Grabación de audio mediante el grabador provisto por Android (via Intent).

1. La forma más sencilla de capturar audio en Android es mediante el grabador que provee el sistema operativo Android. Invocamos la aplicación de grabación y luego recuperamos el audio grabado.
2. Tiene como ventaja que la interfaz le es familiar al usuario, ya que muchas aplicaciones utilizan esta característica.

3. Problema:

Disponer dos objetos de la clase Button con las etiquetas "grabar" y "reproducir". Cuando se presione el primer botón proceder a activar la grabadora provista por Android. Cuando se presione el segundo botón reproducir el audio grabado.

Crear un proyecto (Proyecto031) e implementar la interfaz, inicializar los eventos onClick de cada botón:



4. El código fuente es:

```

package com.tutorialesprogramacionya.proyecto031;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    int petition = 1;
    Uri url1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void grabar(View v) {
        Intent intent = new Intent(MediaStore.Audio.Media.RECORD_SOUND_ACTION);
        startActivityForResult(intent, petition);
    }

    public void reproducir(View v) {
        MediaPlayer mediaPlayer = MediaPlayer.create(this, url1);
        mediaPlayer.start();
    }
}

```

@Override

```
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
    if (resultCode == RESULT_OK && requestCode == petición)
        url1 = data.getData();
}
```

}La forma más sencilla de capturar audio en Android es mediante el grabador que provee el sistema operativo Android. Invocamos la aplicación de grabación y luego recuperamos el audio grabado.

Tiene como ventaja que la interfaz le es familiar al usuario, ya que muchas aplicaciones utilizan esta característica.

Problema:

Disponer dos objetos de la clase Button con las etiquetas "grabar" y "reproducir". Cuando se presione el primer botón proceder a activar la grabadora provista por Android. Cuando se presione el segundo botón reproducir el audio grabado.

Crear un proyecto (Proyecto031) e implementar la interfaz, inicializar los eventos onClick de cada botón:

grabar audio android

El código fuente es:

```
package com.tutorialesprogramacionya.proyecto031;
```

```
import androidx.annotation.Nullable;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Intent;
```

```
import android.media.MediaPlayer;
```

```
import android.net.Uri;
```

```
import android.os.Bundle;
```

```
import android.provider.MediaStore;
```

```
import android.view.View;
```

```
public class MainActivity extends AppCompatActivity {
```

```
int peticion = 1;
```

```
Uri url1;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
}
```

```
public void grabar(View v) {
```

```
    Intent intent = new Intent(MediaStore.Audio.Media.RECORD_SOUND_ACTION);
```

```
    startActivityForResult(intent, peticion);
```

```
}
```

```
public void reproducir(View v) {
```

```
    MediaPlayer mediaPlayer = MediaPlayer.create(this, url1);
```

```
    mediaPlayer.start();
```

```
}
```

```
@Override
```

```
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
```

```
{
```

```
    if (resultCode == RESULT_OK && requestCode == peticion)
```

```
        url1 = data.getData();
```

```
    }
```

```
}
```

Cuando se presiona el botón de grabar el audio mediante un Intent activamos la aplicación de grabación propia de Android.

Seguidamente llamamos al método `startActivityForResult` para poder recuperar la grabación luego de finalizada a través del método `onActivityResult`:

```
public void grabar(View v) {
```

```
    Intent intent = new Intent(MediaStore.Audio.Media.RECORD_SOUND_ACTION);
```

```
    startActivityForResult(intent, peticion);
```

```
}
```

Debemos pasar al método `startActivityForResult` además de la referencia del Intent una variable con un valor 0 o positivo (luego este valor retornará al método `onActivityResult`)

Cuando finalizamos la grabación se ejecuta el método `onActivityResult`, donde almacenamos en la variable `url1` la referencia al archivo de audio creado:

```
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
    if (resultCode == RESULT_OK && requestCode == petición)
        url1 = data.getData();
}
```

Por último para ejecutar el contenido de la grabación utilizamos la clase ya vista `MediaPlayer`:

```
public void reproducir(View v) {
    MediaPlayer mediaPlayer = MediaPlayer.create(this, url1);
    mediaPlayer.start();
}
```

Cuando se presiona el botón de grabar el audio mediante un `Intent` activamos la aplicación de grabación propia de Android.

Seguidamente llamamos al método `startActivityForResult` para poder recuperar la grabación luego de finalizada a través del método `onActivityResult`:

```
public void grabar(View v) {
    Intent intent = new Intent(MediaStore.Audio.Media.RECORD_SOUND_ACTION);
    startActivityForResult(intent, petición);
}
```

Debemos pasar al método `startActivityForResult` además de la referencia del `Intent` una variable con un valor 0 o positivo (luego este valor retornará al método `onActivityResult`)

Cuando finalizamos la grabación se ejecuta el método `onActivityResult`, donde almacenamos en la variable `url1` la referencia al archivo de audio creado:

```
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
    if (resultCode == RESULT_OK && requestCode == petición)
        url1 = data.getData();
}
```

Por último para ejecutar el contenido de la grabación utilizamos la clase ya vista `MediaPlayer`:

```
public void reproducir(View v) {
    MediaPlayer mediaPlayer = MediaPlayer.create(this, url1);
    mediaPlayer.start();
}
```


1.1 Captura de audio mediante la clase MediaRecorder

Por: Elder César Oswaldo Martínez Rustrian

1.1.1 ¿para qué sirve?

Grabar flujos multimedia desde el dispositivo del usuario al instante usarlos en nuestra aplicación

1.1.2 ¿Como capturar audio?

1. Verificar si se tienen los permisos necesarios para grabar audio
Este código verifica si la aplicación tiene el permiso RECORD_AUDIO para grabar audio, y si no lo tiene, solicita permisos al usuario. En caso de que se tenga el permiso, se puede continuar con la grabación de audio.
2. Crear una instancia de la clase MediaRecorder y configurar sus parámetros
Es importante tener en cuenta que, para utilizar la clase MediaRecorder, es necesario manejar excepciones en el código. En el ejemplo anterior, se podrían agregar bloques try-catch para manejar excepciones de IO y de grabación de audio, como se muestra en los ejemplos anteriores

```
if (ContextCompat.checkSelfPermission(this, Manifest.permission.RECORD_AUDIO) !=
PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.RECORD_AUDIO}, 1);
    return;
}
```

Fuente: <https://developer.android.com/guide/topics/media/mediarecorder?hl=es-419>

```
recorder = new MediaRecorder();
recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
recorder.setOutputFile(getExternalCacheDir().getAbsolutePath() + "/audio.mp4");
```

Fuente: <https://developer.android.com/guide/topics/media/mediarecorder?hl=es-419>

Análisis de Sistemas

Quinto Grado

Instalacion de SQL Server 2014

Por: Adolfo Isaac Morales Delgado

Existen muchas ediciones de SQL Server 2014 (Enterprise Edition, Enterprise Core Edition, Business Intelligence Edition, Standard Edition, Web Edition, Express y Developer Edition). SQL Server 2014 incluye una edición gratuita: Express y aunque el servidor en sí carece dealgunas opciones que poseen otras ediciones, SQL Server 2014 Management Studio es el mismo y se puede usar con cada una de ellas.

1.1.1 Obteniendo el archivo de instalación

La edición completa de SQL Server Express se puede descargar desde este enlace: <https://www.microsoft.com/en-us/download/details.aspx?id=42299>, sin embargo, para poder descargar solo SQL Server 2014 Management Studio, al hacer clic en el botón Descargar se desplegará una lista de todos los archivos que contiene la instalación completa, pero solo necesitará uno de estas:

<input type="checkbox"/> File Name	Size
<input type="checkbox"/> ExpressAndTools_32BIT\SQLSERVERPRWT_x86_ENU.exe	840.8 MB
<input type="checkbox"/> ExpressAndTools_64BIT\SQLSERVERPRWT_x64_ENU.exe	833.2 MB
<input type="checkbox"/> LocalDB_32BIT\SqlLocalDB.msi	36.6 MB
<input type="checkbox"/> LocalDB_64BIT\SqlLocalDB.msi	43.1 MB
<input type="checkbox"/> MgmtStudio_32BIT\SQLManagementStudio_x86_ENU.exe	673.0 MB
<input type="checkbox"/> MgmtStudio_64BIT\SQLManagementStudio_x64_ENU.exe	683.9 MB

Download Summary:
You have not selected any file(s) to download.

Total Size: 0

Next

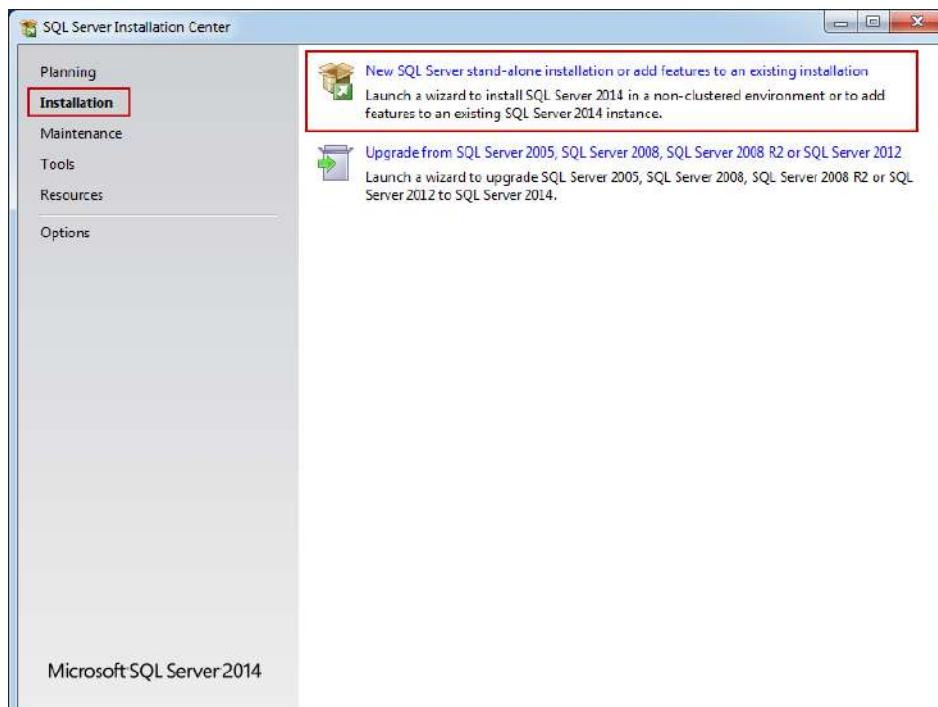
Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

1.1.2 Instalación

El archivo que fue descargado debe estar tener la extensión .exe con un tamaño de aproximadamente 650 MB. Haga doble clic en él para empezar a ejecutar la instalación:

Se muestra la ventana para elegir el directorio en el que se van a guardar los archivos extraídos. Después de elegir el directorio, tiene que hacer clic en el botón Aceptar y comenzará la extracción

Cuando se complete la extracción, el instalador tomará un par de segundos hasta que la configuración de SQL Server 2014 procese la operación actual. Después de eso, se abrirá la siguiente ventana:



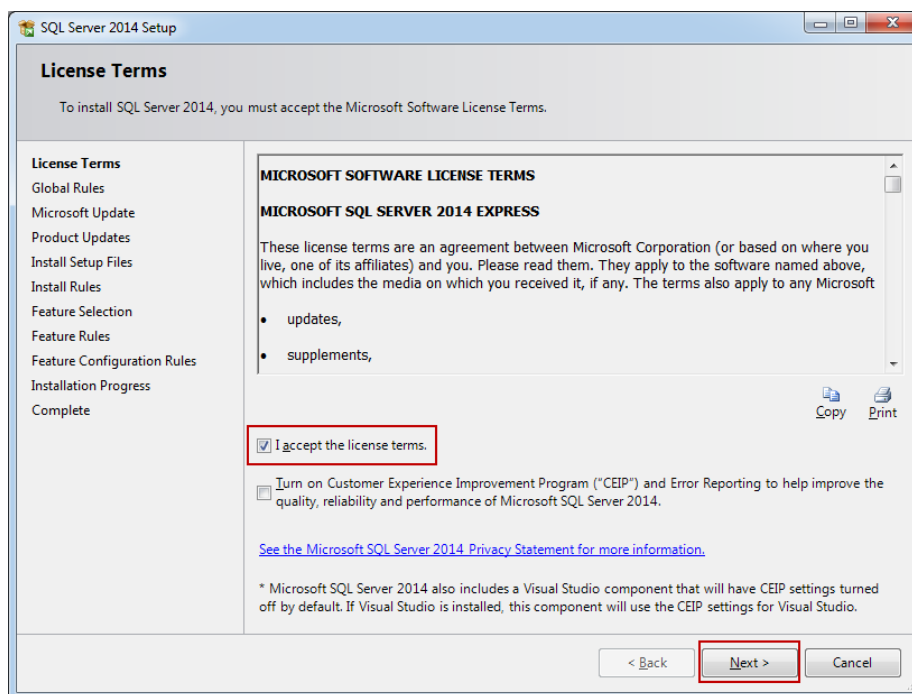
Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

Esta es la pantalla de instalación principal y la instalación de otras herramientas SQL también se pueden iniciar aquí. Si aún no está seleccionado, tiene que seleccionar la pestaña de Instalación en la esquina superior izquierda del Centro de instalación de SQL Server.

Para instalar solo SQL Server Management Studio, seleccione la nueva instalación independiente de SQL Server o agregue características a una opción de instalación existente (como en las capturas de pantalla anteriores).

Una vez iniciado el asistente de instalación, se puede cerrar el Centro de instalación de SQL Server.

Las reglas de configuración podrían demorarán un par de segundos (requisitos previos en el sistema operativo). Si es que se cumplen todos los requisitos, la configuración de SQL Server 2014 comenzará con la pestaña Términos de licencia como primer paso:



Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

Para poder continuar con la instalación, lea los términos de la licencia y luego marque la casilla Acepto los términos de la licencia. La casilla de verificación CEIP (Programa de mejora de la experiencia del cliente) es opcional, pero si se la marca, el programa podrá recopilar información sobre el hardware de la computadora y cómo se está utilizando el producto (sin interrumpir a los usuarios); Las próximas versiones de SQL Server se actualizan de acuerdo a esta información.

Después de marcar en la casilla de Acepto los términos de la licencia, se puede realizar un clic en el botón Siguiente para poder continuar.

El siguiente paso incluye las Reglas globales. Estas reglas globales de configuración identifican los problemas que pueden surgir al instalar SQL Server 2014. Las reglas que se aplican en cierta medida, dependen del sistema operativo que esté instalando, pero generalmente incluyen lo siguiente:

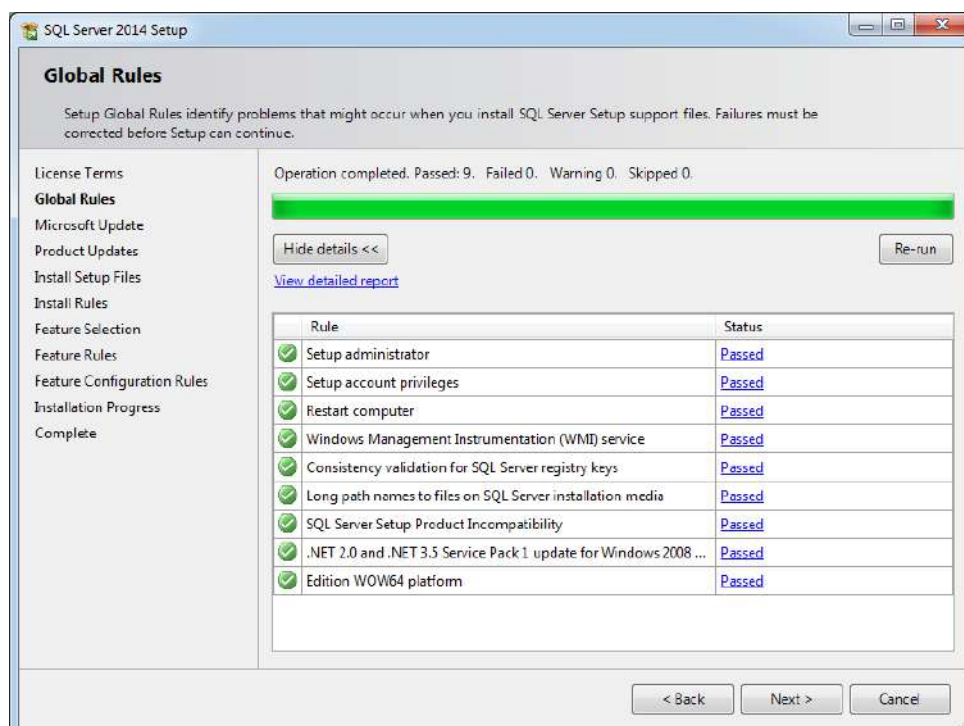
Si el usuario conectado es un administrador del sistema con los privilegios apropiados. (tiene que serlo)

Si hay reinicios pendientes de otros instaladores

Si los componentes necesarios de .NET se encuentran disponibles

Si hay soporte para nombres de ruta largos en donde reside el medio de instalación La consistencia de cualquier clave de registro de SQL Server

Si se pasan todas las reglas, el paso de Reglas globales se procesará y se omitirá automáticamente. Si no se llegan a aprobar algunas reglas, se las deberá corregir para que la configuración pueda continuar:



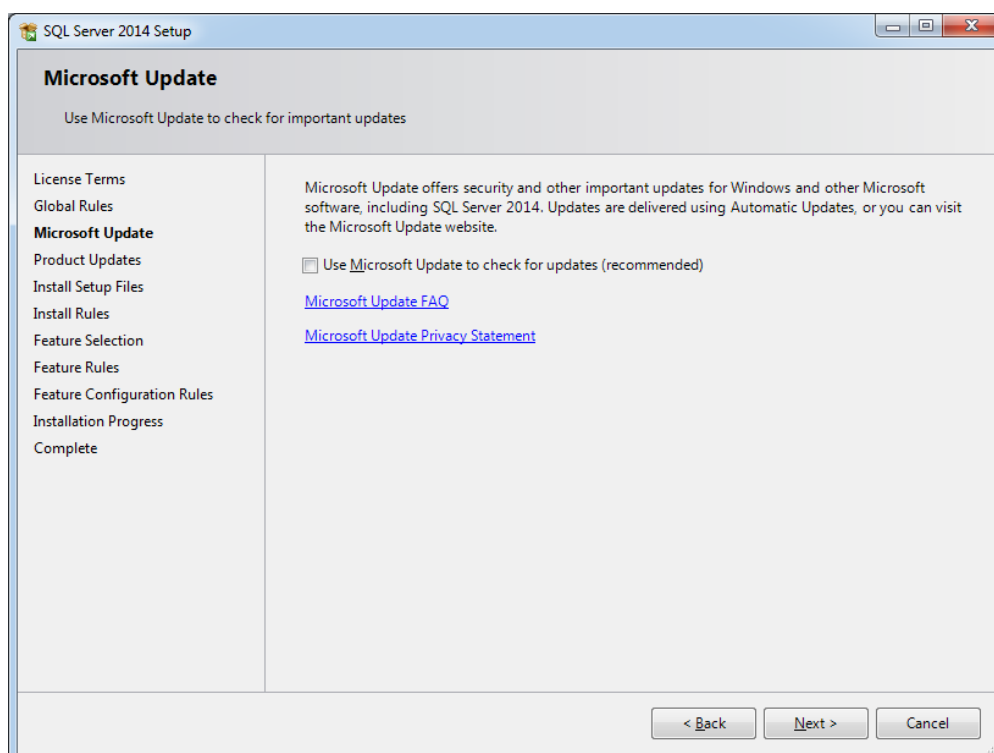
Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

Gran parte de los problemas conocidos se pueden resolver con soluciones del foro de Microsoft y soluciones alternativas.

El siguiente paso de la configuración es la Actualización de Microsoft. Es opcional y si lo selecciona o no, la instalación recién podrá continuar después de realizar un clic en el botón Siguiente.

Una vez marcada, esta opción nos brinda la oportunidad de descargar la actualización e incorporarla a la instalación. Esta acción llevará algún tiempo, ya que esas actualizaciones también deben ser descargadas.

Si continuó sin buscar actualizaciones o si está marcada, pero no hay actualizaciones disponibles, se va a omitir el paso Actualizaciones del producto:



Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

Se le recomienda que la opción de actualización de Microsoft esté marcada para garantizar que se apliquen todos los parches.

Los pasos de instalación de archivos de instalación y reglas de instalación se procesan automáticamente. Sin embargo, las Reglas de instalación son otro paso en el que el asistente de configuración verifica si se cumplen ciertos requisitos de software

Necesidad de reiniciar para que se apliquen los cambios

Si hay algunas versiones anteriores de SQL Server Validación

de claves de registro de SQL Server

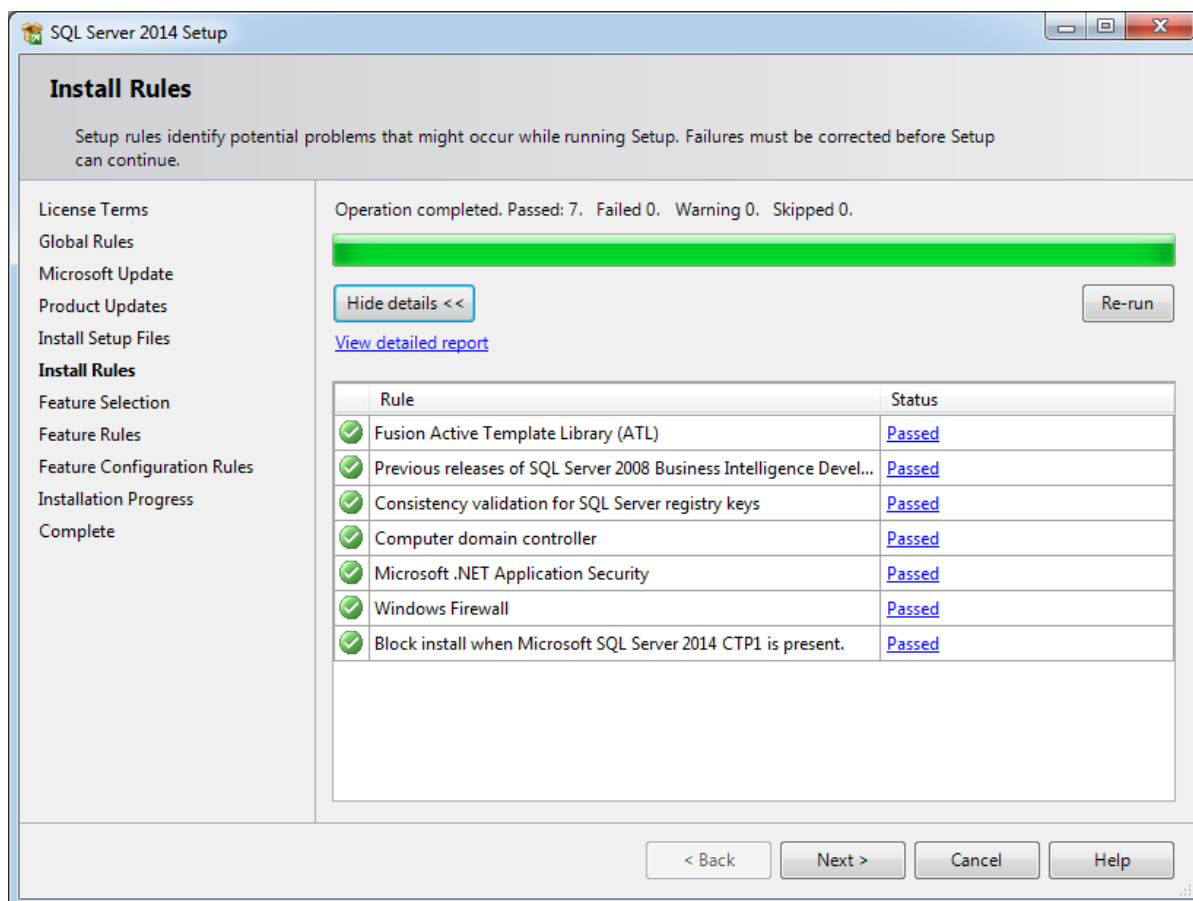
Si la computadora es un controlador de dominio

Configuraciones de seguridad

Cada regla de la lista tiene un estado (ya sea aprobado o no); Al hacer clic en esos enlaces se proporciona información adicional si algo salió mal.

Una regla que casi seguramente se arrojará, es una advertencia del Windows Firewall. Este problema específicamente se puede resolver agregando SQL Server y sus características a la lista de excepciones de Firewall de Windows. De lo contrario, SQL Server Management Studio no puede conectarse a servidores remotos.

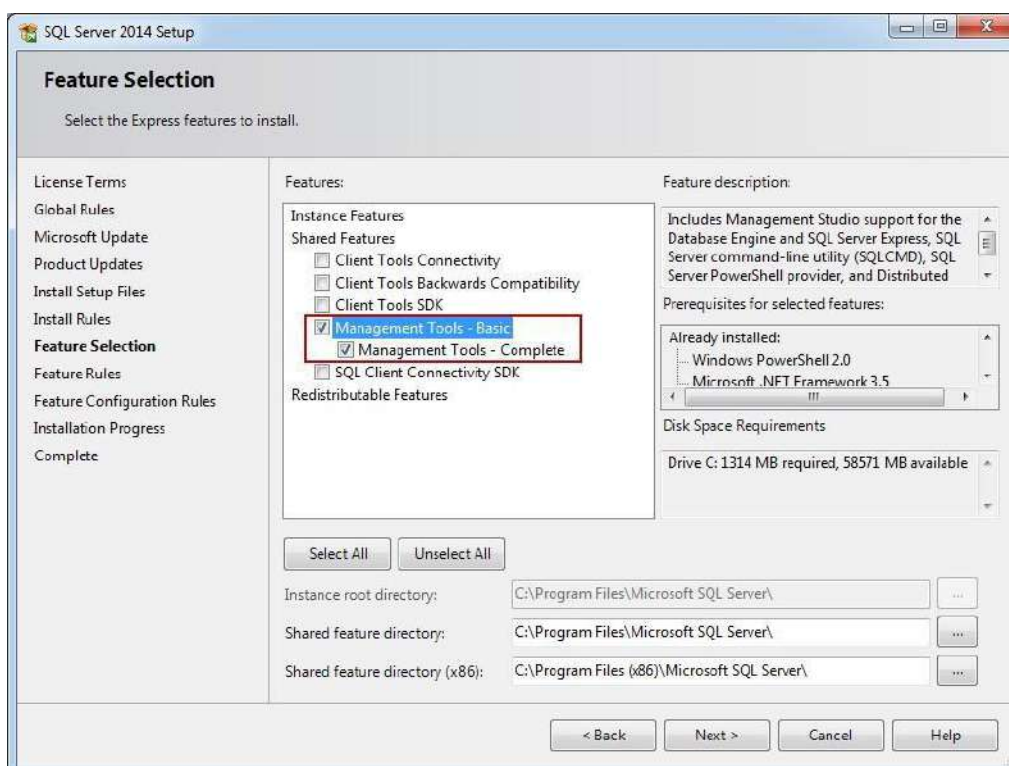
Si algunos de estos no pasan, deben corregirse para que la configuración pueda continuar:



Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

En el paso de Selección de funciones, las únicas opciones que deben verificarse para instalar SQL Server Management Studio son las herramientas de administración: elemento básico y secundario: Herramientas de administración: completa, sin embargo, el SDK de conectividad de cliente SQL se instalará de manera predeterminada, ya sea que usted lo comprobó o no.

Cada característica tiene unos requisitos previos que se muestran en el campo Requisitos previos para las características seleccionadas ubicadas a la derecha del campo Características, y si faltan algunas, se van a instalar junto con SQL Server Management Studio:



Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

Además, existe una breve descripción de cada función en el campo Descripción de la función para obtener más detalles.

SQL Server Management Studio se instalará en el directorio predeterminado en la partición del sistema, pero debajo del campo Características se puede modificar el directorio en el que se desea instalar estos componentes compartidos (como se muestra en la imagen de arriba)

Después de continuar con el botón Siguiente, los pasos Reglas de características y Reglas de configuración de características se procesarán de manera automática si se cumplen todas las Reglas de características:

Instancias anteriores de Visual Studio 2010 que necesitan de una actualización Se requiere Microsoft .NET Framework 3.5 Service Pack 1

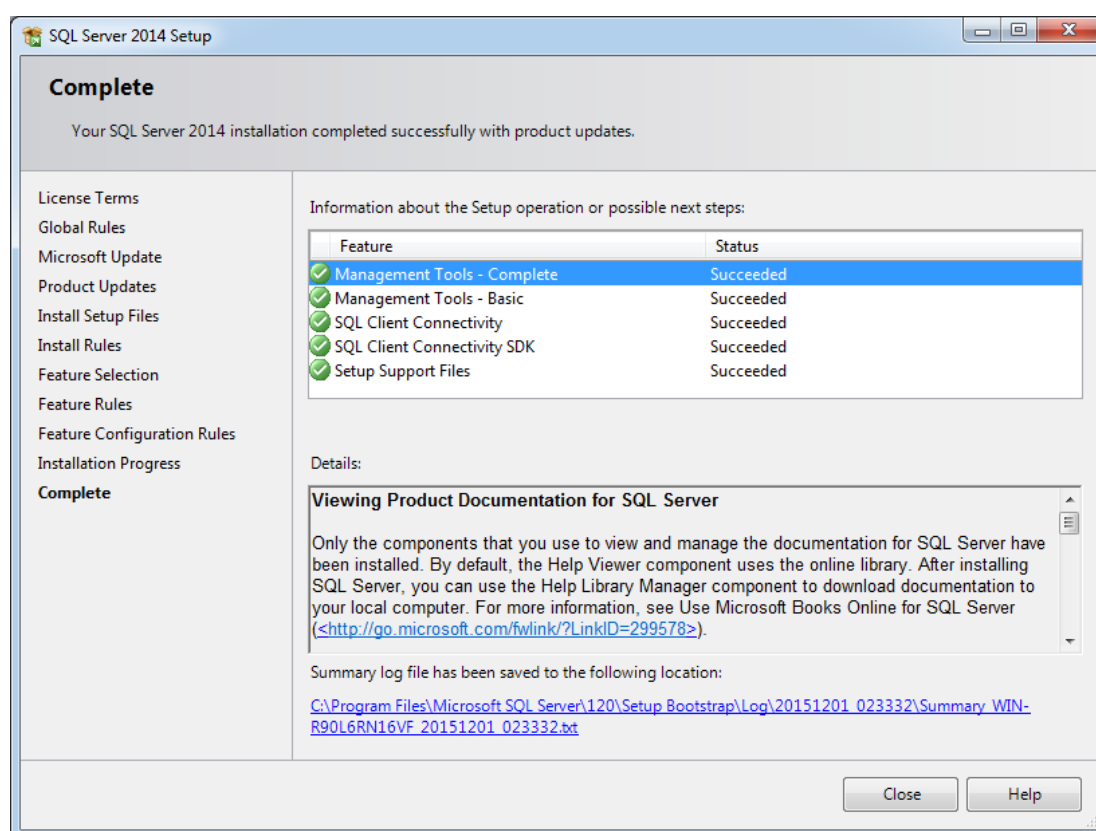
Se deben reparar las fallas para que la instalación pueda continuar.

La actualización que se requiere para Visual Studio está contenida en los instaladores de SQL Server y se llama VS10sp1-KB983509.msp.

El Microsoft .NET Framework 3.5 Service Pack 1 se puede encontrar en este enlace.

Cuando se logra pasar todas las reglas y los pasos de las Reglas de características y las Reglas de configuración de características se procesan automáticamente, el proceso de instalación comenzará en el paso Progreso de la instalación.

El proceso de instalación de SQL Server Management Studio no debería tomar más de 4 minutos aproximadamente, si no hay un servidor real instalado en la máquina. Después de que se complete el proceso de instalación, se presentará el último paso del asistente de instalación:



Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

En el paso

Completo

, se mostrará el resumen de todas las características instaladas y el estado de instalación de cada una.

En el campo a continuación, etiquetado como Detalles, habrá enlaces adicionales para Notas de la versión, Bases de datos de muestra (aplicables solo si tiene un servidor instalado en la máquina), Actualizaciones de Microsoft, etc.

Además, se presentará un enlace en el que puede encontrar el registro de resumen guardado como archivo txt.

1.1.3 Instalación de SQL Server 2014 Management Studio con SQL Server 2014 previamente instalado

Si ya está instalando SQL Server Management Studio en la máquina en la que hay un SQL Server instalado previamente, habrá diferencias menores y un paso adicional en el proceso de instalación:

El primer paso será las Reglas globales en lugar de los Términos de licencia

Después de que se acepten las Reglas de instalación, el paso adicional – Tipo de instalación – será el siguiente:

Como SQL Server Management Studio es un componente compartido y se instala como una característica de SQL Server, pero también puede funcionar como una aplicación independiente (no como una característica de un servidor específico), depende de usted el cómo lo instalará, porque funcionará de cualquier manera.

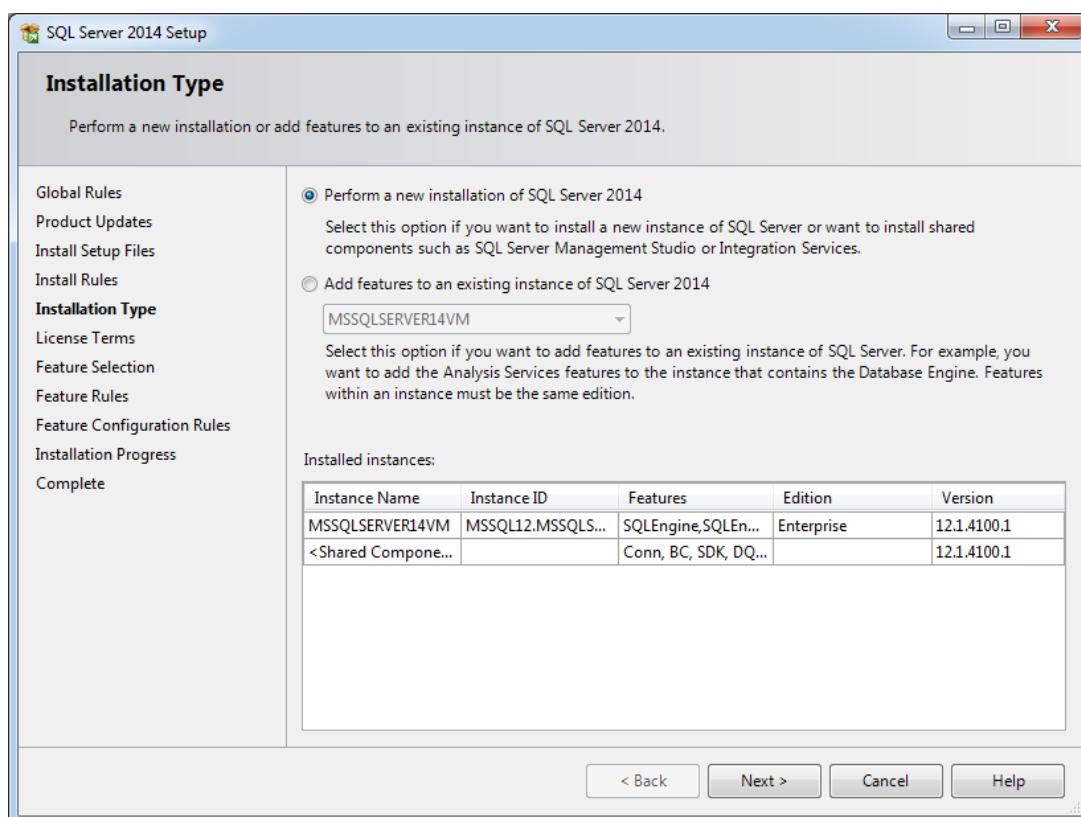
Solo tiene que tener en cuenta que, si selecciona Agregar características a una instancia existente de SQL Server 2014, debe definir la instancia a la que está agregando características en la lista desplegable debajo del segundo botón de opción

La diferencia final consiste en que, como el primer paso, los Términos de licencia seguirán el Tipo de instalación. Para poder continuar con la instalación, lea los términos de la licencia y luego marque la casilla Acepto los términos de la licencia

Para el resto de la instalación, continúe desde el paso número 7, ya que estos pasos son idénticos

Independientemente del tipo de instalación que se esté realizando, una vez ya finalizada la instalación, aparecerá el botón Cerrar, el cual cerrará el asistente de instalación y su SQL Server Management Studio se configurará para el primer uso.

Ahora que ha instalado SQL Server 2014 Management Studio, puede también actualizar este software con algunos complementos gratuitos de ApexSQL que agregarán mayores capacidades de formato, autocompletado, refactorización y búsqueda. Haga clic aquí para ver cómo poder instalarlos.



Fuente: <https://www.sqlshack.com/es/como-poder-instalar-sql-server-2014-management-studio/>

1.1.4 configuracion

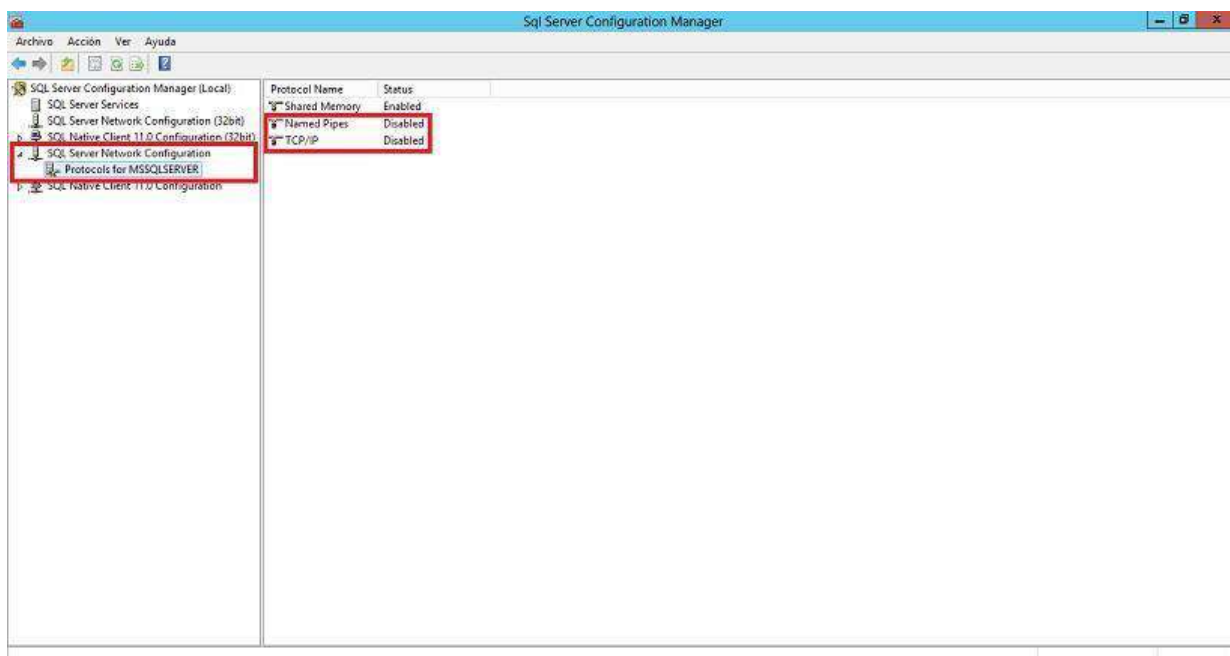
Para habilitar la conexión de Red en la base de datos de SQL debemos ingresar al programa SQL Server 2014 Configuration Manager, el cual se encuentra en el Menú de inicio de Windows dentro de la carpeta Microsoft SQL Server 2014\ Configuration Tools.



Fuente: <https://help.mproerp.com/help/0000000058/>

SQL Server Configuration Manager

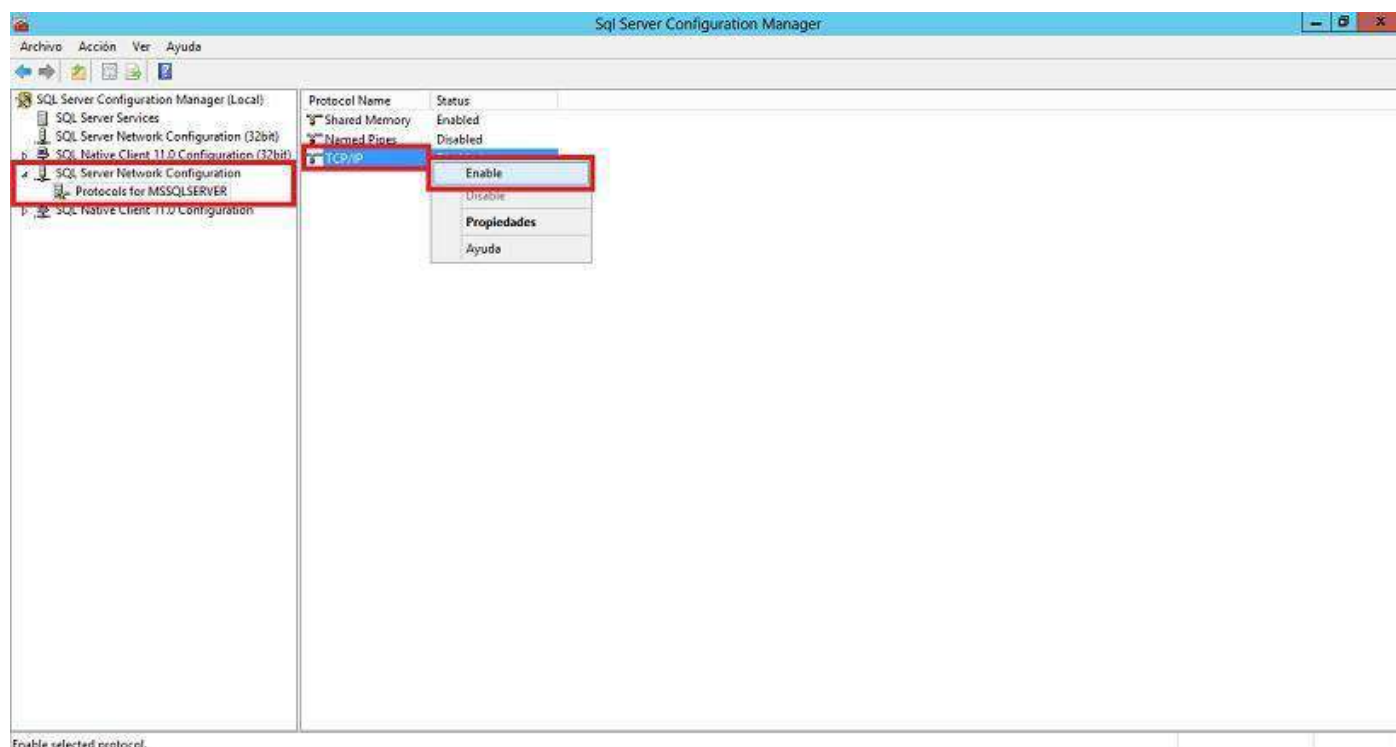
br> Se abra la ventana SQL Server Configuration Manager, dentro del listado de opciones buscaremos la opción SQL Server Network Configuration posteriormente vamos a expandir las opciones y vamos a realizar un clic en el elemento Protocols for MSSQLSERVER, donde encontraremos los protocolos Named Pipes y TCP/IP, los cuales se encuentran deshabilitados



Fuente: <https://help.mproerp.com/help/0000000058/>

1.1.5 Habilitamos Protocolo TCP/IP

El primer protocolo que vamos a habilitar es el TCP/IP, para realizar esto vamos a presionar el botón derecho sobre el nombre del protocolo y este nos mostrará una ventana alternativa en la cual vamos a seleccionar la opción Enable.



Fuente: <https://help.mproerp.com/help/0000000058/>

1.1.6 Advertencia de Sistema

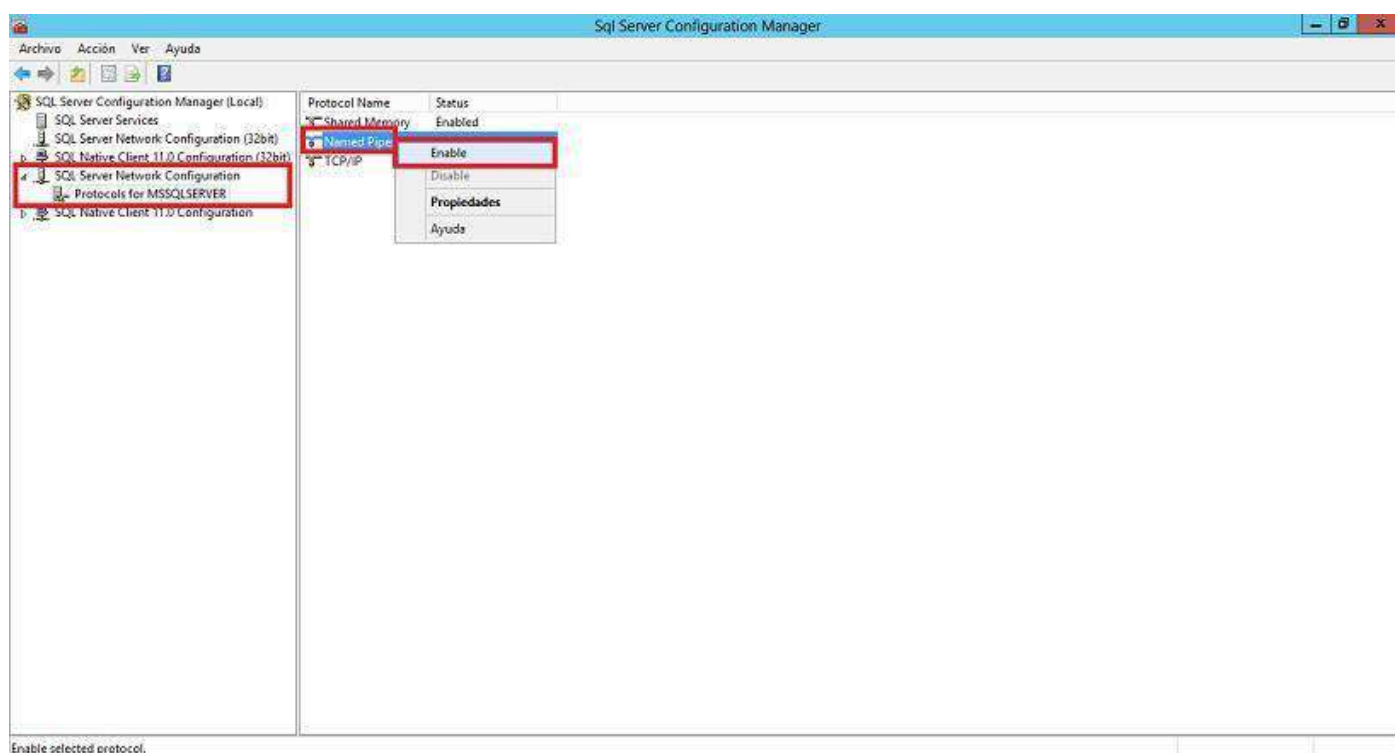
Después de habilitar el protocolo TCP/IP, el sistema nos informa que los cambios surtirán efecto después de reiniciar el servicio, vamos a presionar el botón Aceptar para continuar.



Fuente: <https://help.mproerp.com/help/000000058/>

1.1.7 Habilitamos Protocolo Named Pipes

El siguiente protocolo que vamos a habilitar es el Named Pipes, para realizar esto vamos a presionar el botón derecho sobre el nombre del protocolo y este nos mostrará una ventana alternativa en la cual vamos a seleccionar la opción Enable.



Fuente: <https://help.mproerp.com/help/000000058/>

1.1.8 Advertencia de Sistema

Después de habilitar el protocolo TCP/IP, el sistema nos informa que los cambios surtirán efecto después de reiniciar el servicio, vamos a presionar el botón Aceptar para continuar.

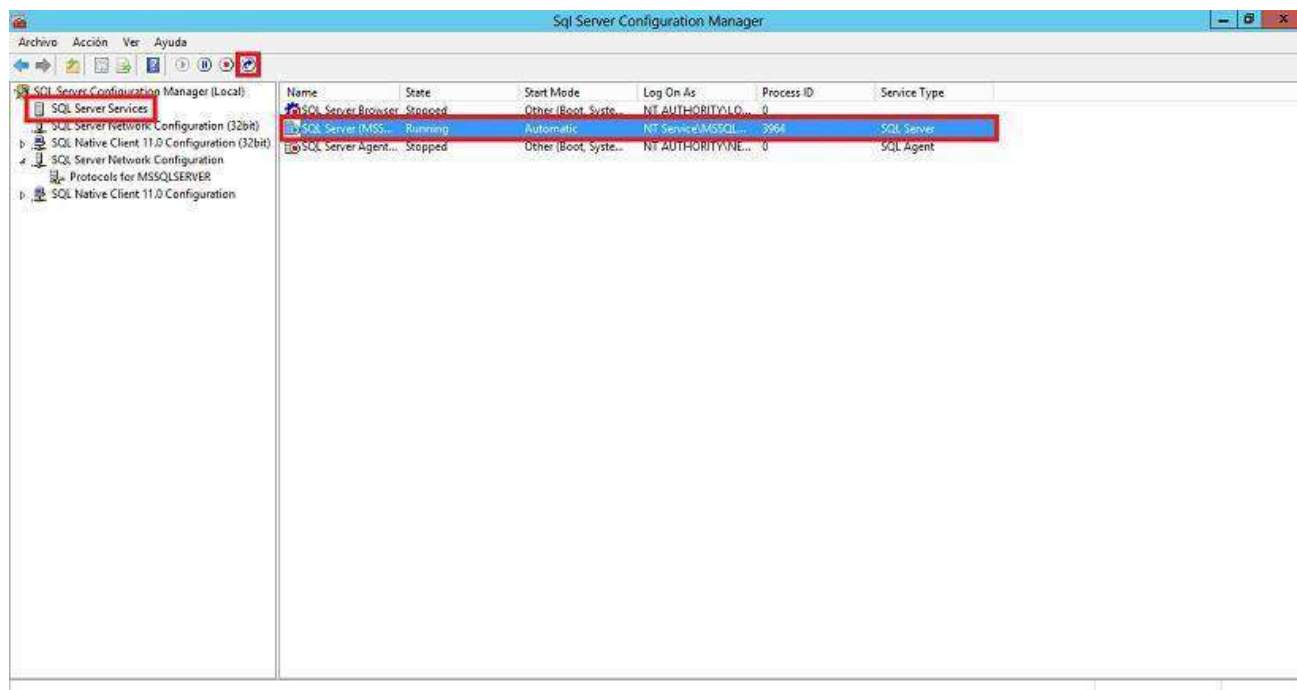
Fuente: <https://help.mproerp.com/help/0000000058/>



1.1.9 Reiniciar el Servicio SQL SERVER (MSSQLSERVER)

Después de habilitar los protocolos, vamos a seleccionar la opción SQL Server Services, donde nos mostrará 3 servicios, uno de los cuales se llama SQL SERVER (MSSQLSERVER) el cual vamos a seleccionar y posteriormente presionaremos el botón de reiniciar servicio que está indicado con una flecha en color azul.

Una vez que se reinicie el servicio, nuestra base de datos estará en condiciones óptimas para funcionar con el sistema.



Fuente: <https://help.mproerp.com/help/0000000058/>

1.1.10 SQL Sintaxis

Select

```
SELECT "nom de colonne" FROM "nombre_tabla";
```

Distinct

```
SELECT DISTINCT "nombre_columna"  
FROM "nombre_tabla";
```

Where

```
SELECT "nombre_columna"  
FROM "nombre_tabla" WHERE  
"condition";
```

And/Or

```
SELECT "nombre_columna"  
FROM "nombre_tabla" WHERE  
"condición simple"  
{[AND | OR] "condición simple"}+;
```

In

```
SELECT "nombre_columna"  
FROM "nombre_tabla"  
WHERE "nombre_columna" IN ('valor1', 'valor2', ...);
```

Between

```
SELECT "nombre_columna"  
FROM "nombre_tabla"  
WHERE "nombre_columna" BETWEEN 'valor1' AND 'valor2';
```

Like

```
SELECT "nombre_columna"  
FROM "nombre_tabla"  
WHERE "nombre_columna" LIKE {patrón};
```


Order By

```
SELECT "nombre_columna"
FROM "nombre_tabla" [WHERE
"condición"]
ORDER BY "nombre_columna" [ASC, DESC];
```

Count

```
SELECT COUNT("nombre_columna")
FROM "nombre_tabla";
```

Group By

```
SELECT "nombre_columna 1", SUM("nombre_columna 2")FROM
"nombre_tabla"
GROUP BY "nombre_columna 1";
```

Having

```
SELECT "nombre_columna 1", SUM("nombre_columna 2")FROM
"nombre_tabla"
GROUP BY "nombre_columna 1"
HAVING (condición de función aritmética);
```

Create Table

```
CREATE TABLE "nombre_tabla"
("columna 1" "tipo_de_datos_para_columna_1",
"columna 2" "tipo_de_datos_para_columna_2",
... );
```

Drop Table

```
DROP TABLE "nombre_tabla";
```

Truncate Table

```
TRUNCATE TABLE "nombre_tabla";
```

Insert into

```
INSERT INTO 'nombre_tabla' ('colonne 1', 'colonne 2', ...)
```

```
VALUES ("valor 1", "valor 2", ...);
```

Update

```
UPDATE "nombre_tabla"
SET "colonne 1" = [nuevo valor]
WHERE "condición";
```

Delete From

```
DELETE FROM "nombre_tabla"
WHERE "condición";
```

1.1.11 Comentar código

Al escribir sentencias SQL en el editor de SQL y XQuery, se pueden añadir comentarios al código SQL de forma rápida mediante un mandato del menú emergente. El mismo mandato puede utilizarse para quitar las marcas de comentario del código SQL.

Se pueden añadir o quitar marcas de comentarios de una sola línea de código en una sentencia SQL, varias líneas adyacentes de código, una sentencia SQL completa o varias sentencias SQL adyacentes.

La sintaxis para un comentario en una línea de código SQL es un guión doble (--) al comienzo de la línea. El comentario afecta a todo el código SQL de la línea.

1.1.12 Procedimiento

Para poner o quitar marcas de comentario del código SQL en el editor SQL y XQuery:

Seleccione el código SQL para el que desea poner o quitar marcas de comentario.

Para una sola línea, pulse en cualquier lugar de la línea de código.

Para varias líneas, arrastre el puntero por las líneas de código.

Pulse con el botón derecho del ratón sobre el código SQL seleccionado y, a continuación, seleccione Habilitar o inhabilitar comentario.

1.1.13 Resultados

El resultado de este proceso depende del estado original del código SQL seleccionado:

Para una sola línea de código:

Si la línea estaba activa, la línea se convierte en un comentario. Si la

línea era un comentario, la línea pasa a estar activa.

Para varias líneas de código:

Si cualquiera de las líneas estaba activo, todas las líneas pasan a ser comentarios. Tenga en cuenta que las líneas que anteriormente eran comentarios tienen ahora una marca de comentario doble.

Si todas las líneas eran comentarios, todas ellas pasan a estar activas.

1.1.14 SQL SELECT

Una de las sentencias SQL más importantes es SELECT, ya que permite realizar consultas sobre los datos almacenados en la base de datos.

1.1.15 Sintaxis SQL SELECT

`ELECT * FROM nombretabla`

`SELECT columna1, columna2 FROM nombretabla`

Para los ejemplos, tendremos la siguiente tabla de personas denominada "personas" Estos son los datos almacenados en la tabla "personas"

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
ANTONIO	GARCIA	BENITO
LUIS	LOPEZ	PEREZ

Fuente: <http://sql.11sql.com/sql-select.htm>

1.1.16 Resultados

Si queremos consultar todos los datos de la tabla "personas"

```
SELECT * FROM personas
```

Este será el resultado:

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
ANTONIO	GARCIA	BENITO
LUIS	LOPEZ	PEREZ

Fuente: <http://sql.11sql.com/sql-select.htm>

Si queremos consultar todos los datos de la tabla "personas"

```
SELECT * FROM personas
```

Este será el resultado:

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
ANTONIO	GARCIA	BENITO
LUIS	LOPEZ	PEREZ

Fuente: <http://sql.11sql.com/sql-select.htm>

Si queremos consultar todos los nombres y primer apellido de todas las personas

```
SELECT nombre, apellido1 FROM personas
```

Este será el resultado:

nombre	apellido1
ANTONIO	PEREZ
ANTONIO	GARCIA
LUIS	LOPEZ

Fuente: <http://sql.11sql.com/sql-select.htm>

1.6 SQL SELECT DISTINCT

Por: Erick Estuardo Moran Pech

se usa para eliminar las columnas duplicadas del conjunto de resultados.

emp_id	emp_name	emp_sal_paid	date_paid	
1	1	Mike	4000.75	2016-07-01 00:00:00.000
2	1	Mike	4500.55	2017-02-01 00:00:00.000
3	1	Mike	5000.35	2017-03-01 00:00:00.000
4	2	Michale	4500.56	2017-06-01 00:00:00.000
5	2	Michale	4500.77	2017-07-01 00:00:00.000
6	3	Jimmy	3000.86	2017-06-01 00:00:00.000
7	3	Jimmy	3500.75	2017-07-01 00:00:00.000
8	4	Shaun	3500.45	2017-06-01 00:00:00.000

emp_name	
1	Ben
2	Jimmy
3	Michale
4	Mike
5	Shaun

SQL DISTINCT

<https://www.jquery-az.com/wp-content/uploads/2018/04/7-1-SQL-DISTINCT.png>

1.7 SQL y, o, no

Los operadores AND y OR de SQL son operadores booleanos utilizados para especificar condiciones compuestas en una cláusula WHERE.



https://datascientest.com/es/wp-content/uploads/sites/7/2021/09/illu_SQLvsNoSQL-46.png.webp

1.8 SQL Ordenar por

Una cláusula ORDER BY en SQL especifica que una instrucción SQL SELECT devuelve un conjunto de resultados con las filas ordenadas por los valores de una o más columnas.

```
70
71 -- Ordenar los resultados por un campo o columna
72 SELECT * FROM productos ORDER BY id_marca;
73
```

	modelo	id_marca	id_categoria	id_subcategoria
1	BL	2	677	683
2	2R4	5	616	585
3	I-53PM	5	616	585
4	I-315R	5	616	585
5	8TB	5	616	585

Execution finished without errors.
Result: 14767 rows returned in 364ms
At line 72:
SELECT * FROM productos ORDER BY id_marca;

<https://www.asesoriaensig.com.mx/wp-content/uploads/2021/07/image-97.png>

1.9 SQL Inserir y Actualizacion

Una instrucción SQL INSERT agrega uno o más registros a cualquier tabla individual en una base de datos relacional, Una declaración SQL UPDATE cambia los datos de uno o más registros en una tabla.

```
onExecute
Theme dracula
1 sc_exec_sql("INSERT INTO customer (name, email, phone, address) VALUES ('$name',
2 '$email', '$phone', '$address')");
3 sc_exec_sql("UPDATE customer SET phone = $phone WHERE customer_id = $id");
4
5 sc_exec_sql("DELETE FROM customer WHERE customer_id = $id");
6
7
8 //returning a two-dimentional array
9
10 sc_lookup(ds, "SELECT name, email FROM customer WHERE customer_id = $id");
11
12 $name = {ds[0][0]};
13 $email = {ds[0][1]};
```

<https://desk.zoho.com/DocsDisplay?zgid=716875472&mode=inline&blockid=k97k9a45b0354e0bb49b28d052f5928d39fd7>

1.10 select de sql top

La cláusula SELECT TOP en SQL se usa para especificar el número de registros a devolver. La cláusula SELECT TOP es útil en tablas grandes con miles de registros.

```
9
10 DELETE FROM [dbo].[SalesData]
11 WHERE CustomerId IN
12 (SELECT TOP 10 CustomerId
13 FROM [dbo].[SalesData]
14 ORDER BY OrderDate ASC);
15 GO
16
17
```

100 %

Messages

{13 rows affected}

<https://www.sqlshack.com/wp-content/uploads/2020/03/sql-delete-statement.png>

1.1 SQL SERVER 2014

Por: Maria Jose Noj Sor.

permite a los usuarios ejecutar aplicaciones y contra los datos almacenados en tablas optimizadas para memoria en lugar del estándar basadas en disco.

1.1.1 SQL mínimo y máximo.

La función MIN () devuelve el valor más pequeño de la columna seleccionada. La función MAX () devuelve el valor más grande de la columna seleccionada.

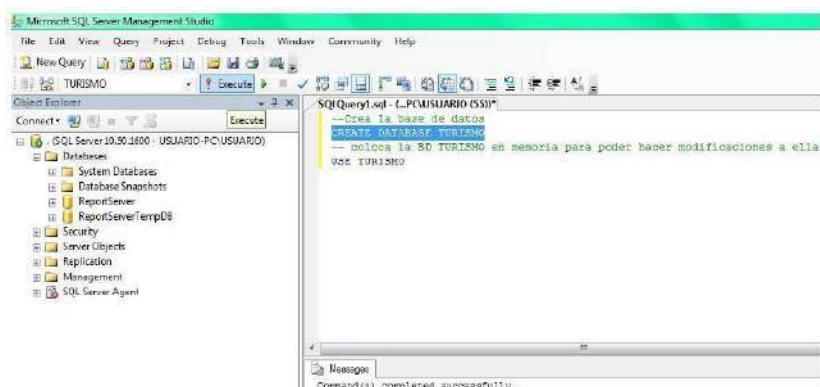
1.1.2 Conde SQL, Media, Suma.

La función SQL SUM () devuelve la suma total de una columna numérica La función SQL avg devuelve el promedio del número total de registros presentes en la columna especificada.

1.1.3 SQL como crear SQL DB

Para crear una base de datos

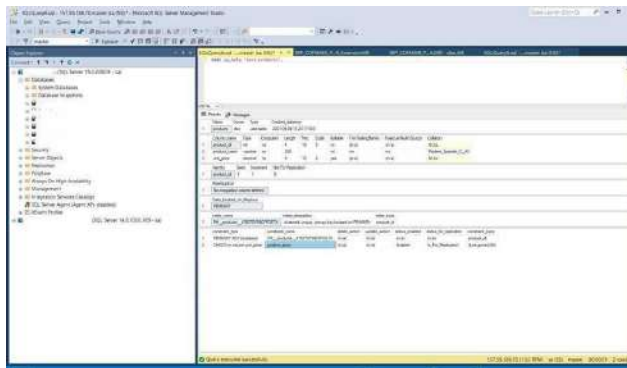
1. En el Explorador de objetos, conéctese a una instancia del Motor de base de datos de SQL Server y expándala.
2. Haga clic con el botón derecho en Bases de datos y, después, seleccione Nueva base de datos.
3. En Nueva base de datos, especifique un nombre de base de datos.



<https://www.jquery-az.com/wp-content/uploads/2018/04/7-1-SQL-DISTINCT.png>

1.1.4 SQL comodines y Check

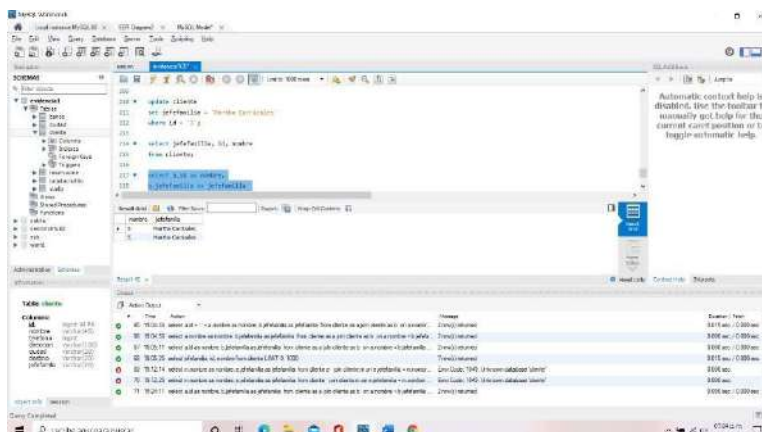
Un carácter comodín se utiliza para sustituir cualquier otro carácter en una cadena. Los caracteres comodines se utilizan con el operador SQL LIKE. El operador LIKE se utiliza en una cláusula WHERE. para buscar. un patrón específico en una columna La restricción CHECK en SQL se usa para limitar el rango de valores que se puede colocar en una columna, por ejemplo, Si define una restricción CHECK en una sola columna, solo se permiten ciertos valores para esta columna.



<https://www.jquery-az.com/wp-content/uploads/2018/04/7-1-SQL-DISTINCT.png>

1.1.5 alias de SQL Inner Join

Un alias en SQL es una manera de renombrar una columna o tabla, cuando estas tienen el mismo nombre y no queremos que haya conflictos entre ellas, mediante un alias le damos un nombre diferente para trabajar sin problemas.



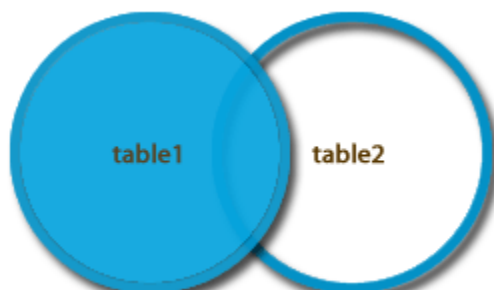
<https://www.jquery-az.com/wp-content/uploads/2018/04/7-1-SQL-DISTINCT.png>

1.1 SQL SERVER 2014

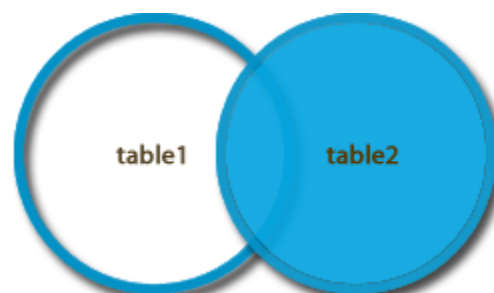
Por: José Antonio Osoy Crespo

1.1.1 SQL Left Join y Right Join

LEFT JOIN: Devuelve todas las filas de la tabla de la izquierda, y las filas coincidentes de la tabla de la derecha.



RIGHT JOIN: Devuelve todas las filas de la tabla de la derecha, y las filas coincidentes de la tabla de la izquierda.

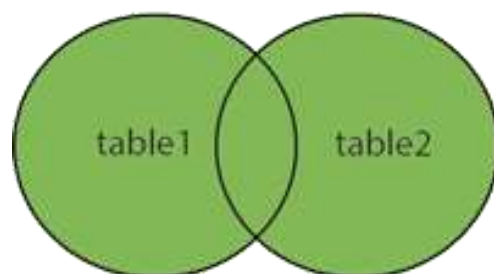


Fuente: <https://programacionymas.com/blog/como-funciona-inner-left-right-full-join>

1.1.2 SQL Full Join

La Full Join o Full Outer Join devuelve un conjunto de resultados que incluye filas de las tablas izquierda y derecha.

FULL OUTER JOIN



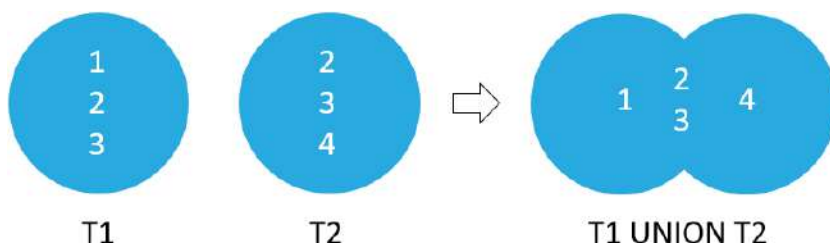
Fuente: <https://keepcoding.io/blog/como-funciona-full-join-en-sql/>

1.1.3 SQL Self Join, Union

SELF JOIN: Permite unir una tabla a sí misma. Es útil para consultar datos jerárquicos o comparar filas dentro de la misma tabla.



UNION: Es una instrucción de conjunto que permite combinar resultados de dos instrucciones SELECT en un solo conjunto de resultados que incluye todas las filas que pertenecen a las instrucciones SELECT en la unión.



Fuente: <https://www.educba.com/mysql-self-join/>

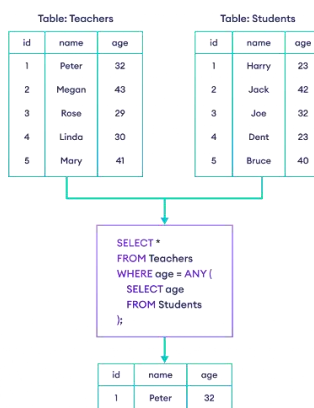
1.1.4 SQL Group By y Any, All

BY: Identifica una columna seleccionada para utilizarla para agrupar resultados.

title	genre	qty
book 1	adventure	4
book 2	fantasy	5
book 3	romance	2
book 4	adventure	3
book 5	fantasy	3
book 6	romance	1

genre	total
adventure	7
fantasy	8
romance	3

ANY: Devuelve un valor booleano como resultado.



ALL: devuelve VERDADERO si TODOS los valores de la subconsulta cumplen la condición.

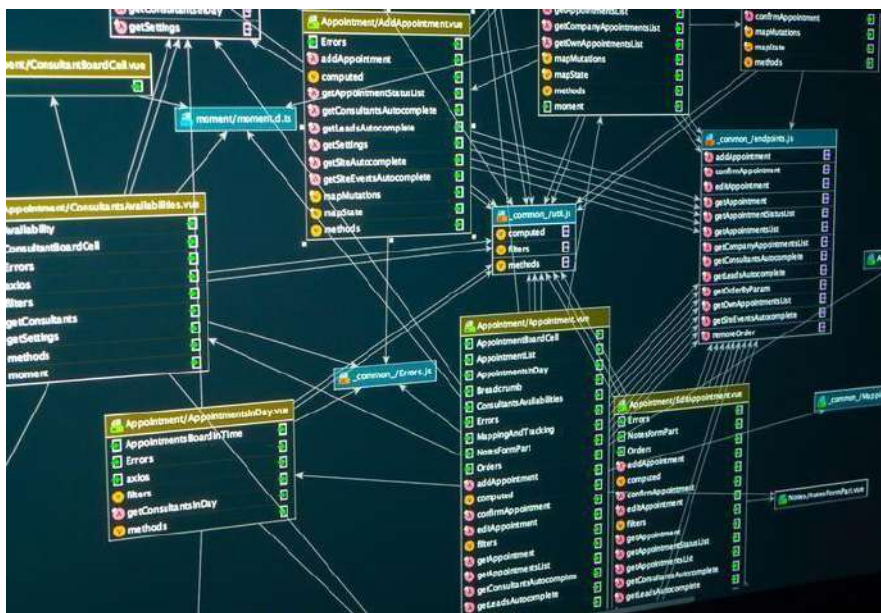
topic_id	subject	sessions	expert_name	charges
1	SQL	750	Payal	3750.00
2	MySQL	700	Vyankatesh	3500.00
3	PostgreSQL	600	Omprakash	3000.00
4	Java	750	Payal	2750.00
5	Hibernate	700	Vyankatesh	2500.00
6	Maven	600	Omprakash	2000.00
7	Javascript	950	Payal	4750.00
8	Angular	800	Vyankatesh	4500.00
9	React	800	Omprakash	4000.00

select * from educba_learning LIMIT 0, 1000

Fuente: <https://www.educba.com/sql-group-by-multiple-columns/>

1.1.5 SQL Procedimientos

Los procedimientos almacenados de SQL pueden ser instrucciones de tipo Transact-SQL o referencias a un método de Common Runtime Language (CLR) de .NET. Dichas instrucciones se encuentran almacenadas de forma física con un nombre dentro de la base de datos.



Fuente: <https://blog.mdcloud.es/procedimientos-almacenados-sql-server/>

1.1 SQL comentarios de db

Por: Diego Racanac

SQL es un lenguaje de programación utilizado para administrar y manipular bases de datos relacionales. En este ensayo, exploraremos cómo agregar comentarios a una base de datos en SQL y cómo estos comentarios pueden ayudar a la comprensión y mantenimiento de la base de datos.

1.1.1 ¿Por qué agregar comentarios a una base de datos?

Agregar comentarios a una base de datos puede ayudar a que otros usuarios comprendan cómo funciona la base de datos y qué datos se almacenan en ella. Los comentarios también pueden ayudar a los programadores a comprender el propósito de las tablas, columnas y relaciones en la base de datos, lo que facilita el mantenimiento y la solución de problemas.

1.1.2 Agregar comentarios a una tabla

Para agregar un comentario a una tabla en SQL, se utiliza la instrucción "COMMENT". Por ejemplo, para agregar un comentario a una tabla llamada "clientes", se puede usar la siguiente sintaxis:

```
COMMENT ON TABLE clientes IS 'Tabla que almacena información sobre los clientes';
```

1.1.3 Agregar comentarios a una columna

Para agregar un comentario a una columna en SQL, se utiliza la misma sintaxis que para agregar un comentario a una tabla, pero se especifica el nombre de la columna en lugar del nombre de la tabla. Por ejemplo, para agregar un comentario a la columna "nombre" en la tabla "clientes", se puede usar la siguiente sintaxis:

```
COMMENT ON COLUMN clientes.nombre IS 'El nombre del cliente';
```

1.1.4 Agregar comentarios a una base de datos

Para agregar un comentario a una base de datos en SQL, se puede utilizar la instrucción "COMMENT" seguida del nombre de la base de datos. Por ejemplo, para agregar un comentario a una base de datos llamada "ventas", se puede usar la siguiente sintaxis:

```
COMMENT ON DATABASE ventas IS 'Base de datos que almacena información sobre las ventas';
```

En este ejemplo, se agrega un comentario a la base de datos "ventas" que describe brevemente el propósito de la base de datos.

Conclusión

Agregar comentarios a una base de datos en SQL puede ayudar a que los usuarios comprendan cómo funciona la base de datos y qué datos se almacenan en ella. Los comentarios también pueden ayudar a los programadores a comprender el propósito de las tablas, columnas y relaciones en la base de datos, lo que facilita el mantenimiento y la solución de problemas.

1.2 SQL de copia de seguridad db

La copia de seguridad de una base de datos es una tarea fundamental para cualquier administrador de sistemas y/o de bases de datos. La base de datos contiene información crítica que es necesaria para el funcionamiento de la organización, por lo que es necesario contar con una copia de seguridad en caso de un desastre o falla en el sistema.

1.1.5 ¿Qué es una copia de seguridad de una base de datos?

Una copia de seguridad de una base de datos es una copia de los datos almacenados en la base de datos en un momento determinado. Esta copia se realiza para proteger los datos de la base de datos en caso de una falla en el sistema, un error humano, un ataque cibernético, entre otros posibles escenarios.

1.1.6 ¿Cómo realizar una copia de seguridad de una base de datos utilizando SQL?

Para realizar una copia de seguridad de una base de datos utilizando SQL, se debe seguir los siguientes pasos:

Conectar a la base de datos: Lo primero que se debe hacer es conectarse a la base de datos que se desea respaldar. Para ello, se utiliza el comando "USE" seguido del nombre de la base de datos.

Ejemplo: USE mi_base_datos;

Crear un archivo de copia de seguridad: Se debe crear un archivo donde se almacenará la copia de seguridad de la base de datos.

Ejemplo: BACKUP DATABASE mi_base_datos TO DISK = 'C:\backup\mi_base_datos.bak';

Verificar la copia de seguridad: Una vez creado el archivo de copia de seguridad. Para ello, se puede utilizar el comando "RESTORE VERIFYONLY".

Ejemplo: RESTORE VERIFYONLY FROM DISK = 'C:\backup\mi_base_datos.bak';

Realizar la restauración de la copia de seguridad: En caso de necesitar restaurar la copia de seguridad, se utiliza el comando "RESTORE DATABASE". Este comando permite restaurar la base de datos en su estado previo.

Ejemplo: RESTORE DATABASE mi_base_datos FROM DISK = 'C:\backup\mi_base_datos.bak';

Conclusiones

La copia de seguridad de una base de datos es una tarea importante para garantizar la integridad y disponibilidad de los datos. La realización de la copia de seguridad es una tarea sencilla utilizando SQL, ya que existen comandos específicos para ello.

SQL crear la tabla y DROP

SQL es un lenguaje de programación utilizado para interactuar con bases de datos relacionales. La creación y eliminación de tablas es una de las tareas más comunes en SQL. En este ensayo, exploraremos cómo crear y eliminar tablas en SQL.

1.1.7 Creación de una tabla en SQL

Para crear una tabla en SQL, se utiliza la sentencia CREATE TABLE. Esta sentencia especifica el nombre de la tabla, así como las columnas y sus tipos de datos.

Aquí hay un ejemplo de una sentencia CREATE TABLE simple:

```
CREATE TABLE productos (
  id INT PRIMARY KEY,
  nombre VARCHAR(50),
  precio DECIMAL(10, 2),
  disponible BOOLEAN
);
```

Fuente: <https://www.tutorialesprogramacionya.com/sqlserverya/temarios/descripcion.php?cod=8>

En este ejemplo, se está creando una tabla llamada "productos". La tabla tiene cuatro columnas: "id", "nombre", "precio" y "disponible". La columna "id" es la clave primaria de la tabla, lo que significa que cada fila tendrá un valor único en esta columna. Las columnas "nombre" y "disponible" son de tipo VARCHAR y BOOLEAN respectivamente, mientras que la columna "precio" es de tipo DECIMAL con una precisión de 10 y una escala de 2.

1.1.8 Eliminación de una tabla en SQL

Para eliminar una tabla en SQL, se utiliza la sentencia DROP TABLE. Esta sentencia especifica el nombre de la tabla que se desea eliminar.

Aquí hay un ejemplo de una sentencia DROP TABLE simple:

```
DROP TABLE productos;
```

Fuente: <https://www.youtube.com/watch?v=StSX1QsfnlQ> En

este ejemplo, se está eliminando la tabla "productos".

Conclusión

Crear y eliminar tablas son tareas fundamentales en SQL. La sentencia CREATE TABLE se utiliza para crear una tabla con columnas y tipos de datos específicos, mientras que la sentencia DROP TABLE se utiliza para eliminar una tabla existente.

Modificar tablas y restricciones de SQL

Modificar una tabla

1.1.9 Agregar una columna a una tabla existente

Para agregar una columna a una tabla existente, puedes utilizar la sentencia ALTER TABLE seguida del nombre de la tabla y la cláusula ADD COLUMN seguida del nombre y tipo de datos de la nueva columna. Por ejemplo:

```
ALTER TABLE mi_tabla ADD COLUMN nueva_columna INTEGER;
```

1.1.10 Eliminar una columna de una tabla existente

Para eliminar una columna de una tabla existente, puedes utilizar la sentencia ALTER TABLE seguida del nombre de la tabla y la cláusula DROP COLUMN seguida del nombre de la columna que deseas eliminar. Por ejemplo:

```
ALTER TABLE mi_tabla DROP COLUMN columna_a_eliminar;
```

Modificar una restricción

1.1.11 Agregar una restricción a una tabla existente

Para agregar una restricción a una tabla existente, puedes utilizar la sentencia ALTER TABLE seguida del nombre de la tabla y la cláusula ADD CONSTRAINT seguida del nombre de la restricción y la cláusula que define la restricción.

1.1.12 Eliminar una restricción de una tabla existente

Para eliminar una restricción de una tabla existente, puedes utilizar la sentencia ALTER TABLE seguida del nombre de la tabla y la cláusula DROP CONSTRAINT seguida del nombre de la restricción que deseas eliminar.

SQL de clave principal

En SQL, una clave principal (primary key) es un campo o conjunto de campos que identifican de manera única cada registro de una tabla. La clave principal se utiliza para garantizar la integridad de los datos y evitar la duplicación de registros.

Para definir una clave principal en una tabla en SQL, se utiliza la cláusula PRIMARY KEY después de la definición de la tabla y se especifican los nombres de los campos que formarán parte de la clave principal, separados por comas. Por ejemplo:

```
CREATE TABLE clientes (
  id_cliente INT PRIMARY KEY,
  nombre VARCHAR(50),
  direccion VARCHAR(100),
  telefono VARCHAR(20)
);
```

Fuente: <https://www.mundosql.eu/2014/07/28/crear-indices-o-clave-primaria-con-t-sql/>

En este ejemplo, se crea una tabla llamada "clientes" con cuatro campos: "id_cliente", "nombre", "direccion" y "telefono". La clave principal está definida por el campo "id_cliente", que es de tipo INT.

Otro ejemplo de definición de clave principal en una tabla sería el siguiente:

```
CREATE TABLE pedidos (
  id_pedido INT,
  id_cliente INT,
  fecha_pedido DATE,
  total DECIMAL(10,2),
  PRIMARY KEY (id_pedido, id_cliente)
);
```

Fuente: <https://juankartips.wordpress.com/tag/clave-externa/>

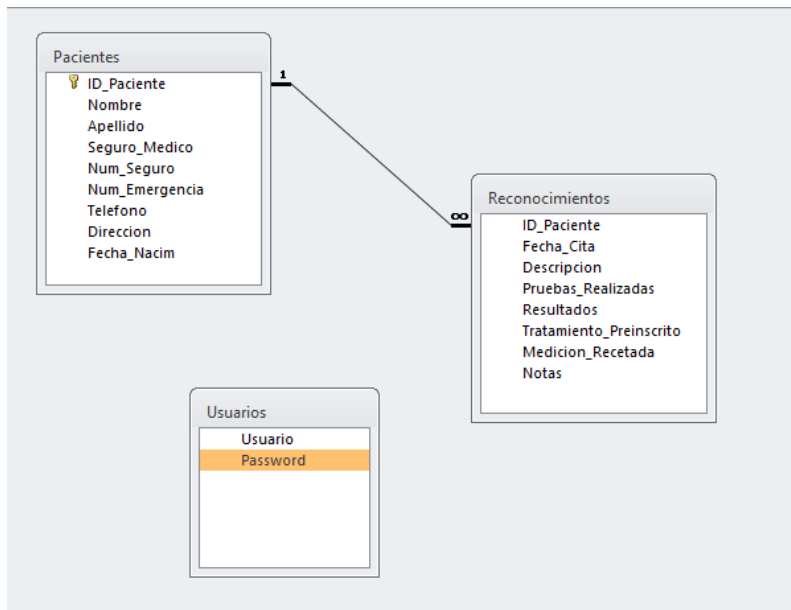
En este caso, se crea una tabla llamada "pedidos" con cuatro campos: "id_pedido", "id_cliente", "fecha_pedido" y "total". La clave principal está definida por una combinación de dos campos: "id_pedido" y "id_cliente". Esto significa que cada registro en la tabla debe tener un valor único para la combinación de estos dos campos.

En resumen, una clave principal en SQL es una herramienta importante para garantizar la integridad de los datos y evitar la duplicación de registros en una tabla. Se puede definir utilizando la cláusula PRIMARY KEY y especificando los campos que la conforman.

1.26 SQL clave externa

Por: Pavel Palma

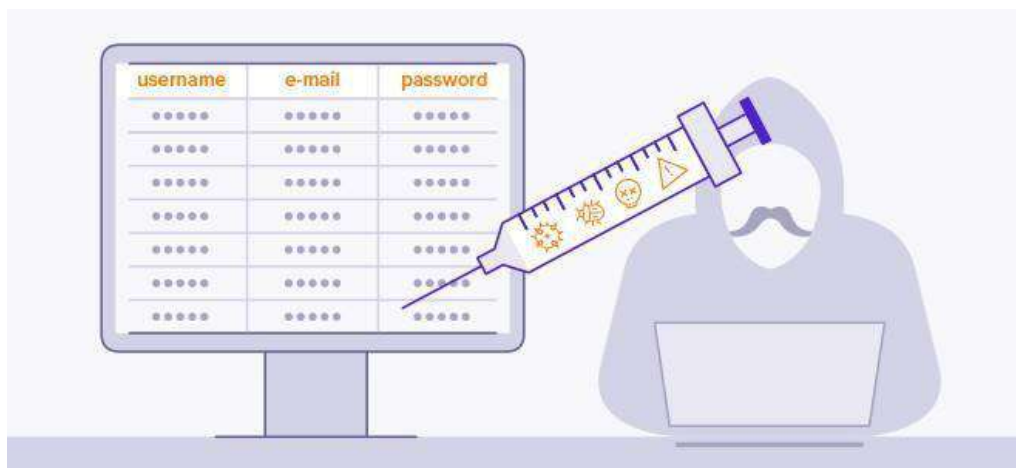
Es un campo que nos señala la clave primaria de otra tabla, pues el propósito es asegurar la integridad referencial de los datos. En otra palabra este solo permite los valores que se parezcan en la base de datos.



Fuente: [Crear aplicacion usando clave foranea y clave primaria \(microsoft.com\)](https://www.microsoft.com/es-es/sql-server/sql-server-developer/quickstart/getting-started/relationships)

1.27 Inyecciones SQL

Es un tipo de vulnerabilidad en la que un atacante usa un trozo de código SQL para poder manipular una base de datos y poder acceder a la información potencialmente valiosa. Este es uno de los ataques mas frecuentes y amenazadores, donde pueden atacar prácticamente de cualquier aplicación o sitio web que use una base de datos en SQL.



Fuente: [¿Qué es la inyección de SQL? | Explicación y protección | Avast](https://www.avast.com/es-es/what-is-sql-injection)

1.28 SQL predeterminado

Un valor predeterminado se inserta solo cuando no esta presente al ingresar un registro y en algunos casos que el dato ingresado es invalido.

Para campos de cualquier tipo de dato no declarado le ponemos "not null", es decir, que admite valores nulos, pues el valor por defecto es "null", y para los campos declarados "not null", no existe ningún valor por defecto, a menos de que se declare la clausula "default".

1.29 SQL incremento automático

En ciertas ocasiones es necesario la creación de un campo autoincremental para usar como un identificador de una tabla.

Pues SQL un campo comienzo y uno de incremento para el autoincrementable, por ejemplo: ID int PRIMARY KEY, en donde int es de donde debe empezar a contar e incremento es el valor donde va incrementando.

```
/*
  parzibyte.me/blog
*/
CREATE TABLE videojuegos(
  id bigint identity(100, 2) primary key,
  nombre varchar(50) NOT NULL
);

insert into videojuegos (nombre) values
('Super Mario Bros'), ('Snow Bros'), ('Metal Slug'),
('Resident Evil 4'), ('Color Sheep'), ('Cut the rope');

select * from videojuegos;
```

Fuente: [Definir tabla con autoincremento en SQL Server - Parzibyte's blog](https://parzibyte.me/blog/2018/07/27/definir-tabla-con-autoincremento-en-sql-server/)

1.1 SQL SERVER 2014

Por: Wilmer Patzán

SQL Server es uno de los principales sistemas de gestión de bases de datos relacionales del mercado que presta servicio a un amplio abanico de aplicaciones de software destinadas a la inteligencia empresarial y análisis sobre entornos corporativos, basada en el lenguaje Transact-SQL, incorpora un conjunto de extensiones de programación propias de lenguaje estándar y su aplicación está disponible para usarse tanto a nivel on premise o bajo una modalidad cloud.

1.1.1 Alojamiento SQL

El alojamiento de bases de datos de lenguaje de consulta estructurado (SQL) se utiliza para alojar la base de datos de un sitio web o de una empresa en línea, lo que lo hace más fácil y generalmente más económico que administrar un servidor de base de datos internamente el alojamiento compartido de bases de datos SQL es una solución más barata y el usuario posee parte de un servidor, pero esto generalmente solo es bueno para una base de datos pequeña.

1.1.2 Modificar tabla y restricciones

Las restricciones son reglas que el motor de base de datos de SQL Server exige. Por ejemplo, puede usar restricciones UNIQUE para garantizar que no se escriben valores duplicados en columnas específicas que no forman parte de una clave principal.

1.1.3 SQL no nulos y únicos

La restricción NOT NULL impone una columna para NO aceptar valores NULL. Esto obliga a que un campo siempre contenga un valor, lo que significa que no puede insertar un nuevo registro o actualizar un registro sin agregar un valor a este campo. Por defecto, las columnas pueden contener valores NULL. Se usa una restricción NOT NULL en SQL para evitar insertar valores NULL en la columna especificada, considerándolo entonces como un valor no aceptado para esa columna.

Imagen: Ejemplo del tamaño solicitado (información) Fuente:
<https://www.google.com/search?q=imagen+ejemplo>



Fuente: <https://masrobeznoquenunca.wordpress.com/2016/09/26/instalar-microsoft-sql-server-2014-en-un-servidor/>



Fuente: <https://www.pngegg.com/es/png-tryef>

CREACIÓN DE RESTRICCIONES AL MOMENTO DE CREAR UNA TABLA

```

CREATE TABLE Productos
(
  ProductID int IDENTITY (1,1) NOT NULL,
  ProductName nvarchar (40) NOT NULL CONSTRAINT U_NAME UNIQUE,
  SupplierID int NULL,
  CategoryID int NULL,
  QuantityPerUnit nvarchar (20) NULL,
  UnitPrice money NULL CONSTRAINT DF_Products_UnitPrice DEFAULT(0),
  UnitsInStock smallint NULL CONSTRAINT DF_Products_UnitsInStock DEFAULT(0),
  UnitsOnOrder smallint NULL CONSTRAINT DF_Products_UnitsOnOrder DEFAULT(0),
  ReorderLevel smallint NULL CONSTRAINT DF_Products_ReorderLevel DEFAULT(0),
  Discontinued bit NOT NULL CONSTRAINT DF_Products_Discontinued DEFAULT(0),
  CONSTRAINT PK_Products PRIMARY KEY CLUSTERED (ProductID),
  CONSTRAINT FK_Products_Categories FOREIGN KEY (CategoryID)
REFERENCES dbo.Categories(CategoryID) ON UPDATE CASCADE,
  CONSTRAINT FK_Products_Suppliers FOREIGN KEY (SupplierID)
REFERENCES dbo.Suppliers(SupplierID) ON DELETE CASCADE,
  CONSTRAINT CK_Products_UnitPrice CHECK (UnitPrice >= 0),
  CONSTRAINT CK_ReorderLevel CHECK (ReorderLevel >= 0),
  CONSTRAINT CK_UnitsInStock CHECK (UnitsInStock >= 0),
  CONSTRAINT CK_UnitsOnOrder CHECK (UnitsOnOrder >= 0)
)
GO
    
```

UNIQUE Restricción de valor único, permite que solo se permitan valores únicos sobre este campo.

DEFAULT Restricción de valor predeterminado, permite que de no llegarse el campo el asigne de manera automática un valor.

PRIMARY KEY Restricción de llave primaria, garantiza que este campo no permita duplicados y será el identificador único de cada fila.

FOREIGN KEY Una Restricción de llave foránea hace que este campo guarde relación con un campo de otra tabla, al que hace referencia y no permite ingresar un valor si no existe en el campo de la otra tabla.

CHECK Restricción de regla de validación, no permite ingresar un valor al campo si este no cumple con la condición especificada.

←

←

←

←

←

Fuente: <https://visoal.net/2014/12/explicacion-grafica-de-la-creacion-de-constraint-en-una-tabla-de-sql-server/>

```
SELECT * FROM ConstraintDemo1
```

100 %

Results Messages

	ID	Name
1	1	Ali
2	2	NULL

Fuente: <https://www.sqlshack.com/es/restricciones-en-sql-server-sql-not-null-unique-y-sql-primary-key/>

2.1 GNU/Linux en un entorno de trabajo

Por: Leonel Pérez

GNU/Linux es un sistema operativo compuesto por dos partes principales: GNU y Linux. GNU es una colección de herramientas y aplicaciones de software libre y de código abierto, mientras que Linux es el núcleo del sistema operativo responsable de interactuar con el hardware de la computadora.

2.1.1 Objetivo

El objetivo principal de GNU/Linux es proporcionar un sistema operativo de computadora libre y de código abierto que sea estable, seguro, flexible y personalizable para una amplia variedad de usuarios y casos de uso. Como sistema operativo de código abierto, GNU/Linux permite a los usuarios tener acceso al código fuente del software, lo que significa que pueden modificarlo y distribuirlo libremente. Esto ha llevado a una gran comunidad de desarrolladores y usuarios que colaboran en la mejora continua del sistema operativo.

2.1.2 Credito

GNU/Linux fue creado por una comunidad de desarrolladores y usuarios que trabajaron juntos para crear, mejorar y mantener el software. Aunque el núcleo Linux fue creado por Linus Torvalds en 1991, el sistema operativo completo incluye muchas herramientas y aplicaciones de software libre y de código abierto creadas por la comunidad de desarrolladores de GNU.



Linux Torvalds

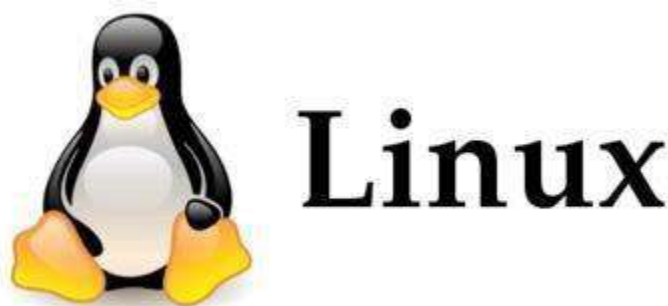
Fuente: <https://thurrott.s3.amazonaws.com/2018/09/linus-torvalds.jpg>

2.1.3 Licencias

GNU/Linux es un sistema operativo compuesto por herramientas y aplicaciones de software libre y de código abierto, y está licenciado bajo diversas licencias de software libre, incluyendo la Licencia Pública General de GNU (GPL) y la Licencia Pública Generalde GNU Menor (LGPL).

2.1.4 Conclusión

En conclusión, GNU/Linux es un sistema operativo de código abierto y libre que ha sido creado por una comunidad de desarrolladores y usuarios comprometidos con la filosofía del software libre. Es una alternativa popular y confiable a otros sistemas operativos comerciales, ya que ofrece una amplia gama de herramientas y aplicacionesque se pueden personalizar y mejorar según las necesidades del usuario.



Sistema Operativo Linux

Fuente: https://i.blogs.es/11abb8/650_1000_gnu-linux/450_1000.png



¿Qué son Unix, Linux y GNU?

Por: Krystel Popa

UNIX es un sistema operativo comercial que fue desarrollado a finales de la década de 1960 como simplificación de un proyecto más grande denominado Multics.

Linux es el núcleo de un sistema operativo tipo POSIX creado por LinuxTorvalds a principios de la década de 1990

GNU (GNU's Not Unix) es un esfuerzo por crear un sistema operativo libre (donde hay pleno acceso al código fuente y permiso para cambiarlo a voluntad) iniciado por Richard Stallman a mediados de la década de 1980.

Directorios y Comandos de UNIX:

- /bin/
- /boot/
- /dev/
- /etc/
- /etc/opt/

- Mkdir
- Pwd
- Cd
- Is
- Chmod

Directorios y Comandos de LINUX:

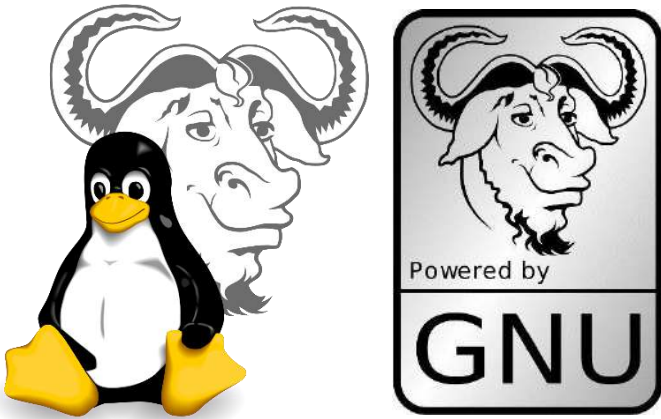
- /bin
- /boot /dev
- /etc
- /home
- /initrd

- Comando sudo

- Comando pwd
- Comando cd
- Comando ls
- Comando cat

El árbol de directorios:

Es el encargado de almacenar los archivos de configuración tanto a nivel de componentes del sistema operativo en sí, como de los programas y aplicaciones instaladas a posteriori. Es un directorio que debería contener únicamente ficheros de configuración, y no debería contener binarios.



Fuentes:

<https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Gnulinix.svg/1200px-Gnulinix.svg.png>

<https://2.bp.blogspot.com/-Ab7YO0yCkL0/TawvG7zgFvi/AAAAAAAAAN8/VCGZhvaFh0s/s1600/gnu.png>

1.1 GNU/LINUX

Por: Rodrigo Ramírez.

Este es un conjunto de programas que le permiten interactuar con el PC y de esa manera ejecutar otros programas.

Linux es un sistema operativo Unix quiere decir que trae aplicaciones, bibliotecas y herramientas.

2.8.3. Directorio de Trabajo

Llamado también **Directorio Raíz** contiene al resto del sistema de archivos. Es el directorio en el cual se está ejecutando un proceso, como la interpretación de comandos. Este es simbolizado por medio del punto (.).

2.8.4. Refiriéndose al directorio

Junto con los directorios de Linux, los archivos y directorios que crean los usuarios se almacenan en el directorio **/home**. En este directorio hay un subdirectorio para cada tipo de usuario.

Con este tenemos ventajas como el software para almacenar la información que necesita para su funcionamiento.

Los usuarios saben en todo momento el contenido que hay en cada de las carpetas de la PC.

Entre estos hay tipos hay: **no compatibles, variables y estáticos**.

2.8.5. Primero pasos en Linux

Para realizar una Buena instalación y rendimiento de cualquier sistema operativo de Linux debemos de realizar estos pasos:

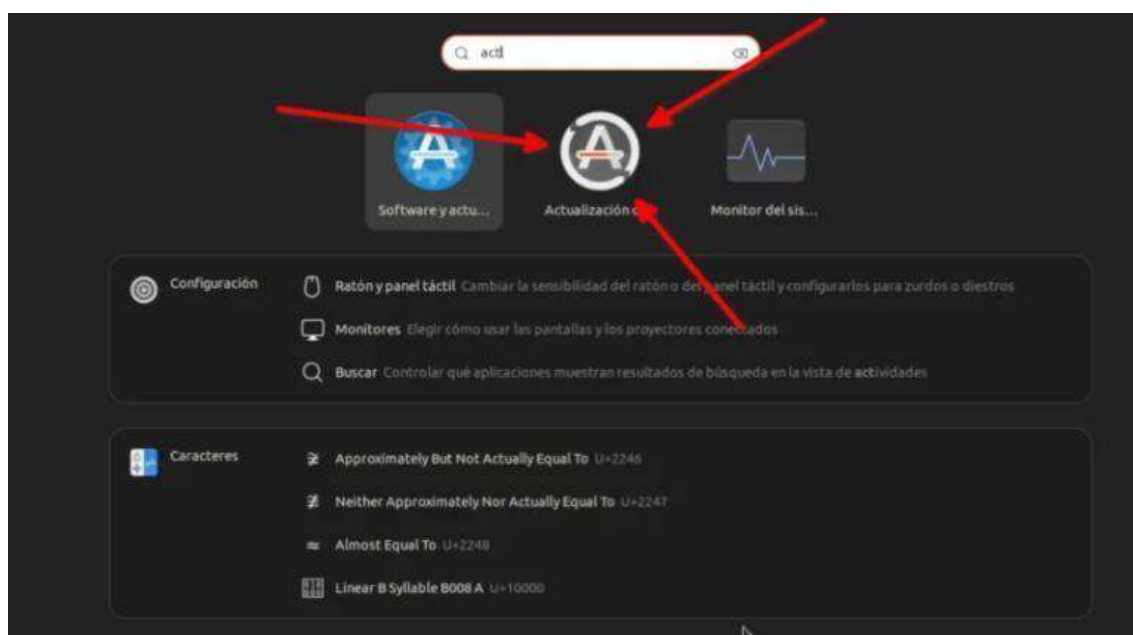
1. **Drivers actualizados.**
2. **Ver e instalar nuevas aplicaciones,**



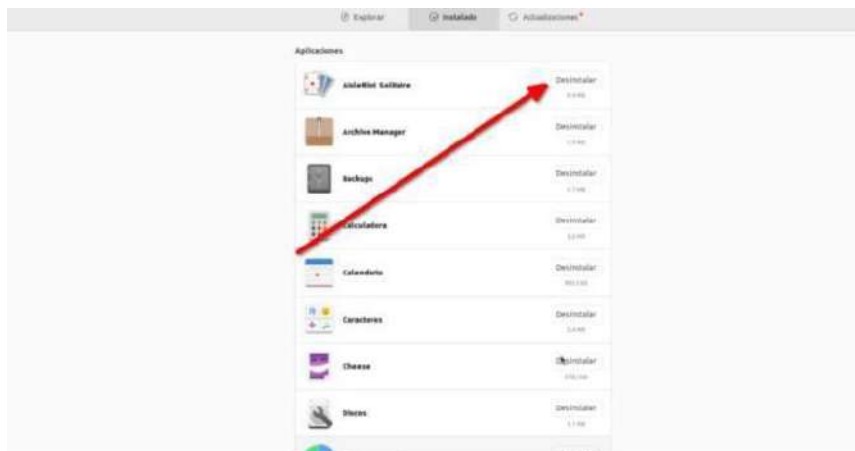
Fuente:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.hostinger.es%2Ftutoriales%2Fque-es-ubuntu&psig=A0vVaw3fjGkH7LWHR7Ce33QVp6kG&ust=1681225937518000&source=images&cd=vfe&ved=0CBEQj RxqFwoTCiCQsunMn_4CFQAAAAAdAAAAABAE

1. Actualizar el sistema.



2. Instalar o eliminar aplicaciones.



Instalar códecs y drivers que son parte de cada sistema operativo para su correcto funcionamiento.

3. Personalizar la interfaz.

```
1 | sudo apt install mate
```

Para instalar el entorno Cinnamon (Linux Mint) escribiremos lo siguiente:

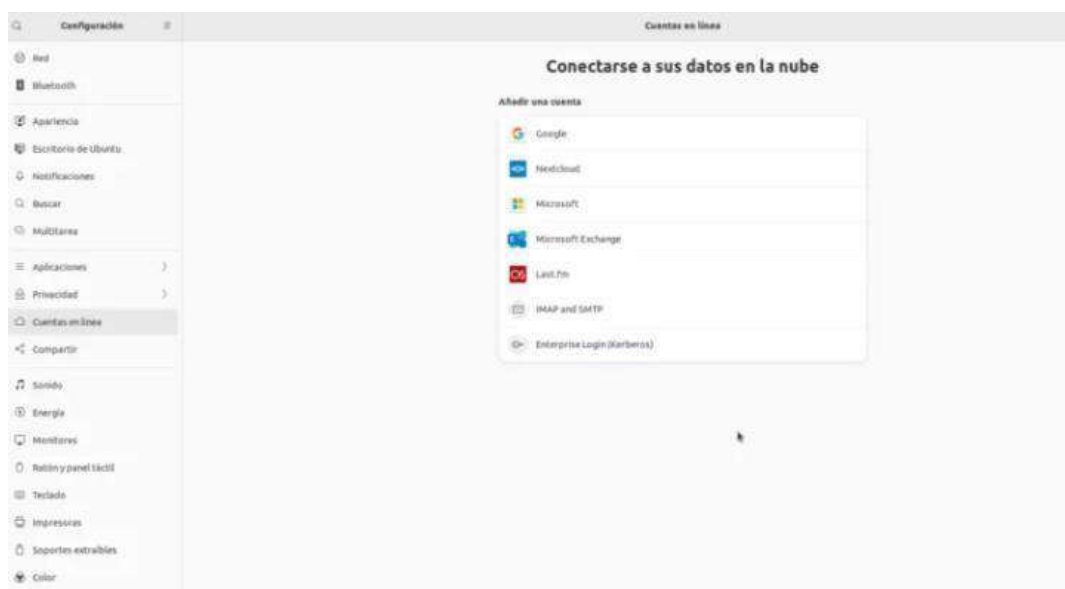
```
1 | sudo apt install cinnamon
```

Y para Plasma, lo siguiente:

```
1 | sudo apt install kde-plasma-desktop
```

4. Configuración avanzada.

En este paso podemos empezar iniciando sesión o creando una cuneta para nuestro sistema operativo.



Fuente: <https://ubunlog.com/wp-content/uploads/2015/12/Cuentas-en-linea-840x461.webp>

2.8.7. Contenido de los directorios

Como parte de los contenidos se creo un estándar de jerarquía, estos nacieron en 1994 por Linux Foundation.

Filesystem Hierarchy Standard que es el sistema de archivos en sistemas de Linux y otros derivados de UNIX.



bin	Binarios de usuario
boot	Ejecutables y archivos requeridos para el arranque
dev	Archivos de información de todos los volúmenes
etc	Archivos de configuración del sistema y de aplicaciones
home	Directorio personal con las carpetas de usuario
lib	Bibliotecas necesarias para la ejecución de binarios
media	Directorio de montaje de volúmenes extraíbles
opt	Ficheros de aplicaciones externas que no se integran en /usr
proc	Ficheros de información de procesos
root	Directorio personal de superusuario
sbin	Binarios de sistema
srv	Archivos relativos a servidores web, FTP, etc.
sys	Archivos virtuales con información de eventos del sistema
tmp	Directorio de ficheros temporales
usr	Archivos de programas y aplicaciones instaladas
var	Archivos de variables, logs, emails de los usuarios del sistema, etc.

Fuente: <https://computernewagedotcom.files.wordpress.com/2018/08/linux-jerarquia-directorios.png>

1.1 Crear directorio en GNU/Linux

Por: Jose Gabriel Ramos Perez

la creación de directorios en GNU/Linux. crea nuevos directorios también conocidos como carpetas. Para crear un nuevo directorio en GNU/Linux, debemos pasar el nombre de este como argumentos para el comando mkdir. Su uso simple, **mkdir prueba**

1.1.1 Copiando Archivos

El comando cp copia un archivo a otro. También puede copiar varios ficheros en un directorio determinado. cp [Origen] [Destino] esto significa que **cp texto.txt /home/usuario/carpeta_de_destino/**

1.1.2 Moviendo Archivos

Con el comando mv podemos mover un archivo de una ubicación a otra, también se puede utilizar para cambiar el nombre de un fichero. mv [parámetros] [origen] [destino] esto es **mv lista-de-la-compra.txt /mnt/usb/cosas/**

1.1.3 Borrar archivos y directorio

Introduce el comando "rm -r" abreviatura de remove y el nombre del directorio para eliminar toda la carpeta y su contenido. Usa el comando rmdir o **rm -d** para eliminar directorios vacíos. Usa el comando **rm -r** para eliminar directorios no vacíos. borrar un archivo con el comando rm

rm archivo.txt y para directorios **rm -r Directorio**



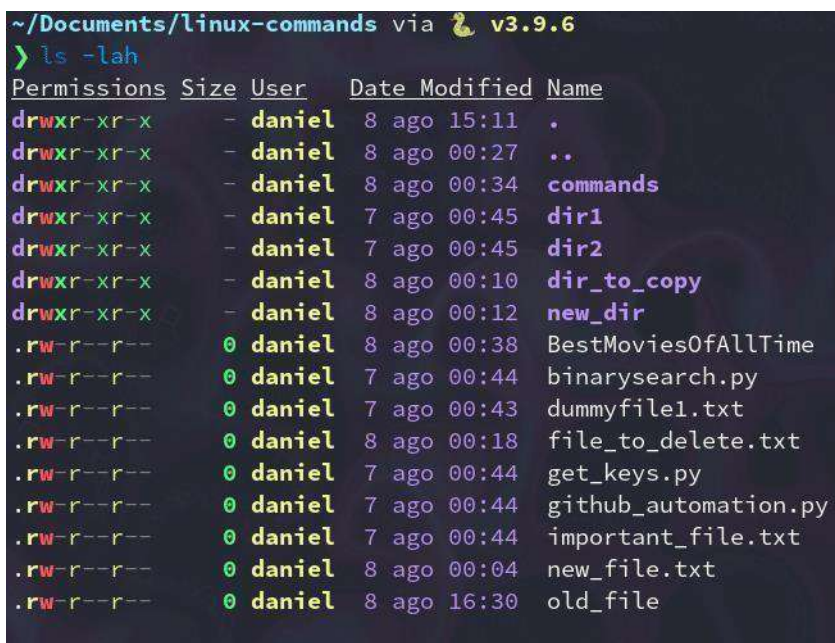
Comandos basicos de linux

Fuente <https://www.elmundogeek.com.mx/wp-content/uploads/2020/03/Comandos-basicos-de-linux.jpg>

1.1.4 Ver contenido del directorio

El comando ls se usa para ver el contenido de un directorio. Por defecto, este comando mostrará el contenido de tu directorio de trabajo actual.

ls/inicio/nombredeusuario/Documentos



Fuente: <https://kinsta.com/wp-content/uploads/2021/08/new-date.png>

1.1 GNU/LINUX- Ayuda en Linea

Por: Fernando Raxon

La ayuda en línea de GNU/Linux es una herramienta muy útil para los usuarios que buscan información y soluciones para resolver problemas relacionados con el sistema operativo GNU/Linux. La ayuda en línea puede ser proporcionada por diferentes fuentes, como la documentación oficial, los foros de la comunidad, los sitios web de ayuda, las listas de correo, entre otros.

1.1.1 Fuentes de Informacion

Una de las principales fuentes de ayuda en línea es la documentación oficial del sistema operativo, que puede incluir manuales, guías de referencia, páginas de manual y páginas de información del sistema. Estos recursos proporcionan información detallada sobre los comandos, programas y herramientas del sistema, así como sobre cómo configurar y personalizar el sistema.

Otra fuente importante de ayuda en línea son los foros y comunidades de usuarios de GNU/Linux. Estos foros permiten a los usuarios compartir información y conocimientos, hacer preguntas y obtener respuestas de otros usuarios experimentados. Algunos de los foros y comunidades más populares incluyen Ubuntu Forums, LinuxQuestions.org y Redditi/linux.

Los sitios web de ayuda y las listas de correo también son fuentes útiles de información y soporte para los usuarios de GNU/Linux. Estos sitios web proporcionan tutoriales, guías paso a paso y respuestas a preguntas comunes sobre GNU/Linux. Las listas de correo son un medio para que los usuarios hagan preguntas y obtengan respuestas de otros usuarios y desarrolladores.

1.1.2 Sumario de Comandos básicos de GNU/LINUX

- ls: muestra una lista de archivos y directorios en el directorio actual.
- cd: cambia el directorio actual a otro directorio.
- mkdir: crea un nuevo directorio en el directorio actual.
- rm: elimina archivos o directorios.
- cp: copia archivos o directorios de un lugar a otro.
- mv: mueve archivos o directorios de un lugar a otro.
- cat: muestra el contenido de un archivo en la pantalla.
- grep: busca una cadena de texto en uno o más archivos.
- chmod: cambia los permisos de acceso a archivos y directorios.
- sudo: permite a los usuarios ejecutar comandos con privilegios de superusuario.
- top: muestra información sobre los procesos en ejecución en el sistema.
- ps: muestra información sobre los procesos en ejecución en el sistema.
- df: muestra información sobre el espacio en disco disponible en el sistema.
- du: muestra información sobre el uso del espacio en disco por parte de los archivos y directorios.
- tar: crea o extrae archivos comprimidos.
- wget: descarga archivos desde Internet.
- ssh: permite conectarse a un servidor remoto de forma segura.
- ping: verifica la conectividad de red con una dirección IP o un nombre de dominio.
- ifconfig: muestra información sobre las interfaces de red en el sistema.
- netstat: muestra información sobre las conexiones de red en el sistema.

1.1.3 Operaciones sobre directorios

- Crear un directorio: se puede crear un nuevo directorio utilizando el comando "mkdir". Por ejemplo, si se desea crear un directorio llamado "nuevo_directorio", se puede ejecutar el comando siguiente:
mkdir nuevo_directorio
- Eliminar un directorio: para eliminar un directorio, se utiliza el comando "rmdir". Este comando solo se puede utilizar si el directorio está vacío. Por ejemplo, para eliminar el directorio "nuevo_directorio", se puede ejecutar el comando siguiente: **rmdir nuevo_directorio**
- Si el directorio no está vacío, se puede utilizar el comando "rm" con la opción "-r" para eliminarlo de forma recursiva. Por ejemplo, para eliminar el directorio "nuevo_directorio" y todo su contenido, se puede ejecutar el comando siguiente:
rm -r nuevo_directorio
- Cambiar el directorio actual: se puede cambiar el directorio actual utilizando el comando "cd". Por ejemplo, para cambiar al directorio "nuevo_directorio", se puede ejecutar el comando siguiente:
cd nuevo_directorio

1.1.4 Operaciones sobre Archivos

- Crear archivos: Para crear un archivo en GNU/Linux, se puede utilizar el comando touch. Por ejemplo, para crear un archivo llamado "archivo.txt", se puede escribir en la terminal: touch archivo.txt.
- Copiar archivos: Para copiar un archivo en GNU/Linux, se puede utilizar el comando cp. Por ejemplo, para copiar un archivo llamado "archivo.txt" a una carpeta llamada "carpeta", se puede escribir en la terminal: cp archivo.txt carpeta/.
- Mover archivos: Para mover un archivo en GNU/Linux, se puede utilizar el comando mv. Por ejemplo, para mover un archivo llamado "archivo.txt" a una carpeta llamada "carpeta", se puede escribir en la terminal: mv archivo.txt carpeta/.
- Eliminar archivos: Para eliminar un archivo en GNU/Linux, se puede utilizar el comando rm. Por ejemplo, para eliminar un archivo llamado "archivo.txt", se puede escribir en la terminal: rm archivo.txt.
- Ver el contenido de un archivo: Para ver el contenido de un archivo en GNU/Linux, se puede utilizar el comando cat. Por ejemplo, para ver el contenido de un archivo llamado "archivo.txt", se puede escribir en la terminal: cat archivo.txt.

1.1 Gnu/Linux

Gabriel Rivas

Es un núcleo mayormente libre semejante al núcleo de Unix.⁴ Es uno de los principales ejemplos de software libre y de código abierto.

1.1.1 Caracteres "Comodin"

Son símbolos o caracteres especiales que representan otros caracteres. Puede usarlos con cualquier con cualquier comando LS o RM.

1.1.2 Shell de Linux

Es el término usado en informática para referirse al intérprete de comandos de los sistemas operativos basados en Unix y similares, como GNU/Linux, y que es su interfaz de usuario tradicional. Mediante las instrucciones que aporta el intérprete, el usuario puede comunicarse con el núcleo y por extensión, ejecutar dichas órdenes, así como herramientas que le permiten controlar el funcionamiento de la computadora.

1.1.3 Comunicación entre p

Cuando escriba un mandato en el indicador y pulse la tecla Intro, el shell evaluará el mandato e intentará ejecutarlo. Según las instrucciones del mandato, el shell grabará la salida del mandato en la pantalla o lo dirigirá a la salida. A continuación, restaura el indicador de mandatos para que escriba otro mandato.

1.1.4 Entrada y Salida Estándar

Los comandos en Linux tienen una entrada estándar (stdin 0) y dos salidas estándar: una para la salida normal del comando (stdout 1) y otra para la salida de los mensajes de error (stderr 2), por defecto la entrada como las salidas se muestran en la terminal del usuario, sin embargo al usar un operador de redirección

- Listar: si queremos que esta salida se almacene en el fichero directorios_app, en su lugar usaremos el operador de redirección \> que permitirá almacenar en el archivo el resultado del comando ls, si el fichero no existe este comando creará el fichero y si existe se sobrescribirá su contenido. \$ ls > directorios_app
- Cat: Si nuestra intención no es sobrescribir el contenido del fichero directorios_app sino agregar la nueva información al final del archivo, entonces usaremos el operador de redirección >>. \$ ls >> directorios_app

1.1.5 Redirigiendo la Entrada y Salida

Ahora ya puedes combinar tanto entrada como salida en una misma línea de comando. Esto, quizá no resulte tan intuitivo como lo que has visto anteriormente, al menos a mi así me resulta.

Por ejemplo, puedes utilizar el comando `rev` para darle la vuelta al contenido de un archivo de texto. Así por ejemplo, en el caso de que tengas el archivo `hola.txt`, que contenga `hola`, y utilices `rev hola.txt` te devolviera por pantalla `aloh`.

Ahora vas a redirigir entrada y salida. La entrada será `hola.txt` y la salida será `hola_al_reves.txt`. Así la línea de comando será: **`rev < hola.txt > hola_al_reves.txt`**

1.1.6 Redirigiendo no Destructiv

Podemos hacer una redirección no destructiva de la salida estándar 1 utilizando el operador `>>`. Este tipo de redirección es no destructiva: si el archivo no existe lo crea, y si existe añadirá la salida al final del archivo, en lugar de sobrescribirlo. Por ejemplo, para añadir el archivo `file2` al final de `file1` haremos:

```
$ cat file2 >> file1
```

También se puede aplicar la redirección no destructiva a la salida estándar 2, la salida de errores. Por ejemplo, para añadir los mensajes de error producidos al compilar `prueba.c` en segundo plano al archivo `errores` haremos:

```
$ gcc prueba.c 2>> errores &
```

1.1.7 Permisos de Archivos

Sus comandos para los archivos Linux son:

- para indicar que el usuario no cuenta con permisos.
- `r` significa que cuentas con el permiso de lectura (read).
- `w` es el permiso de escritura (write).
- `x` se refiere el permiso de ejecución.

1.1 Gnu/Linux SHELL

1.2 Tipos de permisos

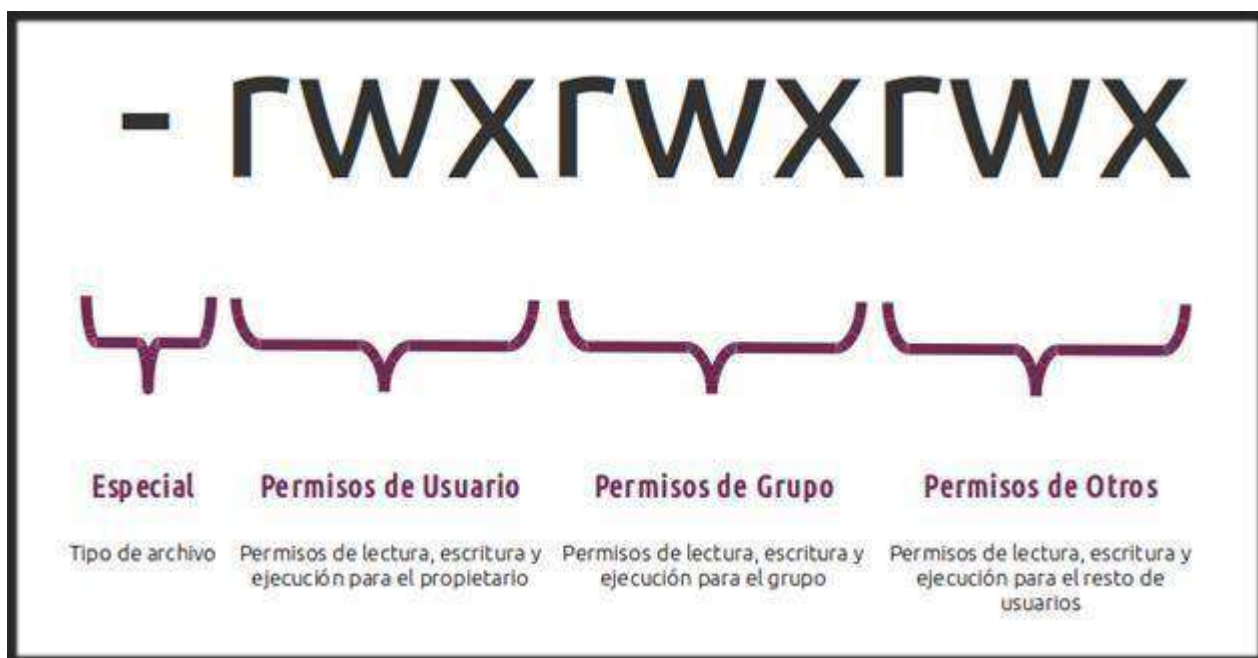
Por: Ariel Rodriguez

Los permisos son uno de los aspectos más importantes de Linux (en realidad, de todos los sistemas basados en Unix). Estos se utilizan para varios fines, pero sirven principalmente para proteger el sistema y los archivos de los usuarios y por ello el comando CHMOD nos permite modificar cualquier permiso.

Manipular los permisos es una actividad interesante, pero compleja al mismo tiempo. Pero tal complejidad no debe ser interpretada como una dificultad, sino como una posibilidad de hacer frente a una gran variedad de configuraciones, lo que permite crear diversos tipos de protección a los archivos y directorios.

Solo el super-usuario (root) tiene acciones sin limites en el sistema, justamente por ser el usuario responsable de la configuración, administración y mantenimiento de Linux. Depende de este, por ejemplo, determinar lo que cada usuario puede ejecutar, crear, modificar, etcétera.

Un punto interesante de mencionar es que Linux trata a todos los directorios como archivos, por lo tanto, los permisos se aplican de igual manera para ambos. Dichos permisos pueden ser divididos en cuatro partes para indicar: tipo, propietario, grupo y otros permisos.



Fuente: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.redeszone.net%2Ftutoriales%2Fservidores%2Fcontrolar-permisos-archivos-carpeta-servidor->

[linux%2F&psig=AOvVaw2LAJcHreRENYBZKwT7XSJy&ust=1681169200459000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCNC91rX6nf4CFQAAAAAdAAAAABAE](https://www.google.com/url?sa=i&url=https%3A%2F%2Fubunlog.com%2Fcomo-funcionan-los-permisos-de-archivo-y-directorio-ii%2F&psig=AOvVaw2LAJcHreRENYBZKwT7XSJy&ust=1681169200459000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCNC91rX6nf4CFQAAAAAdAAAAABAE)

1.2.1 Interpretando los permisos de archi.

La primera cosa que necesitas entender es a qué se aplican los permisos de archivos. Efectivamente lo que haces con esto es aplicar permisos a un grupo. Pero ¿qué permisos son y a qué grupos se refieren?

Existen tres tipos de permisos que se pueden aplicar, estos son:

lectura: otorga al grupo el permiso de poder leer el archivo. Se indica con la letra r (inicial de read, que significa leer en inglés)

escritura: otorga al grupo el permiso de poder editar el archivo pudiendo escribir en él. Se indica con la letra w (inicial de write, que significa escribir en inglés)

ejecución: otorga al grupo el permiso de poder ejecutar el archivo. Se indica con la letra x (que viene de la palabra execute, que significa ejecutar en inglés)



Fuente: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fubunlog.com%2Fcomo-funcionan-los-permisos-de-archivo-y-directorio-ii%2F&psig=AOvVaw3JiAZGykcjteCRPHglmHse&ust=1681171309453000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCNDzsqOCnv4CFQAAAAAdAAAAABAE>

por ejemplo, darle a un grupo de usuarios el permiso de leer y escribir en un fichero, pero no la capacidad de poder ejecutarlo (si se trata por ejemplo de un script). O podrías darle los permisos de poder leer y ejecutar un archivo, pero no de poder modificarlo. Incluso puedes darle a un grupo todos los permisos de lectura, escritura y ejecución de un archivo, o ningún permiso quitándoselos todos.

Visto el tema de los tres permisos, veamos ahora el tema de los grupos, del que has estado leyendo. Los grupos a los que nos podremos referir son cuatro:

usuario: el propietario actual del archivo (nos referimos a este grupo con la letra u)

grupo: un grupo de usuarios de un archivo (nos referimos a este grupo con la letra g)

otros: otros usuarios distintos que no se encuentran en el grupo (nos referimos a este grupo con la letra o)

todos: todos los usuarios (nos referimos a este grupo con la letra a de all, que significa todos en inglés)

1.2.2 Programación del Shell

Un Shell Script es un programa que está creado con instrucciones que son ejecutadas por un Shell (CLI o intérprete de comandos) de Unix o Linux. El código no es compilado ni precompilado, se va ejecutando línea por línea efectuando lo que cada instrucción le indica. Necesita un programa que entienda los comandos y estructuras que contiene y esto se suele poner en la primera línea del programa. Por ejemplo `#!/bin/bash` significa que le pasaremos al BASH las líneas del fichero de Script. Dado que el BASH es el intérprete de comandos más famoso de Linux, los Script que se crean para este entorno también se pueden llamar Bash Script.

En general se suele usar la extensión `.sh` para identificar qué contiene el fichero, pero no es en absoluto necesario. Lo que indica a Linux que es un programa es el permiso de ejecución y no la extensión que tenga el fichero.

¿Para qué sirve en Shell Script?

Principalmente sirve para automatizar tareas y para realizar procesos más complejos de los que un solo comando puede efectuar. Aunque los comandos se pueden enlazar mediante tuberías o XARGS, a veces necesitamos tomar decisiones condicionales o recorrer elementos mediante bucles. Aquí es donde necesitamos organizarlo todo en un Shell Script.

El propio sistema de Linux tiene programadas multitud de tareas con sus Script del sistema, desde la rotación de logs, actualización del arranque, gestión de servidores, niveles de ejecución etc.

1.2.3 Administración del Shell

El administrador de sistemas GNU/Linux tiene que enfrentarse diariamente a una gran cantidad de tareas. En general, en la filosofía UNIX no suele haber una única herramienta para cada tarea o una sola manera de hacer las cosas. Lo común es que los sistemas UNIX proporcionen una gran cantidad de herramientas más o menos simples para afrontar las diferentes tareas.

1.2.4 Que es una distribución

es una distribución de software basada en el núcleo Linux, y a menudo, un Sistema de gestión de paquetes que incluye determinados paquetes de software, para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores. Por lo general están compuestas, total o mayoritariamente, de software libre, aunque a menudo incorporan aplicaciones o controladores propietarios.

Además del núcleo Linux, las distribuciones incluyen habitualmente las bibliotecas y herramientas del proyecto GNU y el sistema de ventanas X Window System. Dependiendo del tipo de usuarios a los que la distribución esté dirigida se incluye también otro tipo de software como procesadores de texto, hoja de cálculo, reproductores multimedia, herramientas administrativas, etc.

En el caso de incluir paquetes de código del proyecto GNU, se denominan distribución GNU/Linux.

Existen distribuciones que están soportadas comercialmente, como Fedora (Red Hat), openSUSE (SUSE) y Ubuntu (Canonical Ltd.); distribuciones mantenidas por la comunidad, como Debian, Arch Linux y Gentoo; y distribuciones que no están relacionadas con ninguna empresa o comunidad, como es el caso de Slackware.

1.1 Conectar Java con MySql en Apache NetBeans

Por: Iker Rodolfo Rojas Reyes

Para conectar java y mysql es necesario usar un conector que normalmente se distribuye en un archivo .jar que son las librerías de java. Este código lo vamos poner en el archivo pom.xml que está en la carpeta Project Files del proyecto de Java/NetBeans.

1.1.1 Instalación de MySql Workbench 8

El software MySQL Workbench es un entorno de MySQL gráfico de diseño de bases de datos, servidores, administración y mantenimiento para el sistema MySQL. Además, esta herramienta gráfica fue desarrollada y distribuida por la compañía de desarrollo de nubes locales Oracle Corporation.

1.1.2 Instalación de Wamp o Xamp

XAMPP es más poderoso que Wamp. Además, XAMPP tiene la función SSL de la que carece Wamp. Wamp es la elección perfecta para aplicaciones web nativas, y XAMPP es cuando necesitas un conjunto de funciones avanzadas en un paquete de servidor web.

1.1.3 JDBC con MySQL

JDBC (Java Database Connectivity) es una interfaz de programación que permite que las aplicaciones Java accedan a una base de datos relacional. SuperCHANNEL necesita un controlador JDBC para poder acceder al sistema de base de datos relacional (por ejemplo, SQL Server, Oracle, etc.) donde se almacenan sus datos de origen.



Fuente: <https://www.programacion.com.py/moviles/android/utilizar-mysql-en-android-con-jdbc>



Fuente: <https://blog.facialix.com/crud-con-java-y-mysql-2/>

1.1.4 Formularios de registros

Los formularios son recolectores de información con entrada de datos necesarios y así minimizar la cantidad de campos requeridos, de la misma forma minimizar los campos opcionales. La información debe ser relevante.



Fuente: <https://evilnapsis.com/2015/06/22/sistema-de-registro-y-login-sencillo-con-php-y-mysql/>

1. Consultas en SQL y Visual Basic

1.1 Consulta de Datos

Las consultas a bases de datos son solicitudes o instrucciones que se hacen a una base de datos para obtener información específica de ella. Estas consultas permiten buscar y filtrar datos en una base de datos, y se realizan mediante lenguajes de consulta estructurados (SQL, por ejemplo) que permiten interactuar con la base de datos de forma eficiente y precisa. Las consultas pueden ser muy variadas y complejas, y pueden incluir diferentes operaciones como seleccionar datos, actualizarlos, eliminarlos o combinarlos de diferentes maneras. Por ejemplo, una consulta puede ser tan simple como "seleccionar todos los nombres de clientes de una tabla" o tan compleja como "seleccionar el total de ventas por día en el último trimestre, agrupado por región y ordenado por el importe total". En resumen, las consultas a bases de datos son un elemento fundamental de la gestión y análisis de datos, y permiten obtener información útil y relevante de una base de datos de manera rápida y eficiente.

1.2. Leer Datos

Para realizar una consulta a una base de datos en Visual Basic, puedes utilizar el lenguaje de consulta estructurado SQL (Structured Query Language). Para ello, debes seguir los siguientes pasos:

1.2.1. Conectar a la base de datos: antes de realizar cualquier consulta, debes conectarte a la base de datos utilizando una cadena de conexión válida.

2. Crear un objeto de comando: después de conectarte a la base de datos, debes crear un objeto de comando (Command) que te permita ejecutar consultas en la base de datos.

1.2.3. Escribir la consulta: utilizando SQL, escribe la consulta que deseas realizar. Por ejemplo, para seleccionar todos los registros de una tabla, puedes escribir: `SELECT * FROM nombre_de_tabla;`

1.2.4. Ejecutar la consulta: una vez que has escrito la consulta, debes ejecutarla utilizando el objeto de comando que has creado en el paso 2. Puedes hacerlo utilizando el método `ExecuteReader` del objeto de comando.

1.2.5. Procesar los resultados: finalmente, debes procesar los resultados obtenidos de la consulta. Para ello, puedes utilizar el objeto `DataReader` que devuelve el método `ExecuteReader`, o bien crear un objeto `DataTable` y llenarlo con los resultados utilizando el método `Fill` del objeto `DataAdapter`.

1.1 Baja y modificación de datos de una tabla de MySQL

Por: Gabriela Santos

Para modificar utilizaremos el comando UPDATE que va seguido del nombre de la tabla que deseas actualizar, luego la cláusula SET con los campos y su nuevo valor (separados por coma), y por último la cláusula WHERE, para aplicar la modificación de estos campos sólo en los registros que cumplan con las condiciones definidas.

Y para eliminar utilizaremos el comando DELETE que nos permite eliminar registros de forma física en nuestras tablas, es decir, borrará el registro de la tabla completamente

1.1.1 Generar comentarios en Código

El servidor MySQL soporta tres estilos de comentario, desde un carácter '#' hasta el fin de la línea. Desde una secuencia '-- ' hasta el final de la línea. Nótese que el estilo '-- ' (doble guión) requiere que luego del último guión haya por lo menos un espacio en blanco (espacio, tabulación nueva línea, etc.). Esta sintaxis difiere ligeramente de la sintaxis de comentarios SQL estándar, como se trata en Sección 1.7.5.7, "Empezar un comentario con '--". Desde una secuencia '/*' hasta la próxima secuencia '*/'. La secuencia de cierre no necesita estar en la misma línea, lo que permite tener comentarios que abarquen múltiples líneas.

La sintaxis de comentarios descrita se aplica a la forma en que el servidor mysqld procesa las sentencias SQL. El programa cliente mysql también realiza algún tipo de proceso de sentencias antes de enviarlas al servidor. (Por ejemplo, para determinar donde acaba cada sentencia en una línea que contiene varias de ellas).

En MySQL 5.0, la única limitación en la forma en que mysql procesa los comentarios /* ... */ es que un signo admirativo o exclamativo utilizado con este estilo de comentario marca porciones de sentencias SQL de ejecución condicional. Esto se aplica cuando se ejecuta mysql interactivamente y cuando se colocan comandos en un fichero y se emplea mysql para procesar el fichero con mysql < nom_fich.

1.1.2 Estructura de datos

Una base de datos bien diseñada brinda a los usuarios acceso a información fundamental. Al seguir los principios de esta página, puedes diseñar una base de datos que funcione bien y se adapte a tus necesidades futuras. Explicaremos los aspectos básicos sobre el diseño de una base de datos y cómo perfeccionarlo para obtener resultados óptimos.

El sistema mediante el cual se organizan los datos en la memoria de la aplicación que se está programando. Existen muchas formas de organizar los datos en la memoria. Por ejemplo, un array es una colección de elementos de memoria en el que los datos se almacenan de forma secuencial.

1.1.3 Listas, Colas y Pilas

La pila es una secuencia de elementos del mismo tipo en la que el acceso a la misma se realiza por un único lugar denominado cima:

Vemos como el acceso a los elementos de la pila se realiza siempre sobre un único extremo. Las operaciones que caracterizan la pila son las de introducir un nuevo elemento sobre la cima (push) y la de extraer el elemento situado en la cima (pop). Una forma de ver esta estructura de datos es como una pila de libros en la que sólo se puede coger el libro que está en la cima o apilar más libros sobre la misma, pero los libros que sostienen la pila no son accesibles pues de otro modo todo se desmoronaría.

En una Cola los elementos se añaden desde la parte de atrás o la parte final de la cola, sin embargo la información se extrae desde el frente, es decir, los elementos que se añadieron primero serán los primeros en salir, esto se conoce como estructura FIFO (First In First Out).

Los elementos de la cola se añaden y se eliminan de tal manera que el primero en entrar es el primero en salir. La adición de elementos se realiza a través de una operación llamada encolar (enqueue), mientras que la eliminación se denomina desencolar (dequeue). La operación de encolar inserta elementos por un extremo de la cola, mientras que la de desencolar los elimina por el otro.

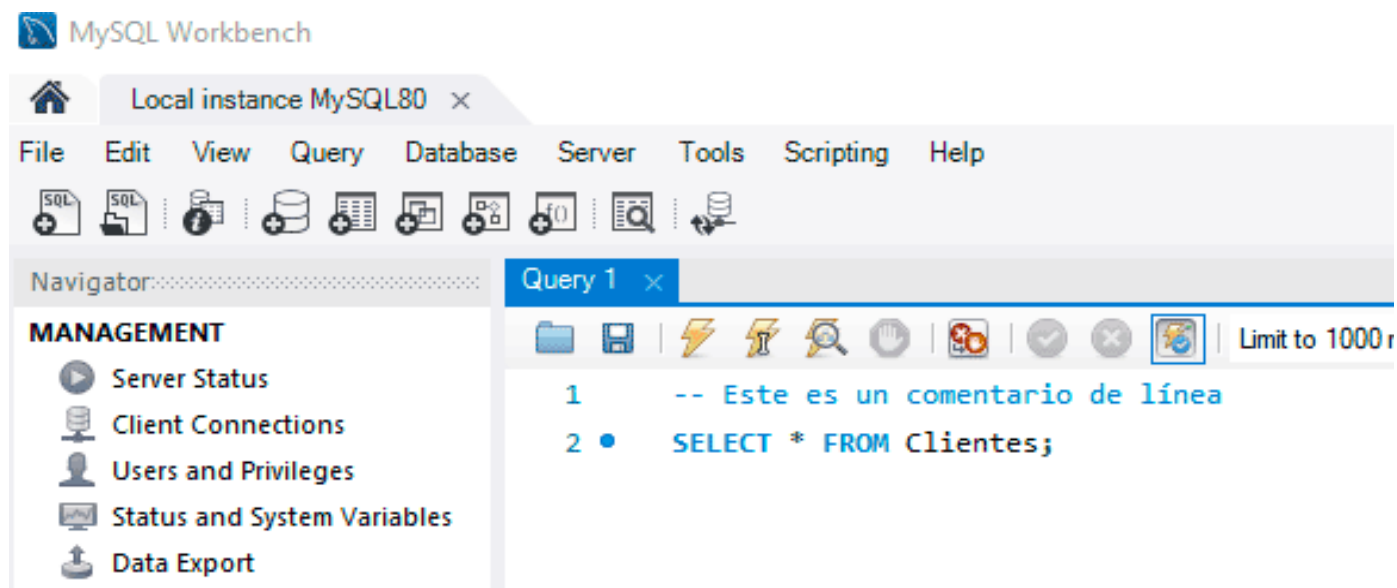


Imagen: Ejemplo de comentarios en SQL

Fuente: <https://programacionfacil.org/blog/wp-content/uploads/2023/01/xcomentarios-sql.png.pagespeed.ic.5YnAQmdHxI.png>

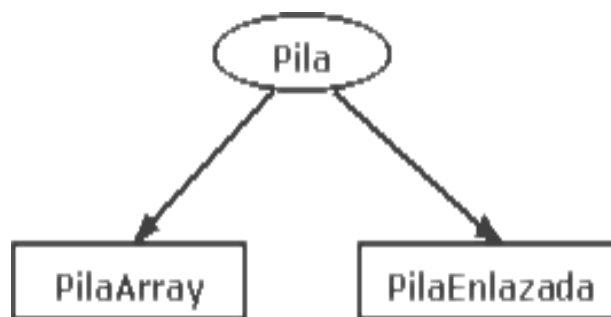


Imagen: Ejemplo de pilas

Fuente: <http://www.ciberaula.com/imagenes/edpila.gif>

1. Conectar Java con MySQL en Apache NetBeans

Por: Dilan Sarceño

Para conectar java y mysql es necesario usar un conector que normalmente se distribuye en un archivo .jar que son las librerías de java.

```
1. <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
2. <dependency>
3.   <groupId>mysql</groupId>
4.   <artifactId>mysql-connector-java</artifactId>
5.   <version>8.0.28</version>
6. </dependency>
```

Código: Este código lo vamos poner en el archivo pom.xml que está en la carpeta Project Files del proyecto de Java/NetBeans

1. Listas tipo Pila

Una Pila en palabras sencillas es un lugar donde se almacenan datos, al igual que en un Array, pero una Pila tiene una filosofía de entrada y salida de datos, esta filosofía es la LIFO (Last In First Out, en español, último en entrar, primero en salir). Esta estructura de datos tiene muchas aplicaciones debido a su simplicidad.

Esta estructura en lenguaje de programación Java, aunque debemos tomar en cuenta que esta clase Pila ya existe en el API de Java, con el nombre Stack (Pila en inglés) en el paquete java.util pero de esta clase hablamos en otro Post. Ahora implementemos desde cero, que nos ayude a entender su funcionamiento y además porque nos lo piden en materias de estructura de datos. La versión que veremos está muy bien estructurada y con tipo de datos primitivos, por eso utilizaremos un Array para almacenar los datos así no dinámica (porque no la dimensión de un vector la declaramos en cuando se la crea, es de dimensión fija).

Esta pila tiene 4 elementos, para la implementación de la clase haremos uso una variable entera tope que se encargara de decirnos en que posición del Array está el elemento de la cima, en este caso tope=3 porque en el Array donde ingresamos los datos desde la posición 0

2. Listas tipo Cola

En las colas (también conocidas como queues, buffers o FIFO – first in, first out) los primeros elementos que entran son los primeros en ser procesados.

Esto lo podemos ver en las filas en los bancos los primeros que se forman son los primeros en ser atendidos (generalmente). Algunos ejemplos de su uso son en la escritura de números grandes (sección 6.3) y en las búsquedas en amplitud (como en la sección 9.4).



Fuente: <https://i.ytimg.com/vi/oduPQpqnI0/hqdefault.jpg>

A screenshot of a web browser window titled "CRUD". The main content is a "SIMPLE CRUD APPLICATION" interface. It features three input fields: "First Name", "Last Name", and "Reg Number". Below these fields are three buttons: "Save", "Update", and "Delete". To the right of the input fields is a table with three columns: "First Name", "Last Name", and "Id Number". The table contains four rows of data.

First Name	Last Name	Id Number
bruno	den	3554877
harry	den	13852000
christine	gray	08552100
jason	park	55423554

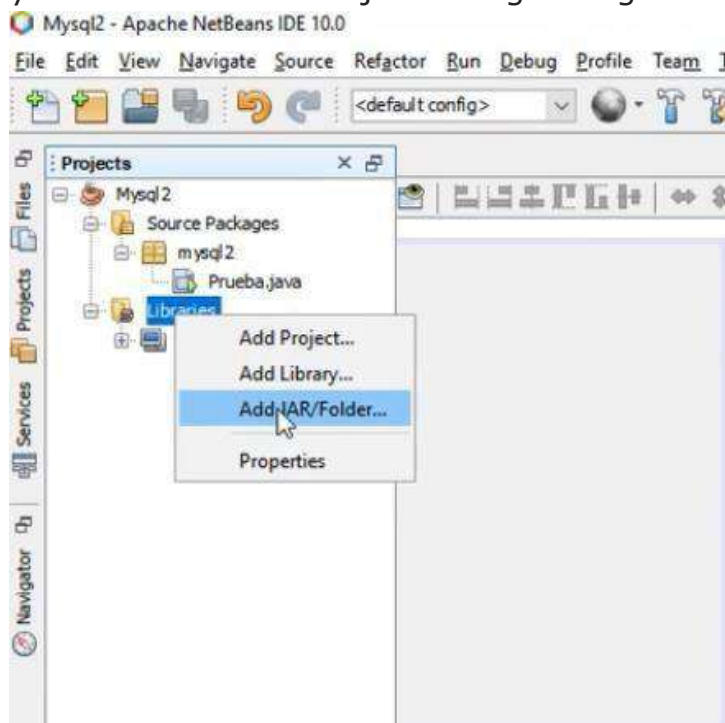
Ejemplo: Un CRUD puede ser un ejemplo en la vida cotidiana ya que es un sistema utilizado en muchas cosas ya que ingresa, edita y elimina datos del sistema.

1.1 Conectar Java con MySQL en Apache NetBeans

Por: Dulce María Shuyá Baten

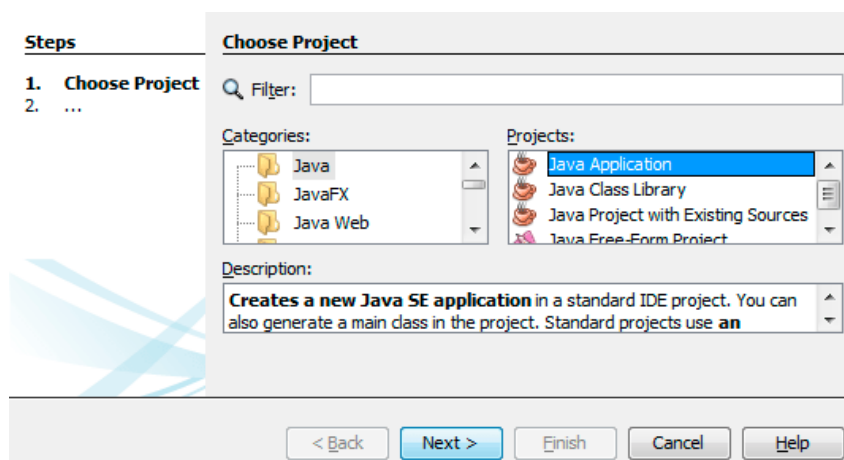
Para conectar Java con MySQL en Apache NetBeans, sigue estos pasos:

- Descarga e instala el conector de MySQL para Java. El conector se puede descargar desde la página oficial de MySQL. Descomprime el archivo descargado y coloca el archivo .jar en algún lugar accesible para tu proyecto.



<http://minicursos.xyz/wp-content/uploads/2020/05/mysqlconnectoradd.png>

- En Apache NetBeans, crea un nuevo proyecto Java o abre uno existente.



<https://assets2-programarya-com.juandmegon.com/img/Cursos/Java/Entornos/Proyectos/proyecto-de-aplicacion-java.png>

- Asegúrate de tener el paquete JDBC instalado. Si no lo tienes, ve al menú "Herramientas" -> "Plugins" y busca el paquete "JDBC". Instálalo y reinicia Apache NetBeans.

- Crea una nueva clase en tu proyecto y escribe el código necesario para conectar con la base de datos de MySQL.

```

package conectar;

import java.sql.Connection;
import java.sql.DriverManager;

public class Conectar {
    public static final String URL = "jdbc:mysql://localhost:3306/pruebajava";
    public static final String USER = "root";
    public static final String CLAVE = "";

    public Connection getConexion(){
        Connection con = null;
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = (Connection) DriverManager.getConnection(URL, USER, CLAVE);
        }catch(Exception e){
            System.out.println("Error: " + e.getMessage());
        }
        return con;
    }
}
    
```

<https://lh3.googleusercontent.com/2tS058jiQrkPmTK57xic73S4AUv-5ViITSZUovHDJO2ZTlpPigZDHIpgceys7i-M3wRp790pCEDL6eRY0LMaFDo8kESWfgsEExjuko>

- Ejecuta la clase y verifica que la conexión se establece correctamente.



<https://www.cablenaranja.com/wp-content/uploads/2021/03/Como-Conectar-Java-Con-MySQL-En-Netbeans-CableNaranja-35.png>

- Con estos pasos, ya deberías ser capaz de conectar Java con MySQL en Apache NetBeans. Recuerda que es importante manejar las excepciones y cerrar la conexión adecuadamente para evitar problemas de seguridad y rendimiento en tu aplicación.

1.1.1 Listas genéricas

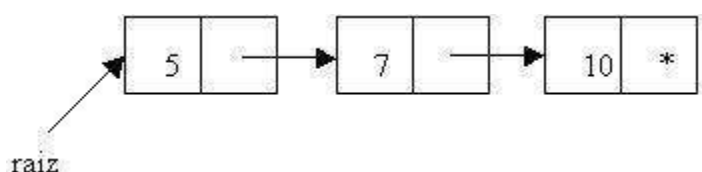
Para trabajar con listas genéricas en MySQL y NetBeans, sigue los siguientes pasos:

- Crea una base de datos en MySQL y una tabla con los campos que desees utilizar para almacenar los datos.
- Abre NetBeans y crea un nuevo proyecto Java.
- Crea una nueva clase en el proyecto y define los campos que desees utilizar en la tabla de la base de datos. Agrega las anotaciones necesarias para que puedas mapear los campos de la clase con los campos de la tabla en la base de datos.
- Crea una nueva clase DAO (Data Access Object) que permita realizar las operaciones de inserción, actualización, eliminación y consulta de datos en la base de datos. En esta clase, crea los métodos que necesites para realizar las operaciones mencionadas.
- En la clase DAO, utiliza una lista genérica para almacenar los datos que se recuperan de la base de datos. Puedes utilizar la clase ArrayList de Java para esto.
- Implementa el código necesario para conectarse a la base de datos y realizar las operaciones CRUD (Create, Read, Update, Delete) en la lista genérica.
- Utiliza la clase DAO en el resto de tu proyecto para manipular los datos que se almacenan en la lista genérica.

1.1.2 Listas genéricas Ordenadas

Para trabajar con listas genéricas ordenadas en MySQL y NetBeans, puedes seguir estos pasos:

- Crea una tabla en MySQL con los campos que necesitas para tu lista. Por ejemplo, si quieres crear una lista de productos, podrías crear una tabla llamada "productos" con los campos "id", "nombre", "precio", etc.
- Crea una clase en NetBeans que represente los datos de la tabla. Esta clase debe tener los mismos campos que la tabla de MySQL, y debe tener métodos para obtener y establecer los valores de cada campo. Además, la clase debe implementar la interfaz Comparable para que puedas ordenar los objetos de la clase según un criterio específico.
- Crea un ArrayList en NetBeans para almacenar los objetos de la clase que representan los datos de la tabla. Puedes llenar este ArrayList con los datos que recuperes de la tabla en MySQL utilizando una conexión JDBC.
- Ordena la lista de objetos utilizando el método sort de la clase Collections, que utiliza la implementación del método compareTo que definiste en la clase Producto.
- Con estos pasos, ya deberías ser capaz de trabajar con listas genéricas ordenadas en MySQL y NetBeans.



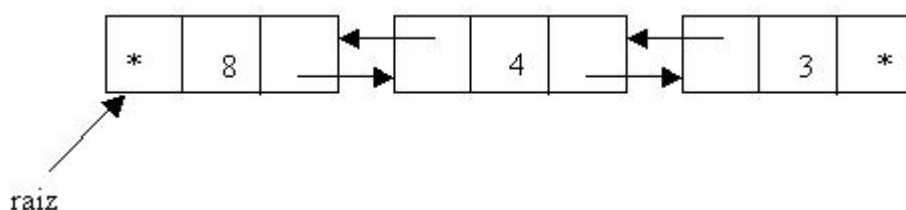
<https://www.tutorialesprogramacionya.com/javaya/imagentema/foto111.jpg>

1.1.3 Listas genéricas doblemente encadenadas

Las listas genéricas doblemente encadenadas son estructuras de datos que permiten almacenar una colección de elementos en un orden específico y proporcionan una forma eficiente de insertar, eliminar y buscar elementos.

Para implementar una lista doblemente encadenada en MySQL y NetBeans, se pueden seguir los siguientes pasos:

- Crear una tabla en MySQL con las columnas necesarias para almacenar los elementos de la lista.
- Crear una clase Java que representa un nodo de la lista y que tenga atributos para almacenar el elemento, el nodo anterior y el siguiente nodo.
- Crear una clase Java que represente la lista y que tenga atributos para almacenar el tamaño de la lista, el primer nodo y el último nodo.
- Implementar los métodos necesarios en la clase lista para insertar, eliminar y buscar elementos en la lista.
- Crear una conexión a la base de datos MySQL desde el código Java y utilizar sentencias SQL para interactuar con la tabla de la lista.
- Utilizar la lista en el código Java para almacenar y manipular los elementos.



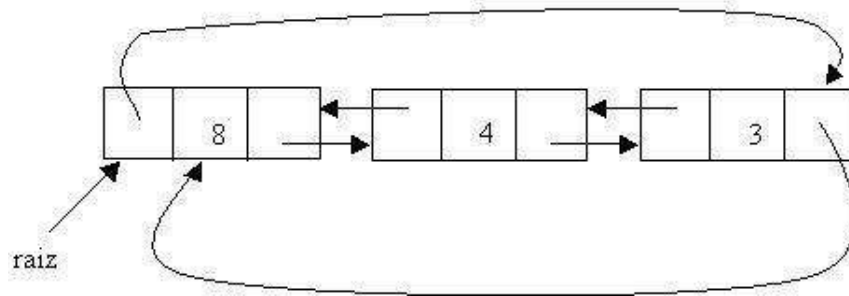
<https://www.tutorialesprogramacionya.com/javaya/imagentema/foto113.jpg>

1.1.4 Listas genéricas circulares

Las listas genéricas circulares son una estructura de datos en la que cada elemento tiene un puntero al siguiente elemento, y el último elemento apunta al primero, formando así un bucle.

Para implementar una lista genérica circular en MySQL y NetBeans, puedes seguir estos pasos:

- Crea una tabla en MySQL con los campos necesarios para almacenar los elementos de la lista. Puedes usar un campo para almacenar el valor del elemento y otro campo para almacenar el puntero al siguiente elemento.
- Crea una clase en NetBeans para representar los elementos de la lista. Esta clase debe tener un campo para almacenar el valor del elemento y otro campo para almacenar una referencia al siguiente elemento.
- Implementa los métodos necesarios para manipular la lista. Algunos métodos básicos incluyen: agregar un elemento al final de la lista, eliminar un elemento de la lista, buscar un elemento en la lista, y mostrar todos los elementos de la lista.
- Utiliza sentencias SQL para interactuar con la tabla de MySQL y realizar las operaciones necesarias en la lista.
- Utiliza un bucle para recorrer la lista, empezando por el primer elemento y terminando en el último elemento. Asegúrate de que el último elemento apunte de nuevo al primer elemento para cerrar el bucle.



<https://www.tutorialesprogramacionya.com/javaya/imagentema/foto115.jpg>

1.1.5 Implementación Java

Para implementar Java con MySQL en NetBeans, sigue estos pasos:

- Descarga e instala el conector de MySQL para Java.
- En NetBeans, crea un nuevo proyecto Java.
- Asegúrate de tener el paquete JDBC instalado.
- Crea una nueva clase y escribe el código para conectar con la base de datos de MySQL. Utiliza la clase "Connection" del paquete "java.sql" y el método "DriverManager.getConnection()". Especifica la URL de la base de datos, el nombre de usuario y la contraseña.
- Escribe el código para ejecutar una consulta en la base de datos utilizando un objeto "Statement" y un objeto "ResultSet". Cierra los objetos y la conexión cuando hayas terminado.
- Ejecuta la aplicación y verifica que la conexión y la consulta se ejecutan correctamente.

1.1 Conectar Java con MySQL en Apache NetBeans.

Por: Maryori Omeydi Sipac Sesam.

Java puede conectarse a múltiples bases de datos, con la que mejor se lleva es con Oracle, sin embargo, en ocasiones puede ser que los que veníamos a la web, al tener la necesidad de hacer una aplicación de escritorio, nos sintamos más acostumbrados a MySQL. El primer lugar, para realizar este ejemplo debemos tener instalado tres cosas en nuestro sistema: La máquina virtual (para Java, claro está) MySQL y NetBeans.

1.1.1 Árboles Binarios.

Los árboles binarios son estructuras de datos muy similares a las listas doblemente enlazadas, en el sentido que tienen dos punteros que apuntan a otros elementos, pero no tienen una estructura lógica de tipo lineal o secuencial como aquellas, sino ramificada. Tienen aspecto de árbol, de ahí su nombre. Un árbol binario es una estructura de datos no lineal en la que cada nodo puede apuntar a uno o máximo a dos nodos. También se suele dar una definición recursiva que indica que es una estructura compuesta por un dato y dos árboles. Esto son definiciones simples. Este tipo de árbol se caracteriza porque tienen un vértice principal y de él se desprende dos ramas. La rama izquierda y la rama derecha a las que también se les conoce como subárboles. Una representación gráfica de la estructura general de un árbol binario se puede visualizar en la imagen1 que presente a continuación.



Fuente: <https://hhmosquera.files.wordpress.com/2014/05/arbol.png>

1.1.2 Teoría de Grafos.

Con esta teoría se busca representar de forma visual conjuntos de datos abstractos en formas de nodos o vértices y la unión o relaciones que estas pueden tener con otros nodos a través de aristas. Gracias a esta teoría se han podido lograr grandes avances en el análisis de amplios volúmenes de data.

1.1.3 Tipos de Árboles Binarios.

Existen cuatro tipos de árboles binarios:

- Distinto. Se dice que dos árboles binarios son distintos cuando sus estructuras son diferentes: ejemplo:



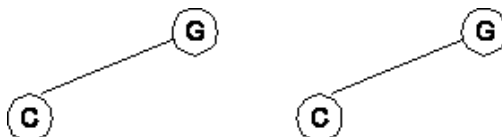
Fuente: <https://www.monografias.com/trabajos92/arboles-binario/image003.gif>

- Similares. Dos árboles binarios son similares cuando sus estructuras son idénticas, pero la información que contiene sus nodos es diferente. Ejemplo:



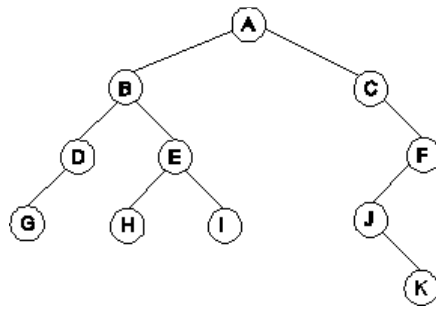
Fuente: <https://www.monografias.com/trabajos92/arboles-binario/image004.gif>

- Equivalentes. Son aquellos árboles que son similares y que además los nodos contienen la misma información. Ejemplo:



Fuente: <https://www.monografias.com/trabajos92/arboles-binario/image005.gif>

- Completos. Son aquellos árboles en los que todos sus nodos excepto los del último nivel, tiene dos hijos; el subárbol izquierdo y el subárbol derecho. Ejemplo:



Fuente: <https://www.monografias.com/trabajos92/arboles-binario/image006.gif>

1.14 Recorridos en Profundidad.

Un Recorrido en profundidad (en inglés DFS o Depth First Search) es un algoritmo que permite recorrer todos los nodos de un grafo. Es una generalización del recorrido preorden de un árbol. La estrategia consiste en partir de un vértice determinado v y a partir de allí, cuando se visita un nuevo vértice, explorar cada camino que salga de él. Hasta que no se haya finalizado de explorar uno de los caminos no se comienza con el siguiente. Un camino deja de explorarse cuando se llega a un vértice ya visitado. Si existían vértices no alcanzables desde v el recorrido queda incompleto; entonces, se debe seleccionar algún vértice como nuevo vértice de partida, y repetir el proceso.



Google Classroom



3.17. Recorridos en pre orden

Por: Edgar Soc

El recorrido en pre orden, también llamado orden previo consiste en recorrer en primer lugar la raíz y luego cada uno de los hijos en orden previo. Hace un recorrido desde hijos izquierdos a derechos, se van imprimiendo los nodos que aparecen, pasando a hermanos más derechos cuando llegamos a una hoja.

En un recorrido en pre orden de un árbol A, la raíz de A es visitada primero y entonces los subárboles enraizados con sus hijos son recorridos recursivamente.

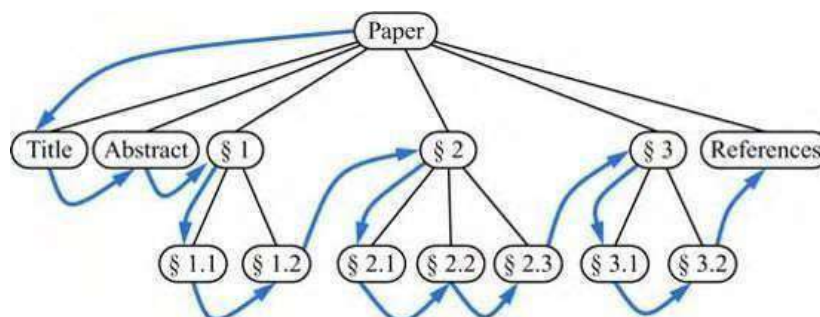


Imagen: Ejemplo recorrido en pre orden

Fuente: <https://f08ff9f7-a-62cb3a1a-s-sites.googlegroups.com/site/edatosubo>

3.18. recorridos en post orden

El recorrido en post orden, también llamado orden posterior consiste en recorrer en primer lugar cada uno de los hijos y por último la raíz.

El algoritmo de recorrido en post orden puede ser visto como el opuesto al recorrido en pre orden, porque este recursivamente recorre los subárboles enraizados en los hijos de la raíz primero, y después visita la raíz.

Al igual que en el recorrido en pre orden se emplea el algoritmo para resolver un problema particular especializando una acción asociada con la "visita" de un nodo v. Si el árbol esta ordenado, se hacen llamadas recursivas para los hijos de un nodo v de acuerdo a su orden indicado.

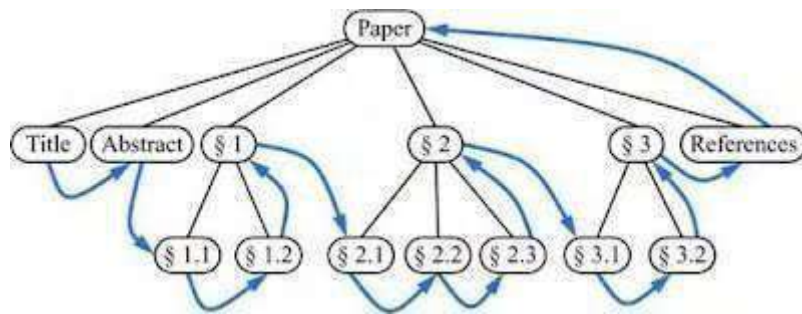


Imagen: Ejemplo recorrido en post orden

Fuente: https://sites.google.com/site/edatosubo/_/rsrc/1434465472376/4-estructura-de-datos-dinamicas/4-arboles/arbol3.jpg?height=152&width=400

3.19. Recorridos en in orden

En el orden in orden se recorre de la siguiente manera: subárbol izquierdo, raíz, subárbol derecho.

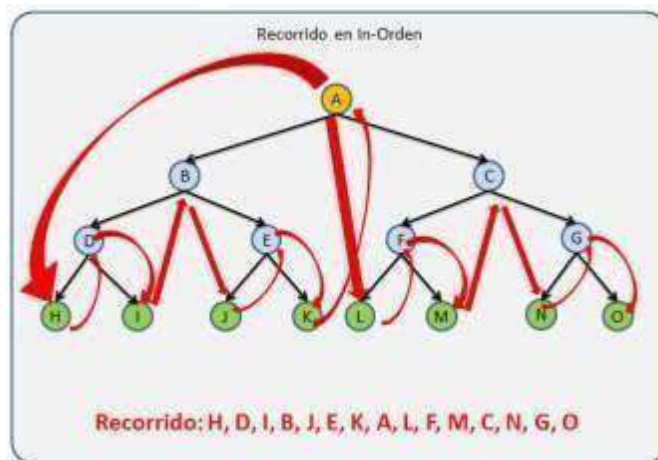


Imagen: Ejemplo recorrido in orden

Fuente: <https://www.oscarblancarteblog.com/wp-content/uploads/2014/08/inorden.png>

3.20. Recursividad

Se llama recursividad a un proceso mediante el que una función se llama a sí misma de forma repetida, hasta que se satisface alguna determinada condición. El proceso se utiliza para computaciones repetidas en las que cada acción se determina mediante un resultado anterior.



Imagen: Ejemplo de recursividad

Fuente: <https://www.campusmvp.es/recursos/image.axd?picture=Recursividad-Portada.png>

3.21. Problemas donde conviene aplicar la recursividad

Se puede usar en toda situación en la cual la solución pueda ser expresada como una secuencia de movimientos, pasos o transformaciones gobernadas por un conjunto de reglas no ambiguas.

3.21.1. Factible de utilizar recursividad

- Para simplificar el código.
- Cuando la estructura de datos es recursiva ejemplo: árboles.

La recursividad se debe usar cuando sea realmente necesaria, es decir, cuando no exista una solución iterativa simple.

4.1 Instalación de XAMPP

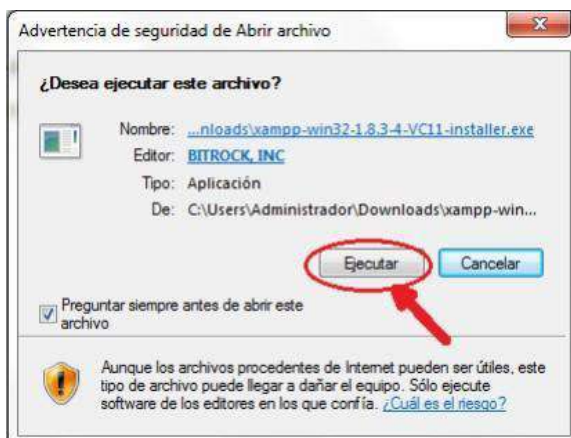
Por: Juan Alejandro Tiney Garcia

XAMPP es una pila AMP que significa multiplataforma. Contiene algunas herramientas de software administrativo adicionales como PHPMyAdmin (para acceso a la base de datos), servidor FTP FileZilla, servidor de correo Mercury y servidor JSP Tomcat.

- Ejecutar el archivo.exe



- Al hacer doble click sobre el ejecutable descargado, comenzará la instalación en sí.



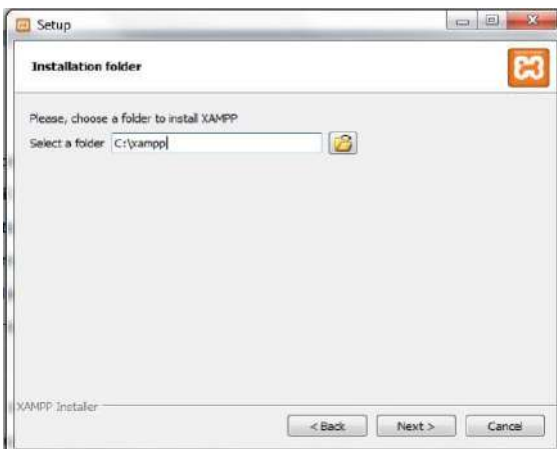
- Al pulsar sobre ejecutar, comienza la instalación, apareciendo:



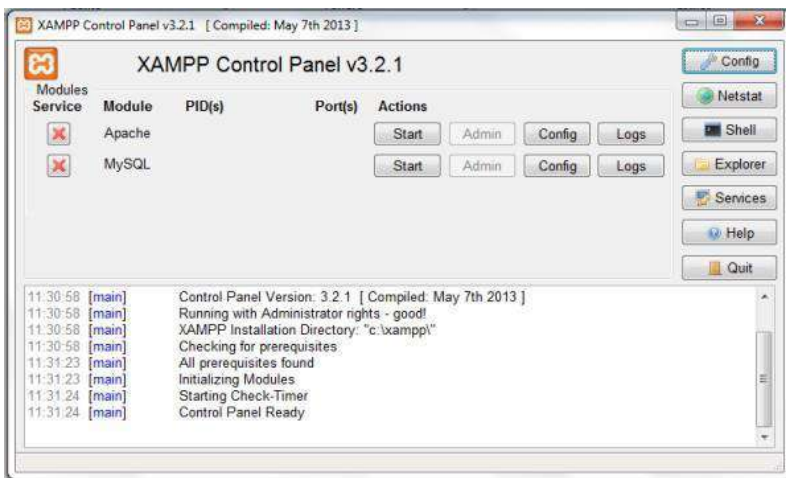
Debemos pulsar sobre "Next" para continuar.



debemos seleccionar la carpeta donde deseamos sea instalado XAMPP.



- Por último, iniciamos el xampp.



<https://biblus.us.es/bibing/proyectos/abreproy/12197/fichero/Manual+de+instalacion.pdf>

4.2 Instalación de sublime text

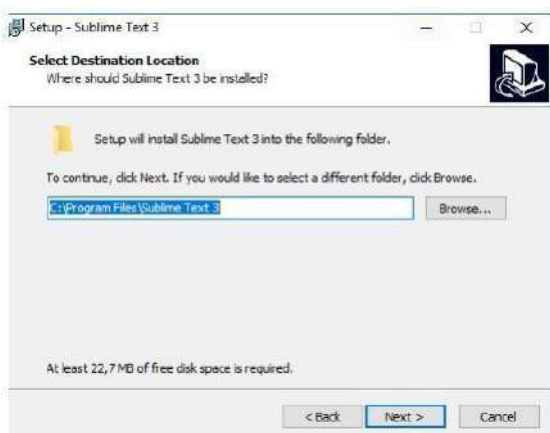
- Bajar ejecutable desde la página del centro de software.
- Click sobre el ejecutable.



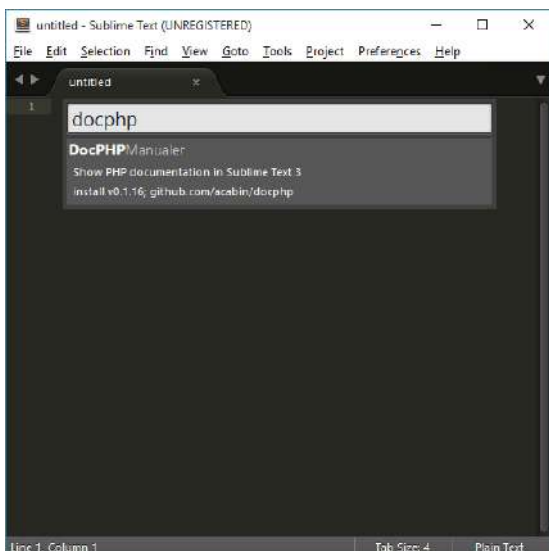
- Ejecutar y seguir el asistente de instalación hasta finalizar.



- Aceptar las condiciones y dejando por default o cambiándolo a preferencia.



- Y por último finalizar la instalación y proceder a usarlo.



<https://dc722rjp2zu8.cloudfront.net/media/django-summernote/2016-02-15/dc5ac6c7-e2c7-4491-a110-8ad00a39c44a.png>

4.3 XML

El lenguaje de marcado extensible (XML) nos permite definir y almacenar datos de forma compatible. XML permite el intercambio de información entre los sistemas de computación, como sitios web, bases de datos entre otros.



<https://rubiko.mx/wp-content/uploads/2021/09/archivo-xml.s.jpeg>

4.4 ¿Qué es XML?

XML, son las siglas en inglés de Extensible Markup Language, es un metalenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium, cuya función es mostrar la información de manera estructurada, documentos, datos, configuración, facturas, entre otros.



<https://static.vecteezy.com/system/resources/previews/000/367/045/original/xml-vector-icon.jpg>

4.5 Principales Componentes.

Un archivo de lenguaje de marcado extensible (XML) es un documento basado en texto que se puede guardar con la extensión .xml. Puede

escribir XML de forma similar a otros archivos de texto. Para crear o editar un archivo XML, puede usar cualquiera de las siguientes opciones:

- Editores de texto como Notepad o Notepad++
- Editores XML en línea
- Navegadores web

Cualquier archivo XML incluye los siguientes componentes.

- Documento XML
- Declaración XML
- Elementos XML
- Atributos XML
- Contenido XML

<https://aws.amazon.com/es/what-is/xml/>

4.6 Tipos de documentos XML

Documentos bien formados

Un documento XML debe estar bien formado, es decir debe cumplir las reglas de sintaxis de la recomendación XML. Para que un documento esté bien formado, al menos debe cumplir los siguientes puntos:

- El documento contiene únicamente caracteres Unicode válidos.
- Hay un elemento raíz que contiene al resto de elementos.
- Los nombres de los elementos y de sus atributos no contienen espacios.
- El primer carácter de un nombre de elemento o de atributo puede ser una letra, dos puntos (:) o subrayado (_).
- El resto de caracteres pueden ser también números, guiones (-) o puntos (.).

□ Documentos validos

Un documento XML bien formado puede ser válido. Para ser válido, un documento XML debe:

- incluir una referencia a una gramática,
- incluir únicamente elementos y atributos definidos en la gramática,
- cumplir las reglas gramaticales definidas en la gramática.

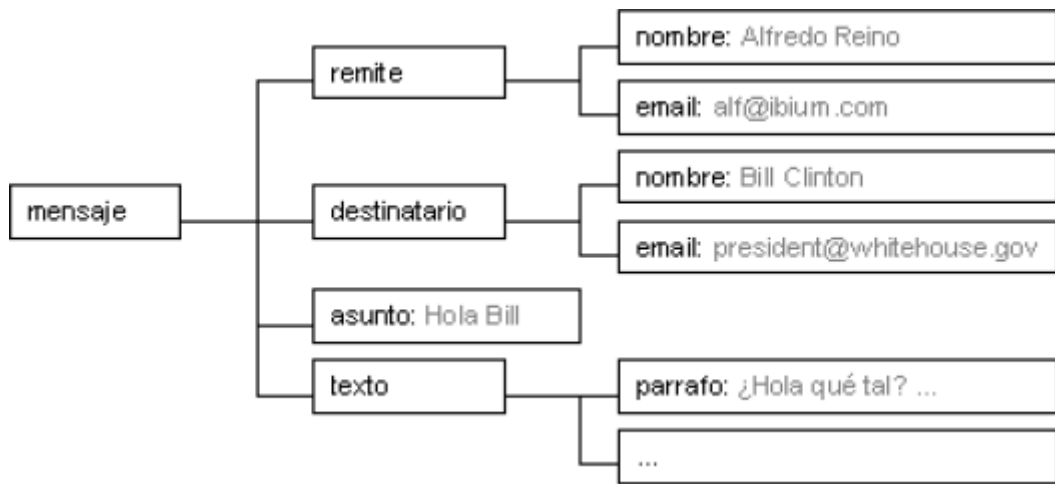
Existen varias formas de definir una gramática para documentos XML, las más empleadas son:

- DTD (Document Type Definition = Definición de Tipo de Documento). Es el modelo más antiguo, heredado del SGML.
- XML Schema. Es un modelo creado por el W3C como sucesor de las DTDs.
- Relax NG. Es un modelo creado por OASIS, más sencillo que XML Schema.

<https://www.mclibre.org/consultar/xml/lecciones/xml-conceptos-basicos.html>

4.7 Estructura de documentos XML

Aunque a primera vista, un documento XML puede parecer similar a HTML, hay una diferencia principal. Un documento XML contiene datos que se autodefinen, exclusivamente. Un documento HTML contiene datos mal definidos, mezclados con elementos de formato. En XML se separa el contenido de la presentación de forma total.



<https://desarrolloweb.com/articulos/intro-xml-y-estructura.html>

1.1 Creando entornos y diseños web con XML

Por: Jeremy Tobar

XML define las asignaciones entre las rutas de URL y los servlets que controlan las solicitudes con esas rutas. El servidor web usa esta configuración para identificar el servlet que maneja una solicitud determinada y llamar al método de clase que corresponde al método de la solicitud. Detrás del diseño y el texto de los sitios web, siempre hay un lenguaje propio y uno de ellos es el XML. Este acrónimo significa Extensible Markup Language, que es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.

1.1.1 Elementos de XML

Una etiqueta es una marca con clase que delimita una región en los lenguajes basados en XML. También puede referirse a un conjunto de juegos informáticos interactivos que se añade a un elemento de los datos para identificarlo. Es un lenguaje marcado extensible que también permite definir y almacenar de forma compatible. XML admite el intercambio de información entre sistemas de computación, como sitios web, bases de datos y aplicaciones de terceros. Las reglas predefinidas facilitan la transmisión de datos con archivos XML a través de cualquier red, ya que el destinatario puede usar esas reglas para leer los datos de forma precisa y eficiente.

1.1.2 Etiquetas de XML

Una etiqueta es una marca con clase que delimita una región en los lenguajes basados en XML. También puede referirse a un conjunto de juegos informáticos interactivos que se añade a un elemento de los datos para identificarlo. Es un lenguaje marcado extensible que también permite definir y almacenar de forma compatible. XML admite el intercambio de información entre sistemas de computación, como sitios web, bases de datos y aplicaciones de terceros. Las reglas predefinidas facilitan la transmisión de datos con archivos XML a través de cualquier red, ya que el destinatario puede usar esas reglas para leer los datos de forma precisa y eficiente.

1.1.3 Atributos de XML

Los atributos se utilizan para distinguir entre elementos del mismo nombre, cuando no se desea crear un nuevo elemento de cada situación. Por lo tanto, el uso de un atributo puede agregar un poco más en detalle a la hora de diferenciar dos o más elementos similares. Existen tipos de atributos: string Type (Toma cualquier cadena literal como un valor), TokenizedType (La validez y las restricciones que se muestran en la gramática se aplican después de que el valor de atributo se normaliza) y EnumeratedType (tiene una lista de valores predefinidos en su declaración y de la cual, se debe asignar un valor).

1.1.4 Nombres de XML

DB2 XPath utiliza espacios de nombres XML para evitar conflictos de denominación. Un espacio de nombres XML es una colección de nombres identificada por un URI de espacio de nombres. Los espacios de nombres proporcionan una forma de calificar los nombres utilizados para los elementos, atributos, tipos de datos y funciones en XPath. En XPath, los nombres se denominan QNames (nombres calificados) y se ajustan a la sintaxis definida en Espacios de nombres en XML de la recomendación W3C. Un QName consta de un prefijo de espacio de nombres opcional y un nombre local. El prefijo de espacio de nombres, si está presente, está enlazado a un URI y proporciona una forma abreviada del URI.

XML

XML, siglas en inglés de eXtensible Markup Language, traducido como 'Lenguaje de Marcado Extensible' o 'Lenguaje de Marcas Extensible'

Fuente: https://www.vectorlogo.zone/logos/w3c_xml/w3c_xml-ar21.png

1.1.5 Referencias y entidades de documentos XML

En XML existen algunos caracteres que son especiales por su significado y, para escribirlos en un documento XML, se pueden utilizar las referencias a entidades mostradas en la siguiente tabla:

Referencias a entidades en XML		
Carácter	Entidad	Referencia a entidad
< (menor que)	lt (less than)	&lt;
> (mayor que)	gt (greater than)	&gt;
" (comilla doble)	quot (quotation mark)	&quot;
' (comilla simple)	apos (apostrophe)	&apos;
& (ampersand)	amp (ampersand)	&amp;

Imagen: Ejemplo del tamaño solicitado (información) Fuente: <https://www.google.com/search?q=imagen+ejemplo>

1.1.6 Entidades

Una entidad es una declaración que establece un nombre que se va a utilizar en el código XML en lugar de contenido o marcado. Para las entidades hay dos partes. En primer lugar, se debe asociar un nombre al contenido de reemplazo, utilizando una declaración de entidad. Una declaración de entidad se crea mediante la sintaxis `<!ENTITY name "value">` incluida o en una DTD o esquema XML. En segundo lugar, el nombre definido en la declaración de entidad se utiliza posteriormente en el XML. Al utilizarlo en el código XML, se conoce como referencia de entidad. Por ejemplo, en la declaración de entidad siguiente se declara una entidad del nombre publisher que está asociada al contenido de "Microsoft Press".

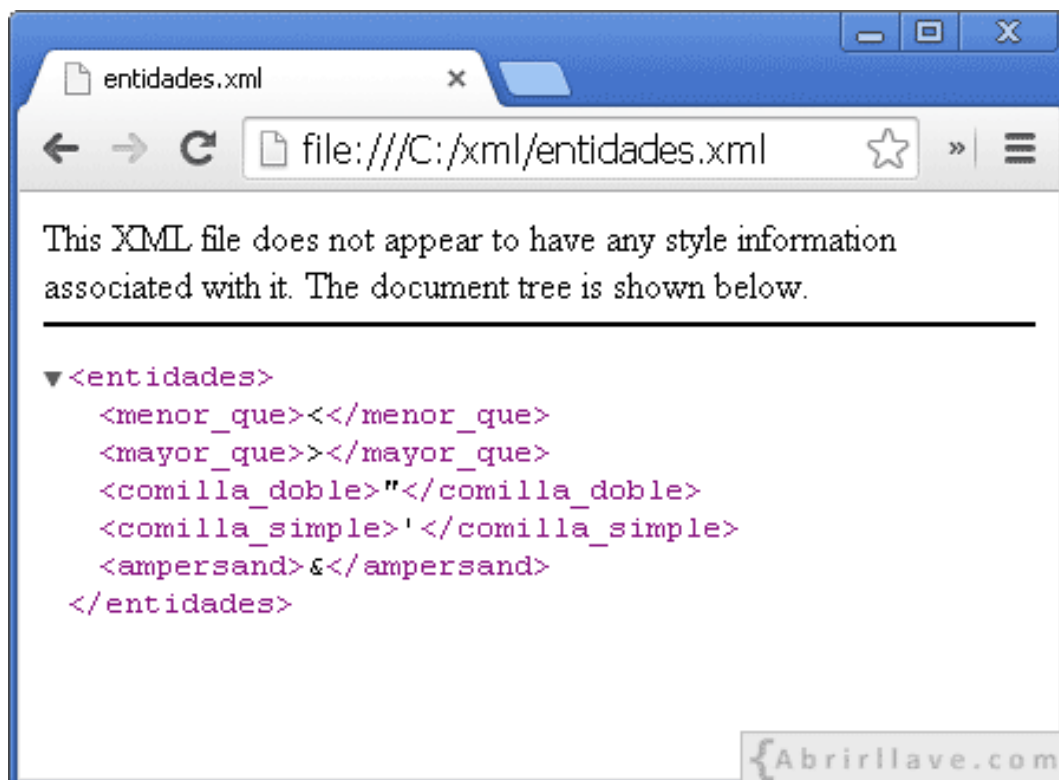


Imagen: Ejemplo del tamaño solicitado (información) Fuente: <https://www.abrirllave.com/xml/images/entidades-xml-chrome.gif>

1.1.7 Creacion de documentos XML

Puede crear un archivo XML nuevo que, a continuación, puede ser editado en el archivo XML. Puede crear un archivo XML nuevo que no está asociado a una DTD o a un archivo de esquema XML, por lo tanto, no hay restricciones en cuanto al contenido que puede contener.

Las instrucciones siguientes se han escrito para la perspectiva Recurso perotambién funcionarán en muchas otras perspectivas.

Para crear un archivo XML nuevo, siga estos pasos:

1. Cree un proyecto para contener el archivo XML si es que todavía no existe uno.
2. En el entorno de trabajo, pulse Archivo > Nuevo > Otro y seleccione XML > XML. Pulse Siguiente.
3. Seleccione el proyecto o la carpeta que albergará el archivo XML.
4. En el campo Nombre de archivo, escriba el nombre del archivo XML y pulse Siguiente. El nombre del archivo XML debe acabar en .xml.
5. Seleccione el botón Crear archivo XML a partir de una plantilla XML y pulse Siguiente.
6. Opcional: Puede elegir ver y editar las plantillas XML que serán utilizadas en el nuevo archivo XML.
7. Pulse Finalizar.

1.1 Título

Por: Diego Gabriel Vasquez Rosales

1.1.1 Comentario en xml

Un comentario comienza con `<!--` y termina con `-->`. Puede agregar notas de texto como de los comentarios entre los personajes. Usted no debe anidar un comentario dentro de la otra.

1.1.2 Declaración de un documento de xml

La declaración de tipo de documento (DTD) de un mensaje XML se representa mediante un elemento de sintaxis de tipo `DocTypeDecl` y mediante sus elementos hijo y elementos inferiores. Éstos contienen la construcción `DOCTYPE`.

1.1.3 Comprobación de documentos de xml

Los documentos XML y objetos de negocio se pueden validar utilizando el servicio de validación.

Además, otros servicios necesitan determinados requisitos mínimos o de lo contrario emiten una excepción de ejecución. Uno de estos servicios es `BOXMLSerializer`.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <persona profesion="centenista">
3    <nombre>Elsa</nombre>
4    <mujer/>
5    <fecha_de_nacimiento>
6      <dia>18</dia>
7      <mes>6</mes>
8      <año>1996</año>
9    </fecha_de_nacimiento>
10   <ciudad>Pamplona</ciudad>
11 </persona>

```

Imagen: Ejemplo del tamaño solicitado (información) Fuente: <https://www.google.com/search?q=imagen+ejemplo>

1.1.4 HTML

Es un *markup language*, lo que significa que está escrito con códigos que puede leer una persona sin que sea necesario compilarlo primero

HTML



Imagen: Ejemplo del tamaño solicitado (información) Fuente: <https://www.google.com/search?q=imagen+ejemplo>

1.1.5 Que es XHTML

Es, básicamente, HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.

1.1.6 Declaracion de un documento de xml

El código HTML está hecho específicamente para diseñar páginas web para su visualización en los navegadores. El XML está pensado únicamente para el transporte y almacenamiento de datos. Aunque es legible para el ser humano, no está pensado para ser visto en el front-end.

1.1.7 Elementos de XHTML

- Elementos estructurales
- Elementos de texto
- Elementos de lista
- Elementos de vinculación
- Elementos de objetos
- Elementos de tabla
- Elementos de formulario
- Elementos de *scripting*

4.0 Creando entornos y diseños web con XML

Por: Eduardo Velásquez

4.22 Atributos XHTML

Atributo	Descripción
id	asigna un identificador único a un elemento.
class	asigna un nombre de clase a un elemento, lo que le permite diseñarlo con CSS.
style	define estilos en línea para un elemento.
title	proporciona información adicional sobre un elemento, como una información sobre herramientas.
lang	especifica el idioma utilizado en el contenido de un elemento.
dir	especifica la dirección del contenido en un elemento, como de izquierda a derecha o de derecha a izquierda.
accesskey	asigna un atajo de teclado a un elemento.
tabindex	especifica el orden en que se accede a los elementos mediante el teclado.
href	especifica la URL de un recurso vinculado.
src	especifica la URL de un archivo externo, como una imagen o un script.

4.23 Documentos XHTML

El lenguaje de marcado XHTML (Lenguaje de marcado de hipertexto extensible) es similar a HTML pero se adhiere a pautas de sintaxis XML más estrictas. Similar a cómo se usa HTML, XHTML también se usa para estructurar contenido en la web, pero ofrece más flexibilidad y es compatible con más herramientas.

Un documento XHTML consta de un conjunto de elementos, que se utiliza para definir la estructura del contenido. Los elementos se encierran en etiquetas, que se encierran entre paréntesis angulares. Por ejemplo, la etiqueta <p> se usa para definir un elemento de párrafo:

```
<p>Esto es un párrafo.</p>
```

Además de los elementos, los documentos XHTML también utilizan atributos, que se utilizan para proporcionar información adicional sobre un elemento. Los atributos se especifican dentro de la etiqueta de apertura de un elemento. Por ejemplo, el atributo de clase se puede usar para especificar un nombre de clase para un elemento:

<p class="elemento">Este párrafo tiene una clase de "elemento".</p>

Los documentos XHTML también deben seguir ciertas reglas para garantizar que estén bien formados y sean válidos. Por ejemplo, todas las etiquetas deben cerrarse correctamente y todos los atributos deben tener un valor especificado. Además, los documentos XHTML deben estar codificados en UTF-8 u otra codificación de caracteres compatible.

En general, XHTML proporciona una forma estandarizada de estructurar y dar formato al contenido de la web, con reglas estrictas para garantizar la interoperabilidad y la compatibilidad con otras tecnologías.



Fuente: <https://cdn-icons-png.flaticon.com/512/28/28780.png>

4.24 CSS

CSS (hojas de estilo en cascada) es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en HTML, XHTML o XML. CSS se utiliza para diseñar la apariencia visual de una página web, incluidos elementos como el diseño, la tipografía, los colores y el espaciado.

CSS funciona aplicando estilos a elementos HTML. Los estilos se definen en un archivo CSS separado o en la sección <head> de un documento HTML usando la etiqueta <style>. Los estilos se definen mediante un conjunto de reglas que especifican cómo se debe diseñar un elemento HTML en particular. Estas reglas constan de uno o más selectores y una o más declaraciones.

Los selectores se utilizan para apuntar a elementos HTML específicos que deben tener estilo. Hay varios tipos de selectores en CSS, incluidos los selectores de elementos, los selectores de clase y los selectores de ID. Por ejemplo, la siguiente regla apuntaría a todos los elementos <h1> en una página:

```
h1 {
  font-size: 16px;color:
  red;
}
```

Las declaraciones se utilizan para definir los estilos que deben aplicarse a los elementos seleccionados. Cada declaración consta de una propiedad y un valor. Por ejemplo, en la regla anterior, la propiedad de tamaño de fuente se establece en 24 px y la propiedad de color se establece en #333.

CSS permite una amplia gama de opciones de estilo, incluida la especificación de estilos de fuente, colores, imágenes de fondo, márgenes, relleno y más. También es compatible con funciones avanzadas, como consultas de medios, que permiten aplicar estilos según el tamaño de la pantalla o la orientación del dispositivo que se utiliza para ver la página.

En general, CSS es una herramienta poderosa para controlar la apariencia visual de las páginas web, lo que permite a los diseñadores y desarrolladores crear interfaces de usuario atractivas y atractivas.



Fuente:

https://upload.wikimedia.org/wikipedia/commons/thumb/d/d5/CSS3_logo_and_wordmark.svg/1452px-CSS3_logo_and_wordmark.svg.png

4.25 Sintaxis y Selectores CSS

Los selectores se utilizan para apuntar a elementos HTML específicos que deben tener estilo. Los selectores pueden basarse en el tipo de elemento, clase, ID, valores de atributos y más. Estos son algunos ejemplos de selectores de CSS de uso común:

Selector de elementos: apunta a todas las instancias de un tipo de elemento HTML específico, como <p> o <h1>.


```
p {
  font-size: 16px;
}
```

Selector de clase: apunta a todas las instancias de un elemento HTML con un nombre de clase específico.

```
.mi-clase {
  color: blue;
}
```

Selector de ID: apunta a un solo elemento HTML con un valor de atributo de ID específico.

```
#mi-id {
  background-color: gray;
}
```

Selector de atributos: apunta a todas las instancias de un elemento HTML con un valor de atributo específico.

```
a[href="https://www.ejemplo.com"] {
  text-decoration: none;
}
```

4.26 Hojas de estilo en CSS

En CSS, las hojas de estilo se refieren a una colección de reglas que definen el estilo visual de un documento HTML. CSS se utiliza para definir el diseño, los colores, las fuentes y otros elementos visuales de una página web.

4.26.1 CSS externo

El CSS externo se refiere a un método para definir las reglas de estilo para una página web en un archivo separado con una extensión `.css`, en lugar de incluir los estilos directamente en el documento HTML. Esto se hace usando el elemento `<link>` en la sección de encabezado del documento HTML, que especifica la ubicación del archivo CSS externo.

```

<html>
  <head>
    <link rel="stylesheet"
      href="style.css">
  </head>
</html>

```

Fuente: <https://img.uxcel.com/practices/external-css-1621258547069/a-1672390170541.jpg>

4.26.2 CSS interno

El CSS interno se refiere a un método para definir reglas de estilo para una página web dentro de la sección de encabezado de un documento HTML utilizando el elemento <style>. En otras palabras, las reglas de estilo se definen dentro del propio archivo HTML, en lugar de en un archivo CSS externo.

```

<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type">
<title>Untitled 1</title>

<style type="text/css">
#video-gallery {
  width:350px;
  padding:12px;
  margin:15px auto;
}
</style>

</head>

```

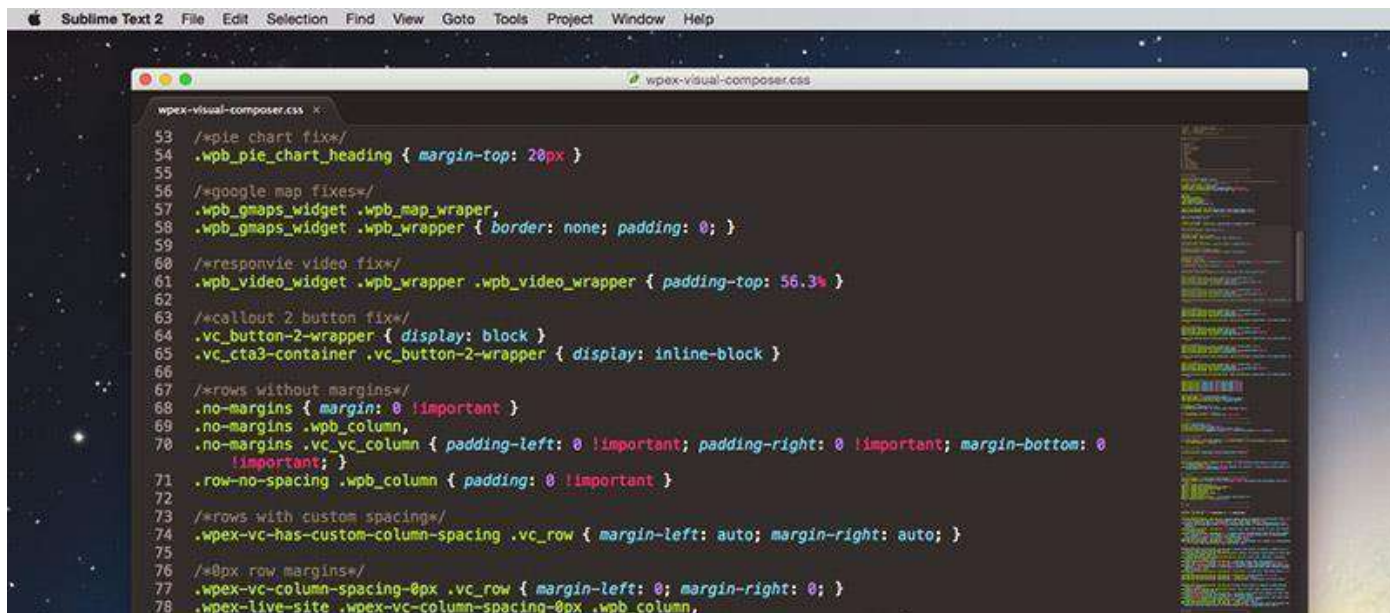
Internal Style:
Style Declaration in the <head> section.

Fuente: <https://i.stack.imgur.com/K59EF.gif>

4.26.3 CSS en línea

El CSS en línea se refiere a un método para definir reglas de estilo para una página web utilizando un editor de código en línea o un entorno de desarrollo integrado (IDE), en lugar de usar un archivo CSS externo o interno.

Un ejemplo de un editor de CSS en línea es CodePen, que permite a los desarrolladores crear y compartir proyectos web utilizando HTML, CSS y JavaScript. En CodePen, el código HTML, CSS y JavaScript se ingresan en paneles separados, con el código CSS ingresado en el panel "CSS".



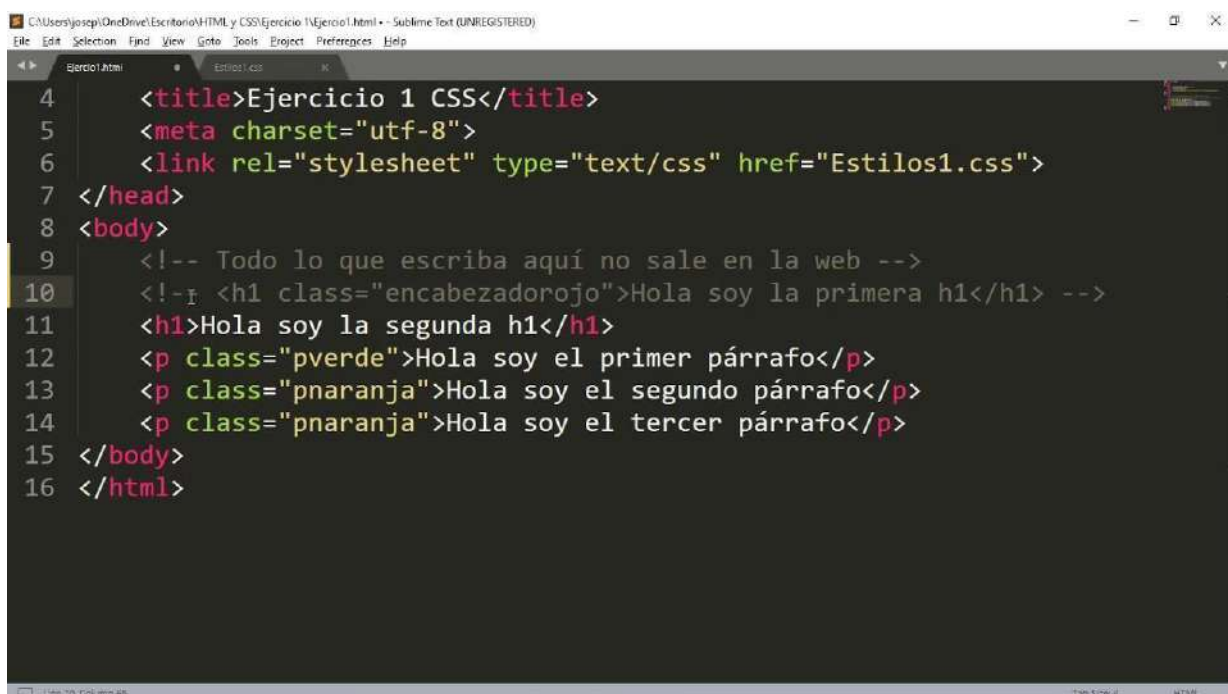
Fuente: <https://wdexplorer.com/wp-content/uploads/learn-css-online.jpg>

4.27 Comentarios en CSS

En CSS, los comentarios se utilizan para agregar texto descriptivo que el navegador web ignora. Los comentarios se pueden usar para explicar el propósito del código, brindar recordatorios o deshabilitar temporalmente un bloque de código sin eliminarlo.

Hay dos formas de agregar comentarios en CSS:

1. Comentarios de una sola línea: los comentarios de una sola línea comienzan con /* y terminan con */. Todo lo que hay entre estos símbolos se trata como un comentario.
2. Comentarios de varias líneas: los comentarios de varias líneas comienzan con /* y terminan con */, y pueden abarcar varias líneas de código.



Fuente: <https://i.ytimg.com/vi/0-fxiqliPAo/maxresdefault.jpg>

4.28 Colores

En el desarrollo web, los colores se utilizan para dar estilo y diseñar páginas web. Hay varias formas de especificar colores en el desarrollo web, incluido el uso de colores con nombre, códigos hexadecimales, valores RGB y valores HSL.

Colores con nombre: los colores con nombre son nombres de color predefinidos que se pueden usar en HTML y CSS. Algunos ejemplos incluyen rojo, verde, azul, negro y blanco. Los colores con nombre son fáciles de usar, pero tienen opciones limitadas.

```
body {
  background-color: red;
}
```

Códigos hexadecimales: Los códigos hexadecimales son códigos de seis dígitos que representan los valores rojo, verde y azul (RGB) de un color en formato hexadecimal. Los códigos hexadecimales se utilizan a menudo en CSS para especificar un color.

```
body {
  background-color: #FF0000; /* Color rojo */
}
```

Valores RGB: los valores RGB representan un color como una combinación de valores rojo, verde y azul. Cada valor varía de 0 a 255. Los valores RGB se usan a menudo en CSS para especificar un color.

```
body {
  background-color: rgb(255, 0, 0); /* Color rojo */
}
```

Valores HSL: los valores HSL representan un color como una combinación de valores de matiz, saturación y luminosidad. El tono es un número entre 0 y 360 que representa la rueda de colores, la saturación es un porcentaje entre 0% y 100% que representa la intensidad del color y la luminosidad es un porcentaje entre 0% y 100% que representa el brillo del color. . Los valores HSL se usan a menudo en CSS para especificar un color.

```
body {
  background-color: hsl(0, 100%, 50%); /* Color rojo */
}
```

4. Creando entornos y diseños web con XML

Por: Ricardo Ventura

4.29. Fondos

Usar la propiedad background-color es una de las opciones para poner un fondo, ya después del atributo nosotros escogemos el color, ya sea por uno predeterminado del sistema (Red, Green, Blue, etc.), también podemos por medio de RGB dando las coordenadas para el color (0,0,0) como también en el sistema hexadecimal

```
body {  
    background-color: #0f1473;  
}
```

Fuente: <https://desarrolloweb.com/articulos/fondoscss#:~:text=Cómo%20aplicar%20fondos%20de%20color%20con%20CSS&text=La%20manera%20más%20sencilla%20de,del%20color%20que%20deseamos%20aplicar.&text=En%20este%20caso%20hemos%20aplicado,afectar%20a%20toda%20la%20página.>

4.30. Márgenes y Bordes

Los márgenes se establecen por medio del atributo margin, ya si queremos medidas más específicas podemos usar las siguientes:

```
margin-top: 10px;  
margin-right: 10px;  
margin-bottom: 10px;  
margin-left: 10px;
```

Fuente: <https://www.lawebera.es/xhtml-css/margen-relleno-bordes-de-capas-con-css-html.php#:~:text=En%20el%20código%20CSS%20el,se%20le%20determina%20el%20margen.>

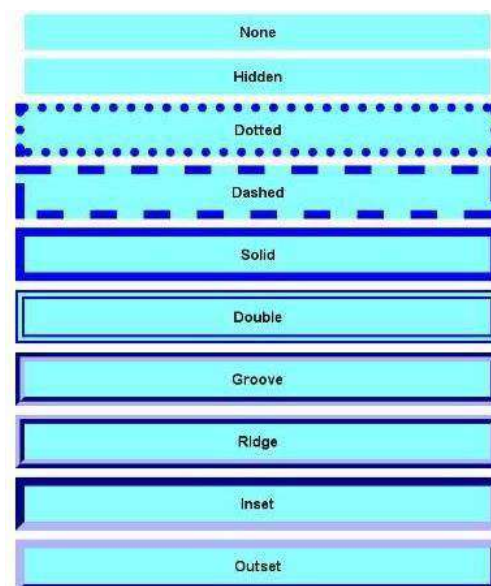
Y la palabra de forma original corresponde a los márgenes superior e inferior y la segunda a la izquierda y derecha de forma pareja:

```
margin: 10px 5px;
```

Fuente: <https://www.lawebera.es/xhtml-css/margen-relleno-bordes-de-capas-con-css-html.php#:~:text=En%20el%20código%20CSS%2C%20el,se%20le%20determina%20el%20margen.>

En cuanto al borde utilizamos el borde el cual el borde es el límite de la capa. A diferencia de los dos atributos anteriores, el borde tiene tres propiedades que puedes ser determinadas.

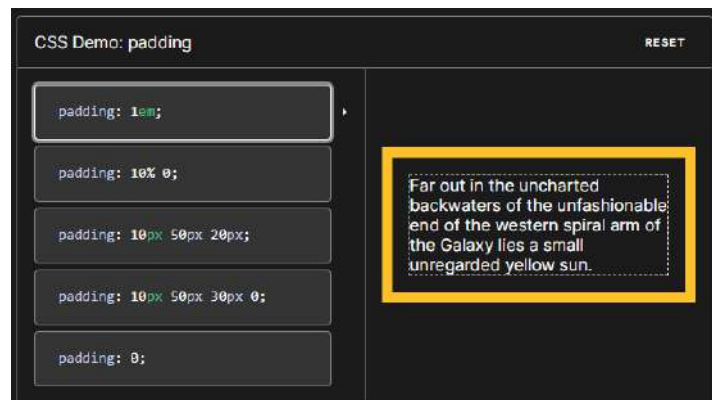
Ancho	border-width	thin, medium, thick (También puede emplearse una medida de longitud)
Color	border-color	Color o transparente
Estilo	border-style	none, hidden, dotted, dashed, solid, double, groove, ridge, inset y outset



Fuente: <https://www.lawebera.es/xhtml-css/margen-relleno-bordes-de-capas-con-css-html.php#:~:text=En%20el%20código%20CSS%2C%20el,se%20le%20determina%20el%20margen.>

4.31. Relleno y altura en CSS

La propiedad de relleno que utilizamos en CSS es el padding, es una propiedad o relleno que se crea alrededor del contenido de un elemento dentro de los bordes definidos y sirve para dar formato y diseñar una página web.



Fuente: <https://developer.mozilla.org/es/docs/Web/CSS/padding>

La propiedad que usamos para la anchura es la de width que es una unidad de medida en el cual nosotros le ponemos el valor y unidad que más nos convenga.

```
#lateral { width: 200px; }
```

Fuente: <https://uniwebdad.com/libros/css/capitulo-4/anchura-y-altura>

En cuanto la altura se denomina el height el cual también es una unidad de medida el cual nosotros le ponemos el valor y unidad deseada

```
#cabecera { height: 60px; }
```

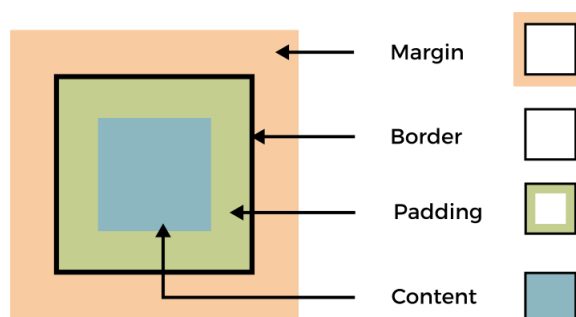
Fuente: <https://uniwebdad.com/libros/css/capitulo-4/anchura-y-altura>

4.32. Modelos de Caja en CSS

Todo en CSS tiene una caja alrededor, y comprender estas cajas es clave para poder crear diseños con CSS o para alinear elementos con otros elementos.

Lo más básico que debemos comprender y lo que se utiliza es:

- **El borde (border)**, en negro, es el límite que separa el interior del exterior del elemento.
- **El márgen (margin)**, en naranja, es la parte exterior del elemento, por fuera del borde.
- **El relleno (padding)**, en verde, es la parte interior del elemento, entre el contenido y el borde.
- **El contenido**, en azul, es la parte interior del elemento, excluyendo el relleno.



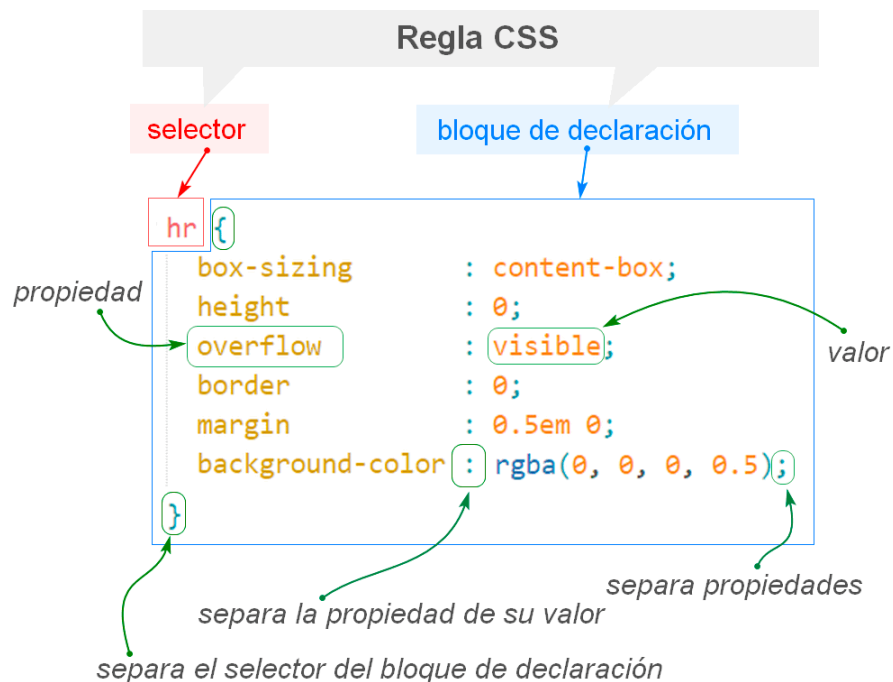
Fuente: <https://lenguajecss.com/css/modelo-de-cajas/que-es/>

4.33. Esquemas

El esquema de la regla en CSS se compone por:

- Una regla CSS se compone de un selector y de un bloque de declaración
- Cada declaración está compuesta por uno o más pares propiedad/valor

- El bloque de declaración empieza y acaba con los signos de llaves {...}
- Cada par propiedad/valor se separa por el signo punto y coma;
- Cada propiedad debe separarse de su valor por el signo dos puntos:



Fuente: <https://3con14.biz/Css/conceptos/5-esquema-de-regla-css.html>

4.34. Texto y Fuentes en CSS

Para declarar fuentes personalizadas se utiliza la regla `@font-face` de CSS, en la declaración de la fuente se utilizan las propiedades:

`font-family`: define el nombre de la fuente.

`src`: define la ruta del archivo y el formato de la fuente.

`font-style`: define el estilo de la fuente y texto (normal, italic, oblique).

`font-weight`: define el grosor del texto (normal, bold, bolder, lighter).

```
@font-face {
  font-family: 'Open Sans';
  src: url('fonts/opensans-light.woff2') format('woff2'),
       url('fonts/opensans-light.woff') format('woff');
  font-style: normal;
  font-weight: 300;
}
```

Fuente: <https://www.kodetop.com/incluir-fuentes-en-css-con-font-face/>

4.35. Iconos y Enlaces

De lo que son enlaces nosotros podemos personalizarlos de la misma forma que un texto normal, por medio de Font-size, text-align, etc.

Esto claro, teniendo el enlace que lo colocamos con un href.

```
<p><a href="#">Un enlace simple</a></p>

p {
  font-size: 2rem;
  text-align: center;
}
```

Fuente: https://developer.mozilla.org/es/docs/Learn/CSS/Styling_text/Styling_links

Un enlace simple

Fuente: https://developer.mozilla.org/es/docs/Learn/CSS/Styling_text/Styling_links

Dispones de varios navegadores,
como **Mozilla Firefox**, Google
Chrome y Microsoft Edge.

Fuente: https://developer.mozilla.org/es/docs/Learn/CSS/Styling_text/Styling_links

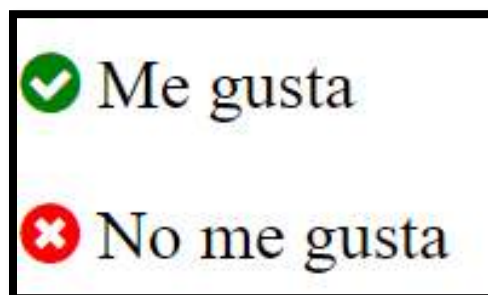
En cuanto a lo iconos , se sacan de internet , por medio de paginas las cuales esten dedicadas a ello, por lo cual vamos a necesitar su link para usarlo y ponerlo en nuestro archivo css, por ejemplo:

```
...
<link rel="stylesheet" href="css/font-awesome.css">
<link rel="stylesheet" href="css/estilo.css">
</head>
<body>
<p><i class="fa fa-check-circle verde"></i> Me gusta</p>
<p><i class="fa fa-times-circle rojo"></i> No me gusta</p>
</body>
```

En esta imagen podemos observar de donde sacamos el icono y que texto va a tener y como identificarlo

Fuente: <https://www.mclibre.org/consultar/htmlcss/css/css-fuentes-iconos.html>

Resultado:



Fuente: <https://www.mclibre.org/consultar/htmlcss/css/css-fuentes-iconos.html>

1.1 Creando entorno y diseños Web con XML

Por: Mario Ricardo Yoc Hernandez

1.1.1 Listas y Tablas

Las listas y tablas son dos tipos de estructuras comunes que se pueden crear con XML. A continuación, se presentan algunos ejemplos de cómo se pueden crear listas y tablas enXML:

Ejemplo de lista en XML:

```
<lista>
  <elemento>Elemento 1</elemento>
  <elemento>Elemento 2</elemento>
  <elemento>Elemento 3</elemento>
  <elemento>Elemento 4</elemento>
</lista>
```

Imagen: Código para usar una lista en XML

Fuente: <https://www.google.com/search?q=listas+y+tablas+en+xml>

En este ejemplo, se utiliza la etiqueta <lista> para crear la estructura general de la lista y la etiqueta <elemento> para cada uno de los elementos de la lista.

Ejemplo de tabla en XML:

```
<tabla>
  <fila>
    <columna>Encabezado 1</columna>
    <columna>Encabezado 2</columna>
    <columna>Encabezado 3</columna>
  </fila>
  <fila>
    <columna>Dato 1,1</columna>
    <columna>Dato 1,2</columna>
    <columna>Dato 1,3</columna>
  </fila>
  <fila>
    <columna>Dato 2,1</columna>
    <columna>Dato 2,2</columna>
    <columna>Dato 2,3</columna>
  </fila>
</tabla>
```

Imagen: Código para usar una tabla en XML

Fuente: <https://www.google.com/search?q=listas+y+tablas+en+xml>

En este ejemplo, se utiliza la etiqueta <tabla> para crear la estructura general de la tabla y la etiqueta <fila> para cada una de las filas de la tabla. Dentro de cada fila, se utilizan las etiquetas <columna> para cada uno de los datos de la tabla.

1.1.2 Pantalla CSS

La pantalla en CSS se refiere a la forma en que se visualiza una página web en diferentes dispositivos, como computadoras de escritorio, tabletas o teléfonos móviles. A continuación, se presentan algunos conceptos clave que se utilizan para definir la pantalla en CSS:

- **Viewport:** es el área visible de la pantalla donde se muestra el contenido de una página web. Se define con la etiqueta <meta> en el encabezado HTML y se utiliza para controlar la escala y el zoom de la página en diferentes dispositivos.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Imagen: Código viewport en XML

Fuente: <https://www.google.com/search?q=viewport+en+xml>

- **Media queries:** son reglas CSS que permiten adaptar la apariencia de una página web a diferentes tamaños de pantalla y dispositivos. Se definen utilizando la sintaxis @media y se pueden aplicar a diferentes características, como el ancho o la altura de la pantalla, la orientación del dispositivo, la densidad de píxeles, etc.

```
@media screen and (max-width: 600px) {
  /* estilos para pantallas con ancho máximo de 600px */
}
```

Imagen: Código @media en XML

Fuente: <https://www.google.com/search?q=media+en+xml>

1.1.3 Barras de Navegación

En XML, las barras de navegación se pueden crear utilizando elementos de marcado para definirla estructura y los enlaces de la barra. Por ejemplo, se puede utilizar el elemento <nav> para crear la barra de navegación y el elemento <a> para crear los enlaces.

```
<nav>
  <a href="#">Inicio</a>
  <a href="#">Acerca de</a>
  <a href="#">Servicios</a>
  <a href="#">Contacto</a>
</nav>
```

Imagen: Ejemplo de código XML para una barra de navegación

Fuente: <https://www.google.com/search?q=barra+de+navegacion+en+xml>

En este ejemplo, se define el elemento <nav> para crear la barra de navegación y se agregan cuatro elementos <a> para crear los enlaces de la barra. Cada enlace se define con el atributo href que especifica la URL a la que se debe dirigir el enlace cuando se hace clic en él.

1.1.4 Menu Desplegable

Para crear un menú desplegable en XML, se puede utilizar una combinación de elementos de marcado y estilos CSS. El menú desplegable se crea con la ayuda del elemento `<select>` y los elementos `<option>`. El elemento `<select>` define el menú desplegable y los elementos `<option>` representan las opciones en el menú.

```
<select>
  <option value="">Selecciona una opción</option>
  <option value="opcion1">Opción 1</option>
  <option value="opcion2">Opción 2</option>
  <option value="opcion3">Opción 3</option>
</select>
```

Imagen: Código XML para un menú desplegable simple

Fuente: <https://www.google.com/search?q=Menu+desplegable+en+xml>

En este ejemplo, el elemento `<select>` define el menú desplegable y los elementos `<option>` representan las opciones en el menú. El valor del atributo `value` se utiliza para identificar la opción seleccionada por el usuario.

1.1.5 Galeria de Imagenes

Para crear una galería de imágenes en XML, se puede utilizar una combinación de elementos de marcado y estilos CSS. Una forma común de crear una galería de imágenes es utilizando el elemento `` y los elementos `` para representar cada imagen en la galería.

```
<ul class="gallery">
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

Imagen: Código XML para una galería de imágenes simple Fuente:

<https://www.google.com/search?q=Galeria+imagenes+en+xml>

En este ejemplo, se utiliza el elemento `` para crear la galería y se agregan los elementos `` para cada imagen en la galería. Las imágenes se agregan como elementos `` con el atributo `src` que especifica la ruta de la imagen y el atributo `alt` que proporciona una descripción alternativa de la imagen para los usuarios que utilizan lectores de pantalla.

1.1.6 Formularios

Para crear formularios en XML, se utilizan los elementos de formulario estándar, como `<form>`, `<input>`, `<select>` y `<textarea>`, junto con sus atributos y valores para crear los campos de entrada y botones de envío.

```
<form action="procesar.php" method="post">
  <label for="nombre">Nombre:</label>
  <input type="text" name="nombre" id="nombre" required>

  <label for="email">Email:</label>
  <input type="email" name="email" id="email" required>

  <label for="mensaje">Mensaje:</label>
  <textarea name="mensaje" id="mensaje" required></textarea>

  <input type="submit" value="Enviar">
</form>
```

Imagen: Código XML para un formulario simple

Fuente: <https://www.google.com/search?q=Galeira+imagenes+en+xml>

En este ejemplo, se utiliza el elemento `<form>` para crear el formulario y se especifica el archivo de procesamiento en el atributo `action` y el método HTTP en el atributo `method`. Dentro del formulario, se agregan elementos de entrada como `<input>` y `<textarea>` para que el usuario pueda ingresar información. Cada campo de entrada tiene un atributo `name` para identificar el campo cuando se procesa el formulario y un atributo `id` para asociar la etiqueta `<label>` al campo de entrada.

Además, se utiliza el atributo `required` en cada campo de entrada para hacer que sea obligatorio que el usuario complete el campo antes de enviar el formulario. También se agrega un botón de envío con el tipo `submit` que enviará los datos del formulario al servidor.

1.1.7 Diseño de Sitio Web

El diseño de un sitio web en XML implica utilizar etiquetas y atributos específicos para estructurar y presentar contenido. XML en sí mismo no es un lenguaje de marcado visual, como HTML o CSS, sino que se utiliza para describir datos y su estructura. Sin embargo, se pueden aplicar hojas de estilo en cascada (CSS) para dar estilo a los elementos XML y crear un diseño visual atractivo.

En resumen, para diseñar un sitio web en XML, debes crear una estructura clara para tu contenido, aplicar hojas de estilo en cascada (CSS) para dar estilo a tus elementos XML y crear un diseño visual atractivo. Utiliza elementos de navegación y crea formularios para recopilar información. Recuerda que XML es un lenguaje de marcado de datos y no visual, por lo que necesitas aplicar CSS para crear un diseño visual atractivo.

TEMARIO CUARTO COMPUTACIÓN

Autoridades

Instituto Emiliani Somascos

Comunidad Somasca
Obras Somascas en Guatemala

Lic. Raúl Hernández Chacón
Director Técnico-Administrativo Instituto Emiliani Somascos

Lic. Henry Caal
Sub-director Instituto Emiliani Somascos

Lic. Juan Carlos Morales
Coordinador Académico

Prof. David Subuyuj
Coordinador Técnico

Armando Garcia
Coordinación de Pastoral

Bachillerato Industrial y Perito con especialidad en Computación

Silvio Ernesto Castro
Asesor de Práctica Supervisada

Christian Morales
Asesor de Práctica Supervisada

Promoción 2023

6to Computación

Sección "A"

- Edwin Eduardo Alvarez Arriaga
- Diego Angel de Jesús Alvizures Orantes
- David Nehemias Bautista Patzán
- Brayan Ismael Chóm Pacajoj
- Cristian Alexander Coromac López
- Willian Estuardo Cuevas Rivas
- Nohelia Guadalupe Culajay Oxcal
- Jenifer Celeste Cuyan Gomez
- Angel Daniel De León Alvarado
- Kevin Alexander De Leon Andrade
- Marta Alicia del Carmen Diaz Cifuentes
- Oscar Alexander Echeverría López
- Aroldo David Escobar López
- Thania Paola García Cos
- David Daniel Gerónimo Vaquias
- Patrick Leonardo Gómez Paz
- Joshua Alessandro González González
- Brandon David Grijalva Marroquín
- Braulio José Alexander Guarcax Lima
- Eddy Junior Guerrero Gomez
- Ricardo Alberto Hernández Bautista
- José Santiago Herrera Furlán
- Cristian Antonio Im Monterroso
- José Carlos Jordán Cardona
- Angel Enrique Juarez Asencio
- Gustavo Rafael Juárez Mendoza
- María Emilia Larios Contreras
- Glendy Sucely Lopez Cruz
- Didier Joshua Lopez Juarez
- Byron Aaron Mansilla Fuentes
- German Daniel Marquez Estrada
- Elder César Oswaldo Martínez Rustrian

Nombres

Sección "B"

- Adolfo Isaác Morales Delgado
- Erick Estuardo Morán Pech
- María José Noj Sor
- José Antonio Osoy Crespo
- Diego Manuel Fernando Pacheco Racanac
- Pavel Alexander Palma Carrillo
- Erwin Gabriel Patzán Roca
- Wilmer Estuardo Patzán Vail
- Leonel De Jesús Erasmo Pérez Lúcas
- Krystel Adriany Popa Patzán
- Rodrigo Fernando Ramírez Guzmán
- José Gabriel Ramos Pérez
- Fernando José Raxón Pirir
- Hilda Aileen Raymundo Diaz
- Gabriel Eduardo Rivas Morales
- Ariel Ernesto Rodríguez Morales
- Iker Rodolfo Rojas Reyes
- David José Gabriel Romero Alvarado
- Ronaldo Hezler Adriel Ruano Valle
- Gabriela Alexandra Santos Carrera
- Dilan Elizandro Sarceño Jacinto
- Dulce María Shuyá Baten
- Maryori Omeydi Sipac Sesam
- Edgar Laureano Soc Rac
- Juan Alejandro Tiney García
- Jeremy Jhosef Tobar Lopez
- Jonathán Javier Tubac Arreaza
- Diego Gabriel Vásquez Rosales
- Eduardo Steve Velásquez Gómez
- Ricardo Mauricio Ventura Martínez
- Axel Daniel Yoc Garcia
- Mario Ricardo Yoc Hernández

Centenario Somasco

en América

Cien años de la comunidad somasca en América, aquel lejano 5 de octubre de 1921 en el cual tuvimos el arribo en el puerto de la libertad, en el vecino país del Salvador, el que significa el primer apostolado somasco. Posteriormente en 1927 comenzaron las obras en Guatemala. Pensar que tenemos ya en este año 2022 más de 100 años de nosotros también ser partícipes de una obra que sigue hasta el día de hoy creciendo y expandiéndose, un mensaje de perseverancia que nos muestra que no todo llega rápidamente, pero que en algún momento todo da sus frutos.

Solamente Dios sabe qué es lo que hace y cómo lo hace, cuál es el rumbo de las cosas, de su corazón nacen las cosas que son buenas, la ayuda, la sabiduría, la inteligencia, solo vienen de él, sus planes para todos en los que en él confían, serán siempre de bendición. El es el que da las herramientas para que podamos salir adelante, siempre interfiere en todo lo que nos pasa en el día a día, en cada hora, en cada minuto, cada instante de nuestras vidas está planeado desde mucho antes de que nosotros conociéramos el primer instante de vida.

Eso significa la comunidad, una comunidad que significó una luz, una muestra de que el amor divino, que la mano de Dios aún está con nosotros y que seguirá estando, de la mano de todos aquellos colaboradores y de todos aquellos corazones que están conectados con la espiritualidad, que son verdaderos devotos a la obra verdaderamente cristiana, que mucha falta hace en estos tiempos tan difíciles.

Una comunidad que ha hecho muchas obras de bien para la sociedad, fundando instituciones educativas, seminarios, con el fin siempre de acercar más a las personas a Jesús, que es la verdadera visión y vocación que siempre se ha querido plasmar y transmitir hacia cada persona que está involucrada en esta obra masiva de devoción, caridad y trabajo. Lo que representó San Jerónimo a partir de su conversión, y que demostró toda su vida a partir de aquella experiencia propia que tuvo con la Virgen María. Su virtud es llevar el mensaje de salvación a partir de la ayuda hacia aquellos que más lo necesitan.

La pandemia que vivimos en estos últimos dos años, sin duda significó muchas cosas perdidas, familias que perdieron trabajo, que están pasando angustias económicas, incluso familias en dónde muchos miembros ya no están, suena como un mensaje pesimista, pero es la verdadera realidad en la cual nos encontramos viviendo, pero cuando hacemos algo, por ayudar a alguien más estamos contribuyendo al propósito de nuestras vidas.

Cuando empezamos a ver más allá de lo que nuestra perspectiva común ve, cuando dejamos de centrarnos en lo que pasamos, o a aún con problemas, luchas externas e internas, podemos ver aquellas realidades en las que nosotros, podemos intervenir, y quizá no cambiemos la realidad de la persona, hay situaciones realmente difíciles, pero, sí podemos dar una palabra incluso que venga del corazón, y dar ánimo nuevamente a una persona.

Porque la obra de caridad es algo más allá que ser llamado propiamente un santo, porque la caridad está en cada uno de aquellos corazones dispuestos a sacrificar algo, por servir, por ayudar a los demás. Porque la caridad requiere de intencionalidad también, el deseo y el sentido del deber, de dar sin esperar recibir nada a cambio, realmente eso es lo que completa una verdadera conversión, porque si recibimos la salvación, y la misericordia de Cristo, cómo no tener misericordia de aquellos que alguna vez fuimos nosotros.

El Centenario Somasco en América, significa más que solo 100 años, significan años de trabajo, de sacrificio, de cambios, de una transformación que incluso todos buscamos a diario. Quizá lo más significativo más allá de toda la historia que engloba estos cien años de la Comunidad Somasca en América, tenemos un proyecto de vida que busca reunir a todas aquellas personas marginadas, necesitadas, dirigirse a los niños, como el mismo Jesús lo haría en su tiempo. Resulta más que grata su presencia en América y propiamente en Guatemala, y deseamos que, así como tenemos un centenario se sigan cumpliendo más y más años de nuevos retos, pero sobre todo de practicar el amor al prójimo, el mensaje de salvación por medio de las obras.

Introducción

6to Computación

A lo largo de este trabajo como alumnos del 6to grado en bachiller industrial y perito con especialidad en computación, daremos a conocer cada uno de los temas vistos e impartidos por nuestros docentes en el ciclo escolar 2021 que corresponde al grado de Cuarto grado de nuestra especialidad ya antes mencionada.

Explicaremos y daremos a conocer en cada tema que en este documento se presenta, cada uno de los aspectos, características, aplicaciones y utilidades de dichos temas, tratando de dejar claro cada uno de ellos.

Abarcando de una manera clara y directa todo lo visto en ese ciclo escolar, principalmente orientado a la programación en C++, programación orientada a objetos en C#, conexiones a bases de datos, Visual Basic, entre muchos otros temas que serán de apoyo y guía para retroalimentar y fomentar en los estudiantes el correcto desarrollo de aplicaciones y proyectos que necesiten de estos tipos de software, lenguajes y conocimientos.

Justificación

El motivo por el cual todos los estudiantes realizamos la recopilación de toda la información que se presenta en este documento, es principalmente para que cada uno de nosotros tenga acceso a un medio de estudio y retroalimentación de todos los temas comprendidos, no solamente este año, sino también para un ámbito laboral y profesional que como futuros graduandos no estamos lejos a alcanzar. Cada uno podrá estudiar y repasar para mejorar y poner en práctica cada una de nuestras habilidades adquiridas a lo largo de todo este ciclo escolar. Ya que será información que no solo nos servirá este año, sino que, si nos esmeramos y ponemos empeño en cada una de las prácticas y competencias en cada tema, vamos a adquirir los conocimientos necesarios que nos servirán en un grado de educación superior universitaria y a lo largo de toda nuestra vida laboral.

Objetivos

6to Computación

En el siguiente documento tenemos una recopilación de todos los temas vistos en el primer año de nuestra carrera, todos estos temas son tanto prácticos como técnicos, este documento nos apoya a la retroalimentación sobre todo lo que hemos aprendido en el transcurso de los años.

El principal objetivo de este documento es poder estudiar todos los temas que contiene e introducimos nuevamente a los temas, y así no dejarlos a un lado ni olvidarlos, ya que tanto estos como los que vemos este último año de carrera son de suma importancia para nuestra realización como profesionales.

Nuestro temario está dividido en clases y capítulos, cada clase tiene cuatro capítulos representando los bimestres transcurridos. En cada uno de los capítulos se pueden identificar los temas principales que vimos todo ese año. Leer el temario es una buena práctica para repasar los conocimientos adquiridos con anterioridad.

Los objetivos específicos de nuestro temario son:

- Retroalimentarnos respecto a los temas pasados.
- Explicar de forma fácil de retener, clara y concisa cada tema tocado en este documento.
- Brindar ejercicios prácticos que ayudan a la retención de información.

El temario tiene como fin el estudio del mismo para los graduandos que cursen el presente año 2022.

Temas

Cuarto Grado Tecnología Vocacional I

Capítulo I

1. Programación básica en C++ (Consola).
 - 1.1 Entorno de trabajo
 - 1.2. Introducción a C++
 - 1.3. ¿Qué es C++?
 - 1.4. Instalar Editores
 - 1.5. Variables de entorno
 - 1.6. Estructura general de un programa
 - 1.7. Salida de datos en consola
 - 1.8. Compilador primario de C++
 - 1.9. Variables e Identificadores
 - 1.10. Tipos de datos
 - 1.11. Tamaños asociados
 - 1.12. Wchar_t
 - 1.13. La palabra reservada "void"
 - 1.14. La palabra "NULL"
 - 1.15. Operaciones aritméticas
 - 1.16. Operadores unitarios
 - 1.17. Operadores aritméticos binarios.
 - 1.18. Operador módulo
 - 1.19. Operadores incremento y decremento
 - 1.20. Estructuras de decisión
 - 1.21. Simple y Doble
 - 1.22. Múltiples o anidadas
 - 1.23. Swich Case
 - 1.24. Estructuras de repetición
 - 1.25. While
 - 1.26. Do While
 - 1.27. For
 - 1.28. Estructuras repetitivas anidadas
 - 1.29. Operadores lógicos
 - 1.30. Operador &&
 - 1.31. Operador | |
 - 1.32. Operador !
 - 1.33. Arreglos y Matrices
 - 1.34. Arreglos
 - 1.35. Unidimensionales y Bidimensionales
 - 1.36. Algoritmos de ordenamiento
 - 1.37. Métodos y Procedimientos
 - 1.38. Funciones
 - 1.39. Parámetros y Parámetros por valor

Capítulo II

0. Programación en C# (POO) Uso de controles de Conexión (ACCES): en Manejo de cadenas.

1. Introducción
2. Conceptos
3. Entorno de trabajo
4. Barra de herramientas
5. Nombramiento de cada herramienta.
6. Cuadro de texto
7. Etiquetas
8. Cuadros de imagen
9. Botones
10. Menús
11. Video
12. Sonido
13. Combo box
14. List box
15. Calendarios
16. Radio buttons
17. Check box
18. Timer
19. Panel
20. Form
21. Origen de las bases de datos
22. Creación de cadenas
23. Búsqueda
24. Concatenación
25. Búsquedas
26. Comparación
27. Generación de subcadenas
28. Implementación de entorno.
29. Creación de base de datos en Access
30. Relación de tablas
31. Implementación de llaves primarias
32. Implementación de llaves foráneas
33. Conexión a base de datos
34. Implementación de consultas
35. Formularios de registro

Capítulo III

0. Programación básica V.B (consola) Nivel 1

1. Definiciones
2. Sintaxis
3. Palabra reservada
4. Variable
5. Constante

0. Tipos de datos

1. Entero
2. Decimal
3. Carácter

0. Operadores aritméticos

1. Suma, resta, multiplicación, división y residuo

2. Operadores de igualdad y relaciones
 3. Igual, menor que, menor o igual que, mayor que, mayor o igual que, desigualdad
0. Estructuras de decisión
 1. If simple
 2. If...else
 3. Switch
0. Estructuras de repetición
 1. While
 2. Do...while
 3. For
 4. Operadores lógicos
 5. And, OR, Not
 6. Vectores y Matrices

Capítulo IV

0. Visual Studio .NET POO
 1. Windows Form App
 2. Framework.net
 3. Controles .Net
 4. Codificación Avanzada
 5. Formularios de .Net
 6. Medidas de seguridad
 7. ErrorProvider
 8. Anti-errores
 9. Try-catch

Análisis de Sistemas I

Capítulo I

1. Lógica en el entorno de programación y Deontología informática Seguridad industrial en informática.
 - 1.1. Lógica (PSEINT)
 - 1.1.1. Resolución de problemas lógicos
 - 1.1.2. Algoritmos
 - 1.1.3. Tipos de algoritmos
 - 1.1.4. Seudocódigo
 - 1.1.5. Diagramas de flujo
 - 1.2. Deontología Informática
 - 1.2.1. Concepto
 - 1.2.2. Medidas de contingencia y seguridad
 - 1.3. Seguridad Industrial en informática.
 - 1.3.1. Introducción a la seguridad industrial
 - 1.3.2. Señalizaciones
 - 1.3.3. Mapa de riesgos
 - 1.3.4. Seguridad informática.
 - 1.3.5. Métodos de seguridad informática.
 - 1.3.6. Procesos de seguridad industrial.

Capítulo II

- 2. Ciclo de vida del software y entorno a UML con su respectiva normalización.
 - 2.1. Ciclo de vida del Software
 - 2.1.1. Concepto
 - 2.1.2. Fases básicas
 - 2.1.3. Tipos de modelos
 - 2.2. UML
 - 2.2.1. Casos de uso
 - 2.2.2. Diagrama de proceso
 - 2.2.3. Diagrama de colaboración
 - 2.2.4. Otros tipos
 - 2.2.5. Diagrama de componentes
 - 2.2.6. Vistas
 - 2.2.7. Objetos
 - 2.2.8. Atributos
 - 2.2.9. Métodos
 - 2.3. Normalización
 - 2.3.1. Caso NF0
 - 2.3.2. Caso NF 1
 - 2.3.3. Caso NF 2
 - 2.3.4. Caso NF 3 Sección B

Capítulo III

- 3. Administrador de bases de datos
 - 3.1 Definiciones y conceptos de base de datos
 - 3.2 Arquitectura de bases de datos: relacionales y no relacionales
 - 3.3 Función del administrador de base de datos
 - 3.4 Tipos de datos estructurados
 - 3.5 Lenguajes estructurados de consultas
 - 3.6 Almacenamiento de datos
 - 3.7 Creaciones de tablas Data
 - 3.8 Warehouse
 - 3.9 introducción a DB2
 - 3.10 Qué es DB2
 - 3.11 Instalación del servidor de base de datos
 - 3.12 Configuraron DB2
 - 3.13 Base de datos de muestra Db2
 - 3.14 Creación de una base de datos de DB2
 - 3.15 Tipos de datos DB2
 - 3.16 Conexión a una base de datos DB2
 - 3.17 Manipulación de datos
 - 3.18 Restricción de DB2

Capítulo IV

- 4.0 Windows Server
 - 4.1 Conceptos
 - 4.2 Windows server 2012
 - 4.3 Diseño de la red
 - 4.4 Creación inicial de la red
 - 4.5 Instalación Ws2012
 - 4.6 Instalación DHCP
 - 4.7 Configuración DHCP
 - 4.8 Instalación DNS
 - 4.9 Configuración DNS
 - 4.10 Configuración DNS
 - 4.11 Inserción de clientes al dominio

- 4.12 Configuración clientes
- 4.13 Configuración de clientes
- 4.14 Instalación ADDS en subdominio
- 4.15 Configuración ADDS para subdominio Inserción clientes al subdominio

Teoría de la Información I

Capítulo I

1. Introducción a las habilidades Psicomotrices en el perfil de un informático y la implementación de sistemas operativos, sistemas de archivos, sistemas numéricos y unidades de medida
 1. Mecanet
 1. Concepto
 2. Instalador
 3. prácticas
 4. Sistemas Operativos
 5. Historia
 6. Versiones de sistemas operativos.
 7. Distribuciones
 8. Unidades de medida
 9. Concepto
 10. Múltiplos (bit, byte,
 11. Megabyte, Gigabyte, etc.)
 1. Conversiones
 12. Sistemas Numéricos
 13. Concepto
 14. Sistema binario
 15. Sistema octal
 16. Sistema hexadecimal
 17. Conversiones
 18. Sistemas de archivos
 19. Rutas y nombres de archivos (Windows, Linux)
 20. Clasificación: de disco, de red, de propósito especial
 21. FAT, NTFS, ext2, ext3, ext4.

Capítulo II

0. Teoría de base de datos, Estructuras de un sistema de base de datos y Seguridad de la información.
 1. Teoría de base de datos
 2. DBMS
 3. Tipos de DBMS
 4. Diferencia entre BD y BDMS
 5. DBMS Acces
 6. Modelo entidad relación
 7. Reglas
 8. Relaciones
 9. Llave primaria
 10. Llave foránea
 11. Llave candidata
 12. Estructura de datos
 13. DDL
 14. DML
 15. Seguridad informática
 16. Conceptos
 17. tipos de amenazas

18. Virus
19. Como proteger los equipos de los virus.
20. Medidas de seguridad
21. Acciones preventivas
22. Acciones detectivas
23. Acciones de restauración
- 24.

Capítulo III

0. Lenguajes de programación, Traducción y generación de código, Ciencias auxiliares de la informática y Licenciamiento de software
 1. Lenguaje de Programación
 2. Concepto
 3. Clasificación: bajo nivel, alto nivel.
 4. Traducción y generación de código.
 5. Compiladores
 6. Fases de un compilador
 7. Interpretes
 8. Fases de un intérprete
 9. Ciencias auxiliares de la informática.
 10. Inteligencia artificial
 11. Robótica
 12. Matemáticas
 13. Química
 14. Electrónica
 15. Mecatrónica
 16. Otras
 17. Licenciamiento de software
 18. Concepto
 19. Tipo de licencia: privada, copyleft, GPL, Free Ware, Shareware.
 20. Ejemplos
 21. Implementación.

Capítulo IV

0. Linux
 - 4.1.1. Historia
 - 4.1.2. Distribuciones
 - 4.1.3. Particionado el disco duro para instalación de Linux
 - 4.1.4. Aplicaciones de Linux: procesadores de texto, visores de imágenes, reproducción de audio y video.
 - 4.1.5. Instalación de software
 - 4.1.6. Bash (scripts básicos)
 - 4.1.7. Sistema de usuario y manejo de clave Archivos y permisos
 - 4.2. Tecnologías de la información
 - 4.2.1. Internet y sus servicios

- 4.2.2. Extranet
- 4.2.3. Intranet
- 4.2.4. Sistemas expertos

CUARTO GRADO

TECNOLOGÍA

BIMESTRE 1

Programacion basica en C++ (Consola)

1.1 Entorno de trabajo

Por: Edwin Eduardo Alvarez Arriaga

El entorno de trabajo que tenemos en c++ es un editor de texto y un compilador también estando lo que es el menú y barra de herramientas como el área de navegación (Proyecto, clases, depurar), área de trabajo (Edición de archivos seleccionados) y área de información (compilador, recursos, resultado de la compilación, depurar, ver resultados).

1.1.1 Editor de texto

Tiene la función de poder escribir nuestro programa con sus extenciones referentes, las fuentes de c++ mas comunes son; .cpp, .cp, o, c.

1.1.2 Compilador

El entorno de trabajo que tenemos en c++ es un editor de texto y un compilador también estando lo que es el menú y barra de herramientas como el área de navegación (Proyecto, clases, depurar), area de trabajo (Edicion de archivos seleccionados) y área de información (Compilador, recursos, resultado de la compilación, depurar, ver resultados).

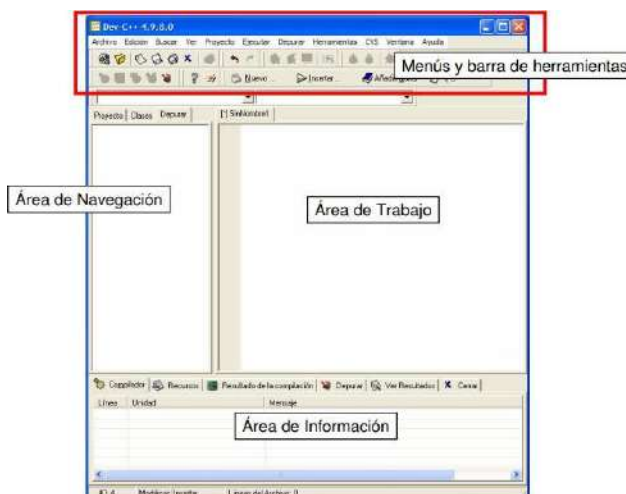


Imagen: Ejemplo del entorno de trabajo (Ejemplificación)

1.2 Introduccion a C++

El lenguaje c++ utiliza los mismos tipos de variables que el lenguaje C, es decir, void, char, int, loat y double, cabe recalcar que existen algunas modificaciones en la longitud de los tipos básicos siendo: short, signed, unsigned y long. La sintaxis para definir una variable es primero definir el tipo y después definir el nombre de la variable, el nombre de estas variables puede estar formado por letras (mayúsculas o minúsculas), números y caracteres especiales, pero esta nunca puede empezar con un numero.

```
1 [signed|unsigned] int posicion;
```

Imagen: Ejemplo de una variable (Ejemplificacion)

1.3 ¿Qué es C++?

Es un lenguaje de programación orientado a objetos que toma la base del lenguaje C, la intención de su creación fue orientado a la manipulación de objetos, siendo un lenguaje híbrido ya que tiene las funciones de: programación estructurada y la programación orientada a objetos.

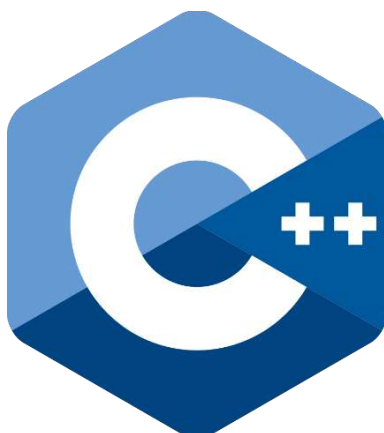


Imagen: Logo de c++ (decorativo)

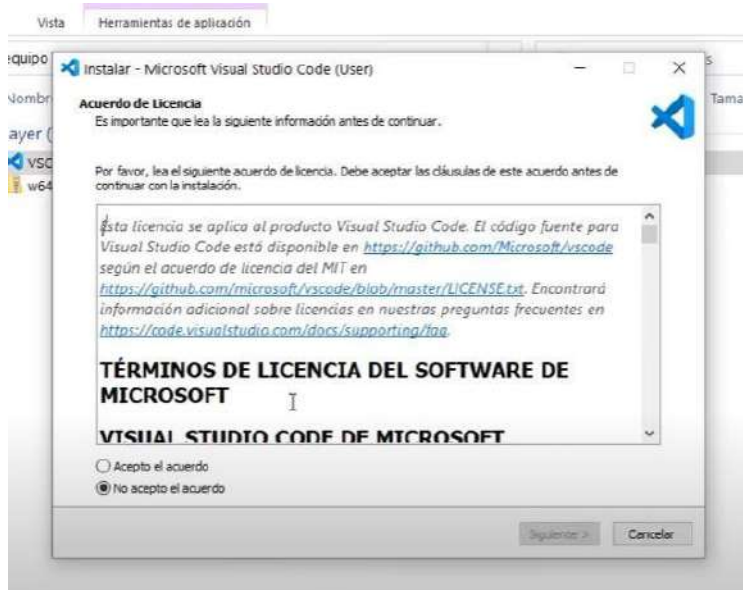
Fuente:

https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/ISO_C%2B%2B_Logo.svg/800px-ISO_C%2B%2B_Logo.svg.png

1.4 Instalar Editores



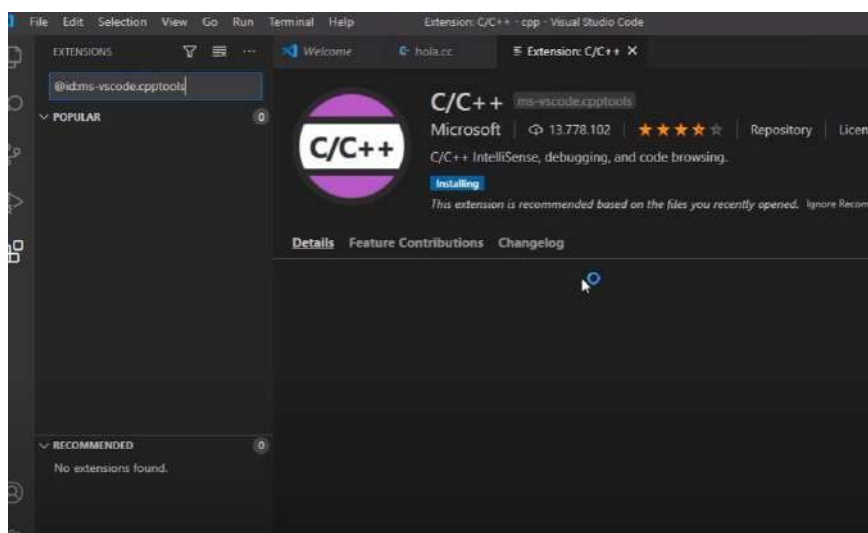
Ir a la pagina de visual code he intalarlo desde el botón download for windows



Iniciar la instalacion y aceptar el trato del usuario



Ejecutar al finalizar



Instalar la extencion de c/c++ , ahora podremos crear un archivo .C++

1.5 Variables de entorno

Los nombres de las variables de entorno tienen que iniciar en mayúsculas, los valores asignados a ellos no distinguen mayúsculas de minúsculas y pueden tener espacios en blanco iniciales y finales, se omiten las modificaciones en los valores después de iniciar el programa.

1.6 Estructura general de un programa

Los programas escritos en C++ tienen una estructura básica por: directivas de preprocesador, declaraciones globales, declaración de funciones.

1.6.1 Directivas de procesador

Hay que colocar al inicio del programa una directiva de preprocesamiento seguida de la cabecera de la biblioteca entre signos de menor que y mayor que, esto lo que hace es que indica el preproceso que debe de incluir la librería que elegimos a nuestro programa, todas estas directivas comienzan con un #.

```
#define identificador reemplazo
```

```
#ifndef TABLE_SIZE
#define TABLE_SIZE 100
#endif
int table[TABLE_SIZE];
```

```
#if TABLE_SIZE > 200
#undef TABLE_SIZE
#define TABLE_SIZE 200

#elif TABLE_SIZE < 50
#undef TABLE_SIZE
#define TABLE_SIZE 50

#else
#undef TABLE_SIZE
#define TABLE_SIZE 100
#endif

int table[TABLE_SIZE];
```

Imagen: Ejemplo de unas cuantas directivas)

1.6.2 Declaraciones globales

En esta sección se declaran todas las variables y cabeceras de funciones que serán vistas en el programas.

```
#include <iostream>
#define PI 3.1415

std::string autor: "Wikibooks";
int adicion(int, int);
```

1.6.3 Declaracion de funciones

Las funciones empiezan en main, siendo la principal, ya que esta es la primera que se manda a llamar al momento de ejecutar el programa, al no estar declarada aquí el programa hará como que nunca se hayan declarado.

```
#include <iostream>
#define PI 3.1415

std::string autor= "Wikibooks";
int adicion(int, int);

int main(int argc, char **argv)
{
    std::cout<<"El resultado de la suma de 1 y 2 es "<<adicion(1,2)<<std::endl;
    return 0;
}

int adicion(int a, int b)
{
    return a+b;
}
```

Imagen: Ejemplo de una declaración de funciones

1.7 Salida de datos en consola

Para poder imprimir datos en la pantalla c++ ofrece la función cout junto con el operador <<. La función cout permite imprimir cadenas de texto y valores almacenados en variables

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     string curso = "Informatica";
7     unsigned int version = 2;
8
9     cout << "Bienvenidos al curso de " << curso << " " << version;
10 }
11
12
```

1.8 Compilador Primario de C++:

Por: Diego Alvizures

El compilador primario es un programa que nos ayuda a traducir un lenguaje de programación (que es de alto nivel) a lenguaje máquina o lenguaje binario. En medio del traspaso del lenguaje de programación a lenguaje máquina existe el lenguaje ensamblador, que se encarga de realizar la traducción para que nuestro procesador pueda interpretar los procesos en código binario.



Imagen: Ejemplo de Lenguaje de alto nivel y bajo nivel (compilador)

Fuente: <https://edteam-media.s3.amazonaws.com/blogs/original/0cc7e2f7-1821-4603-88d6-7d7f271707a5.png>

1.8.1 MinGW

En el caso de C++ el compilador primario es MinGW, que aparte de ser el compilador primario de C++, también lo es de otros lenguajes como Java y Pascal. Este compilador deriva de una capa de UNIX minimalista que permite utilizar aplicaciones GNU desde Windows. En este caso este compilador es utilizado principalmente para con una biblioteca para ser utilizado en C++ es la cual es: GNU libstdc.

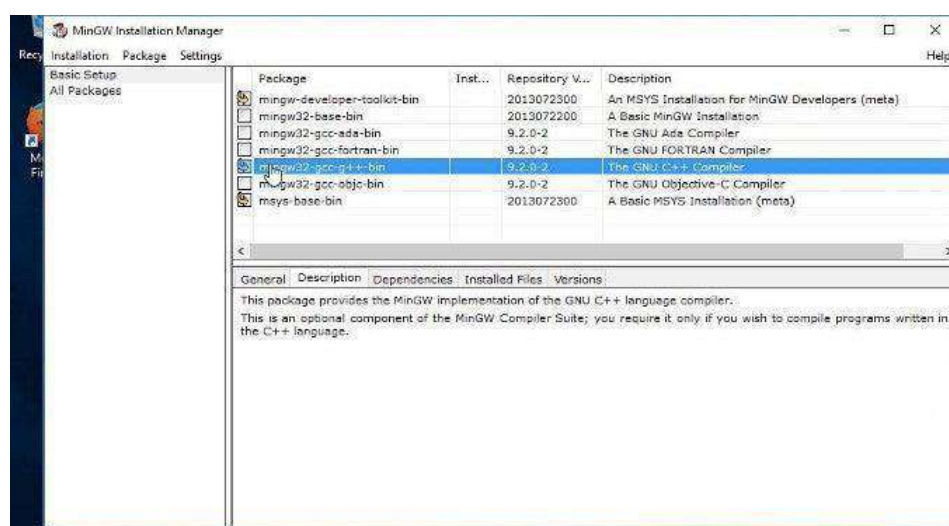
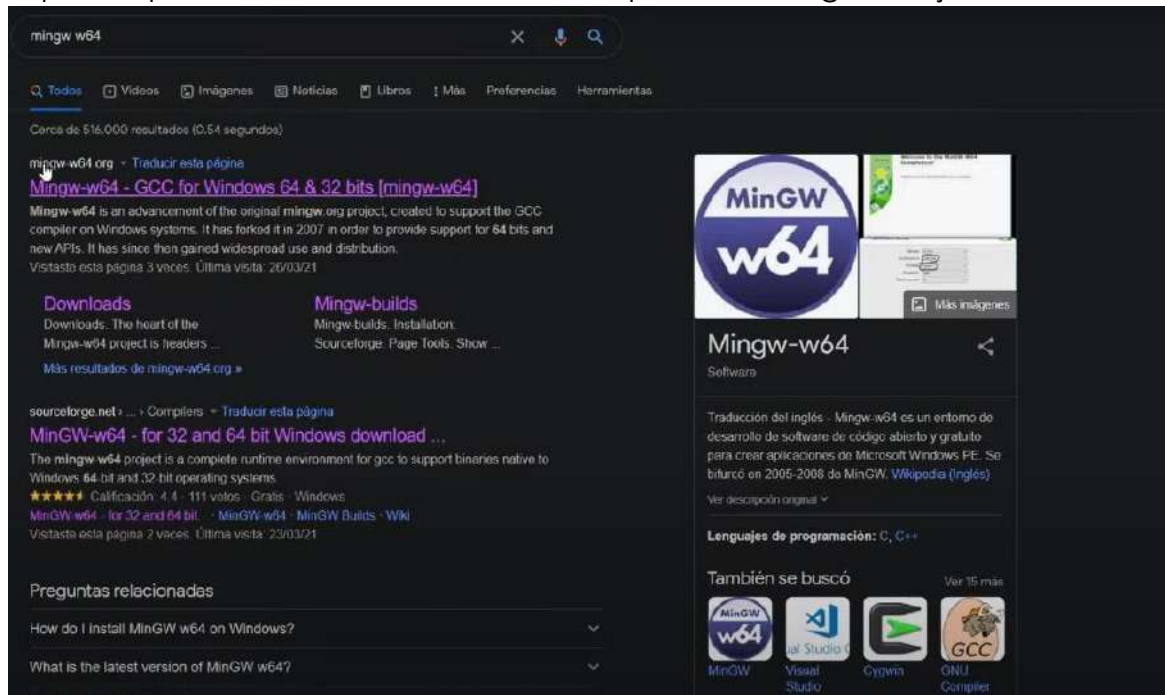


Imagen: Ejemplo de 1.8.1 de MinGW

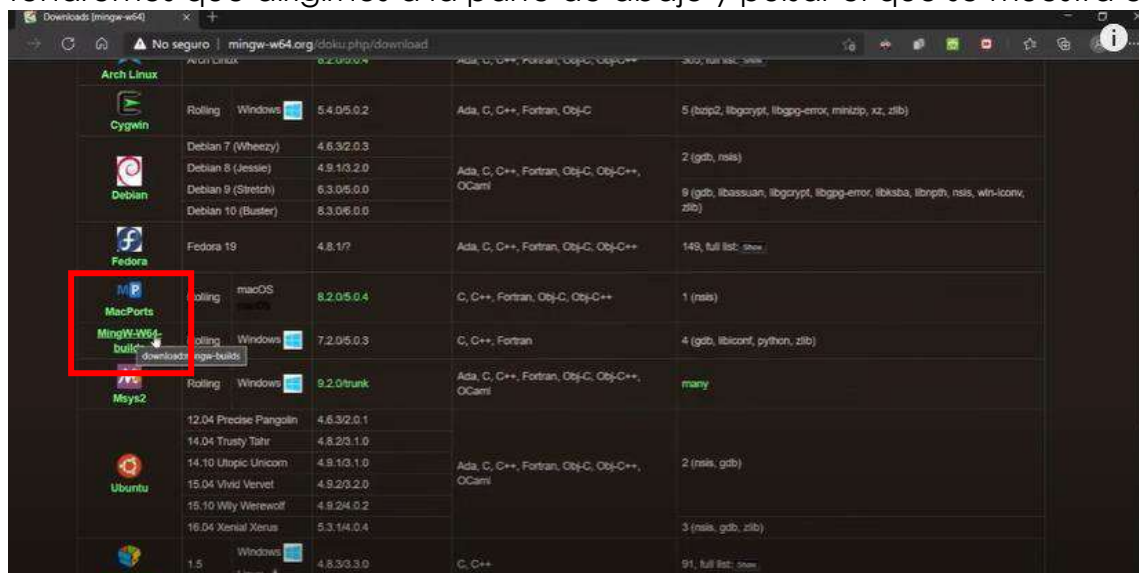
Fuente: <https://edteam-media.s3.amazonaws.com/blogs/original/0cc7e2f7-1821-4603-88d6-7d7f271707a5.png>

1.8.2 Instalación:

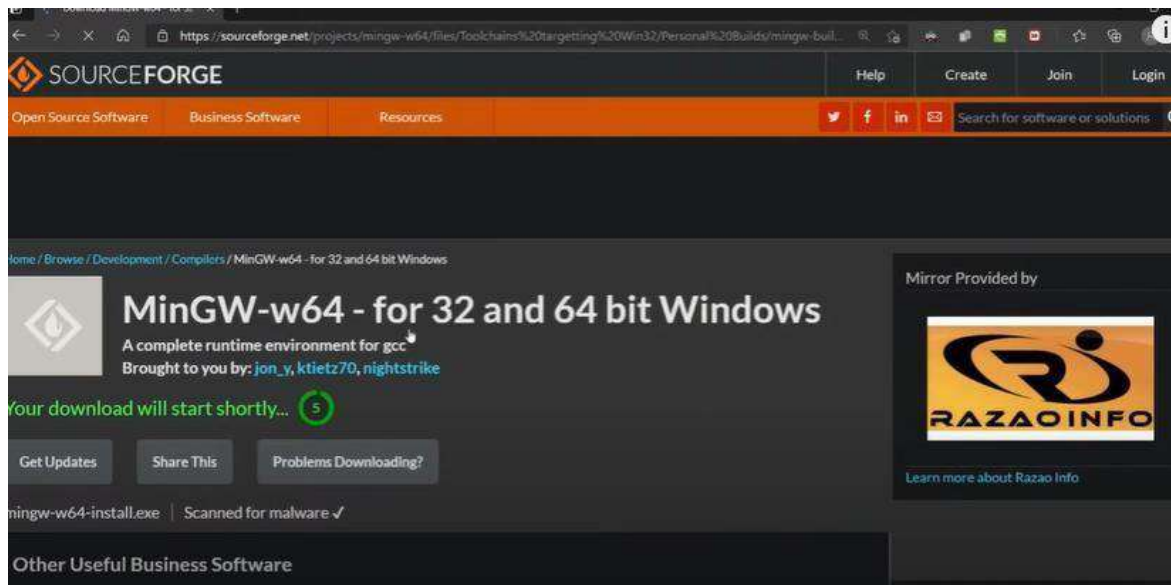
1. El primer paso es entrar a su web oficial para descargar el ejecutable.



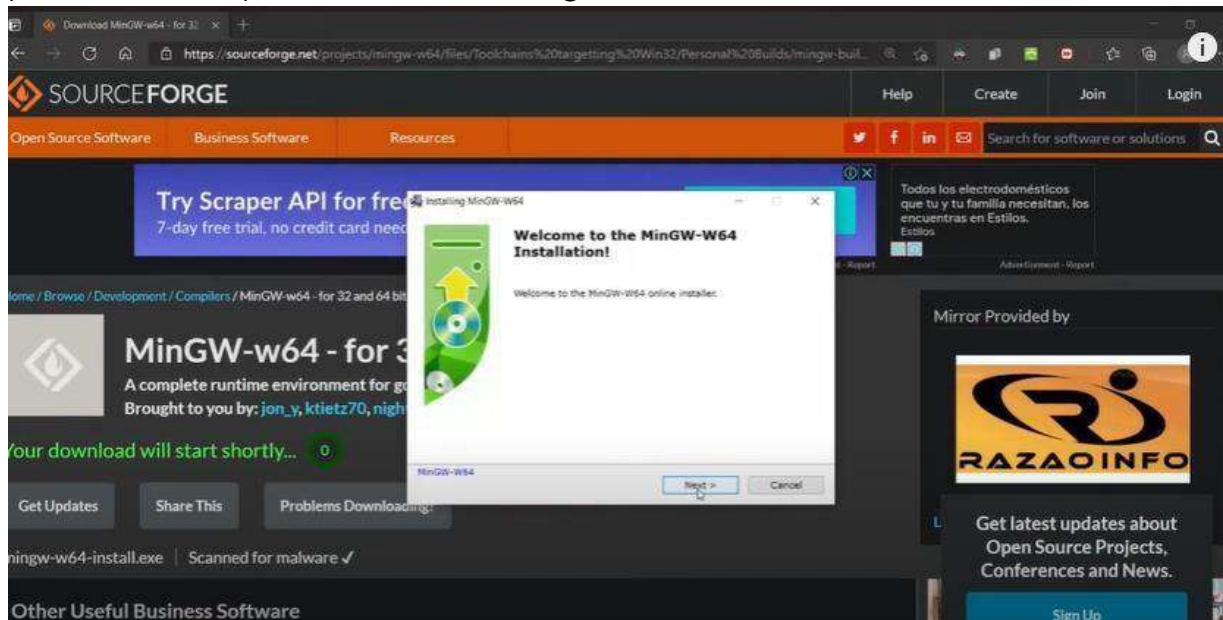
2. Una vez que entremos a la página nos aparecerá la lista de instaladores, tendremos que dirigirnos a la parte de abajo y pulsar el que se muestra en pantalla.



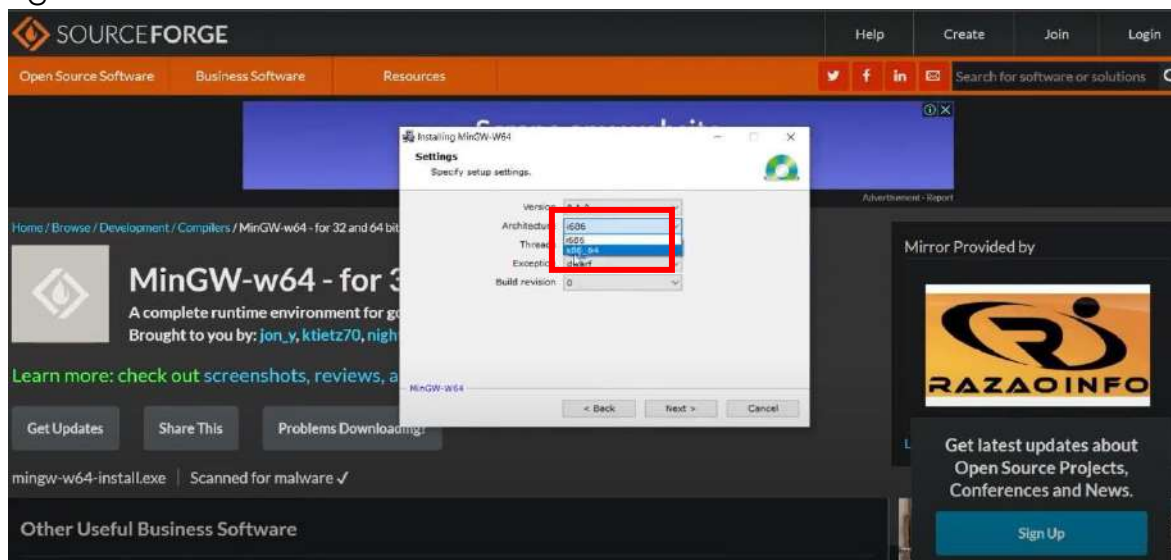
3. Una vez ya hemos pulsado ese icono, entonces nos llevará a un link de SorceForce donde empezará la descarga automáticamente



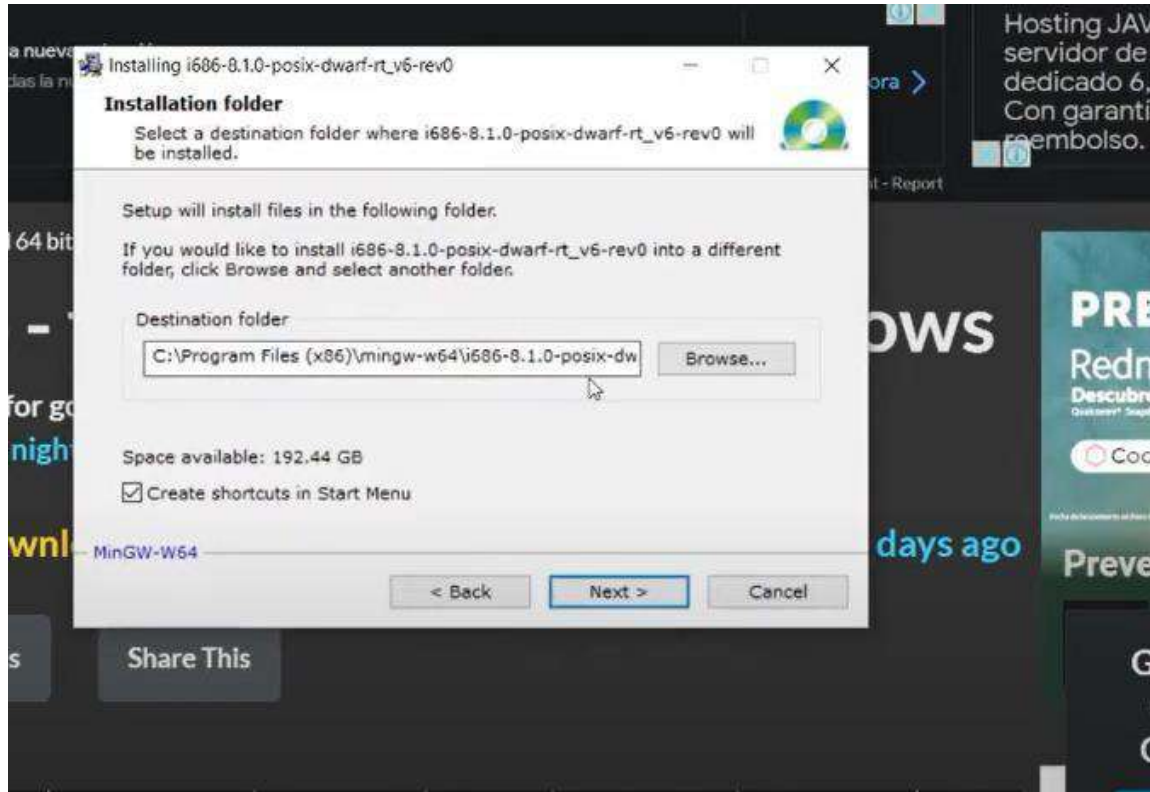
- Después de que haya concluido la descarga tendremos que ejecutar el instalador como administrador y posteriormente nos aparecerá esta ventana, en la cual pulsaremos simplemente el botón siguiente.



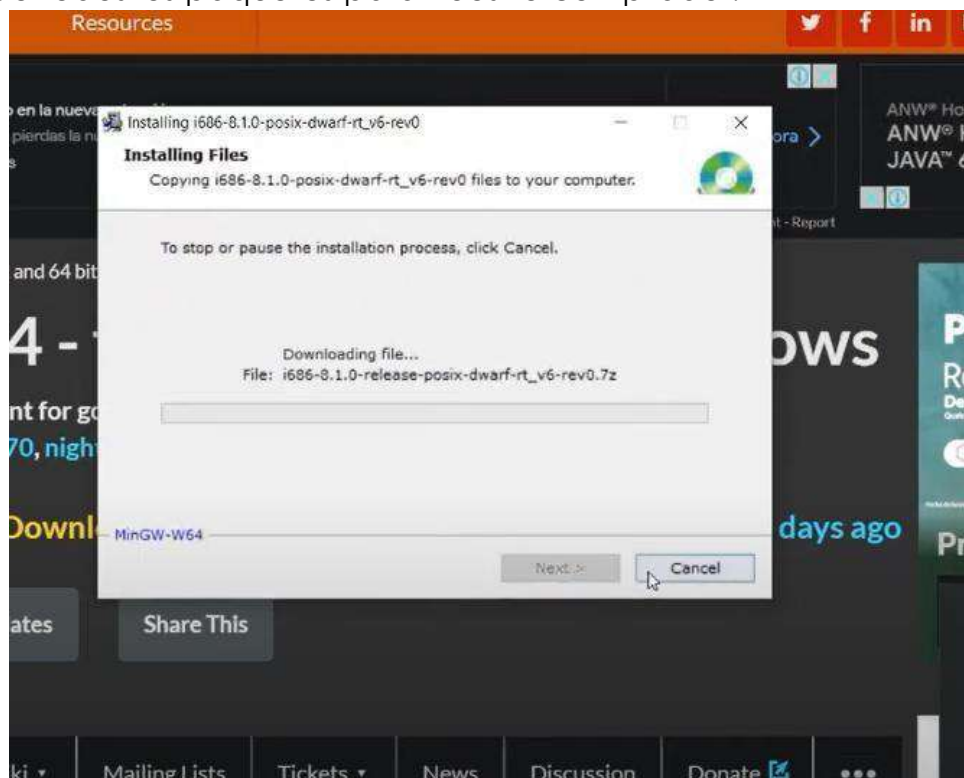
- Seguidamente de eso aparecerá una ventana en la cual tendremos que colocar la configuración. Es recomendable dejarla por defecto, solo tendremos que fijarnos en la arquitectura, ya que si nuestra computadora es de 32 bits tendremos que colocar la opción "i686" y si es de 64 bits "x86_64". Una vez seleccionado solo pulsaremos siguiente.



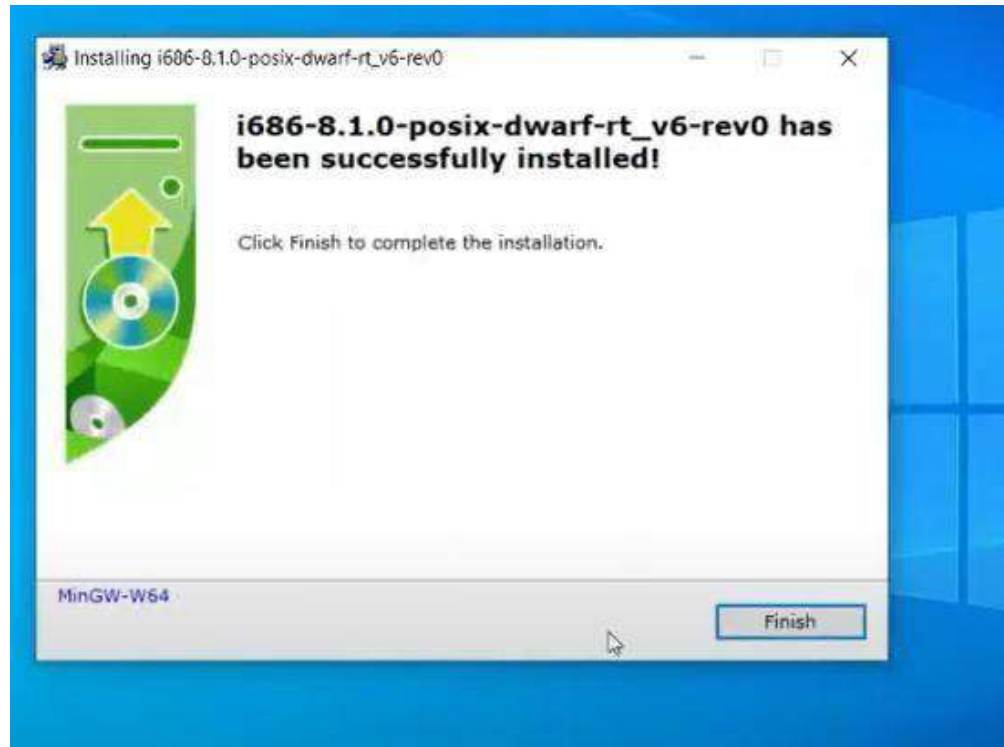
6. Después de presionar siguiente nos aparecerá la ubicación. Es recomendable no cambiar de ruta ya que así tendremos más fácilmente acceso a la ruta de nuestro compilador.



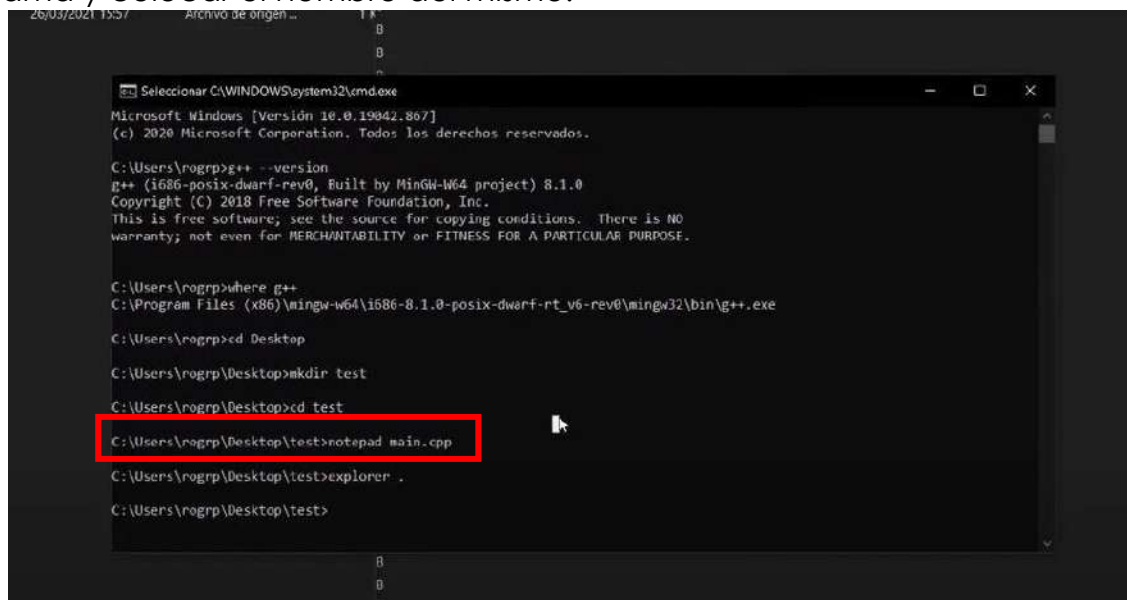
7. Una vez pulsado nos aparecerá una ventana en la cual aparecerá que se están descargando todos los paquetes para nuestro compilador.



8. Después de haber finalizado la descarga solo cerramos la ventana y ya tendremos listo nuestro compilador para poder ejecutar programas de C++ en nuestro equipo.



9. Para ejecutar un programa solo tendremos que dirigirnos a la dirección de nuestro programa y colocar el nombre del mismo.



1.9 Variables e Identificadores:

Los identificadores nos permiten usar fácilmente los datos que están almacenados en memoria, despreocupándonos de su posición en memoria, su dirección.

Con objeto de aportar legibilidad a los programas, debemos usar identificadores autoexplicativos, informando implícitamente del cometido de la variable en el programa.

```

double velocidad;    // PERFECTO
double vel;         // OK
double v;           // POCO EXPLICATIVO
double p;           // NADA EXPLICATIVO
    
```

Imagen: Ejemplo de variables

Fuente: https://www2.eii.uva.es/fund_inf/cpp/temas/2_tipos_variables/identificadores.html

1.9.1 Palabras reservadas

Tienen un significado especial para el compilador. Por ello, no pueden ser utilizadas como identificadores por el programador. Forman parte del vocabulario básico del lenguaje.

Estas son algunas de las más habituales:

```
asm dynamic_cast namespace reinterpret_cast try bool explicit new static_cast typeid catch false operator template
typename class friend private this using const_cast inline public throw virtual delete mutable protected true
wchar_t
```

1.9.2 Normas

Un identificador no puede tener cualquier nombre. Las normas son las siguientes: Estas son algunas de las más habituales:

1. Debe empezar por una letra o carácter de subrayado.

```
natural _var2 alumno_eii x1
```

A pesar de estar permitido, es recomendable no usar el subrayado al inicio, pues es utilizado por los desarrolladores de bibliotecas y compiladores y, eventualmente, podemos generar una colisión.

2. Las mayúsculas son caracteres distintos a las minúsculas

```
interes e Interes
```

Son variables distintas.

3. No pueden utilizarse las palabras reservadas del lenguaje, como `double`, `false` etc.
4. No deben llevar comas, puntos, acentos, espacios en blanco, la letra `ñ`, etc.

```
tamaño var 1 interés dia.24
```

no son válidos.

1.10 Tipos de Datos:

Los tipos de datos nos permiten que nuestras variables tengan un tipo de dato específico, el cual será muy importante a la hora de utilizar dicha variable.

Entre sus tipos se encuentran:

1.10.1 Entero (INT):

Un número entero simplemente es cualquier número (natural) sin punto por ejemplo 0 es de tipo entero, 846582 también lo es, pero 1.2 no porque tiene un punto y por eso no podríamos almacenar este valor en una variable de tipo entero, para declarar una variable de tipo entero utilizamos la palabra reservada `int`.

```
int casas = 5;
int objetos;
```

1.10.2 PUNTO FLOTANTE (FLOAT):

Este tipo de datos es la solución para números que tienen decimales como es el caso de 98.3 como vimos anteriormente no podemos incluirlo en variables de tipo entero, para declararlos se utiliza la palabra reservada float, ejemplos de uso son:

```
float metros=8.53;
float dólares;
```

1.10.3 TEXTO (CHAR Y STRING):

En este tipo de datos además de texto también pueden incluirse números, pero no pueden hacerse operaciones como suma, resta u otras para lo cual se necesitaría int o float porque el lenguaje los interpreta como texto, para declarar una variable con este tipo se emplea la palabra char que significa character y se traduce como carácter.

```
char carta[8] = "manzana" ;
char hoja[800] = "hola" ;
char ultimo[4000];
```

```
string libro = "Aquí un pequeño ejemplo";
string nombre;
```

1.10.4 BOOLEANO:

Una variable de tipo booleano solo acepta dos tipos de valores estos son true que significa verdadero o false que significa falso, se utiliza si se quiere guardar el estado de un resultado

```
bool reunir=false;
bool bicicleta;
```

Imágenes: Tipos de datos

Fuente: <https://www.codigazo.com/en-c/tipos-datos-variables-c-ejemplos>

1.11 Tamaños asociados:

1.11.1 char

Tamaño: 1byte

Rango: -127 a 127 o 0 a 255

```

#include<iostream.h>

main(){
    char nombre[50];
    int edad;
    cout<<"INGRESAR EDAD "<<endl;
    cin>>edad;
    cout<<"INGRESAR NOMBRE "<<endl;
    cin>>nombre;

    cout<<"SU NOMBRE Y EDAD SON : "<<endl;
    cout<<nombre<<" "<<edad<<endl;

    system("pause");
}
    
```

Output: INGRESAR EDAD 29
INGRESAR NOMBRE Carlos
SU NOMBRE Y EDAD SON : Carlos 29
Presione una tecla para continuar . . .

Imagenes: ejemplo char

Fuente: <https://www.codigazo.com/en-c/tipos-datos-variables-c-ejemplos>

1.11.2 unsigned char

Tam

año: 1byte

Rango: 0 a 255

1.11.3 signed char

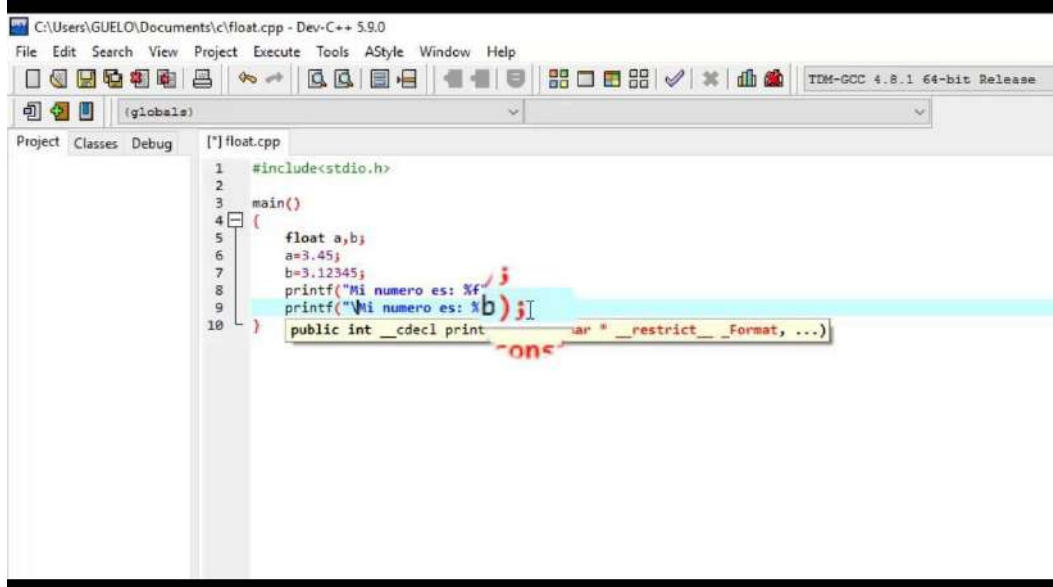
Tamaño

: 1byte Rango: -127 a 127

1.11.4 float

Tamaño: 4bytes

Rango: $\pm 3.4e \pm 38$ (~7 dígitos)



```
1 #include<stdio.h>
2
3 main()
4 {
5     float a,b;
6     a=3.45;
7     b=3.12345;
8     printf("Mi numero es: %f\n",a);
9     printf("Mi numero es: %b\n",a);
10 }
```

The screenshot shows a code editor window titled "float.cpp" with the following code:

```
1 #include<stdio.h>
2
3 main()
4 {
5     float a,b;
6     a=3.45;
7     b=3.12345;
8     printf("Mi numero es: %f\n",a);
9     printf("Mi numero es: %b\n",a);
10 }
```

The code is annotated with red handwritten notes: a semicolon ";" is written above line 8, and "ons" is written below line 9. A yellow tooltip is visible over the code on line 9, displaying "public int __cdecl printf(char * __restrict __format, ...)".

Imágenes: ejemplo float

Fuente: <https://www.youtube.com/watch?v=1jBwNmXeA18>

1.11.5 double

Tamaño: 8bytes**Rango: +/- 1.7e +/- 308 (~15 dígitos)**

```

3  #include<iostream> //biblioteca o libreria
4  //funcion principal
5
6
7  using namespace std; //determino mi forma de trabajo
8  int main(){
9
10 cout<<"hola mundo 2019" <<endl;
11 int numero = 15;
12 float flotante = 10.45;
13 double yoo= 16.5678;
14
15 cout<<numero;
16 cout<<flotante;
17 cout<<yoo;
18 cout<<flotante+numero;
19 return 0;
20 }

```

Imágenes: ejemplo double

Fuente: <https://welcometoprograming.jimdofree.com/lenguaje-c/variables-y-ejemplos-c/>

1.11.6 int

Tamaño: 4bytes**Rango: -2147483648 a 2147483647**

```

Sin_nombre1 contar en c.c. l1 a.cpp
5  int main()
6  {
7      float n;
8      int m;
9      cout<<"ingrese el numero 1: " <<endl;
10     cin>>n;
11     cout<<"digite el numero 2: " <<endl;
12     cin>>m;
13
14     cout<<"el producto es" <<n*m;
15     cout<<" ";
16     if (m<n){
17         cout<<("congrats");
18     }
19
20     return 0;
21 }
22

```

Imágenes: ejemplo int

Fuente: <https://welcometoprograming.jimdofree.com/lenguaje-c/variables-y-ejemplos-c/>

1.12 Wchar_t:

Es un tipo de carácter definido el cual nos ayuda a implementar un ancho definido por la implementación. En C++, los caracteres anchos son como el tipo de datos de caracteres, excepto por el hecho de que el tipo de datos char ocupa un espacio de un byte, mientras que el carácter ancho ocupa un espacio de dos bytes.

```
#include <iostream>
using namespace std;
int main()
{
//declare a wide character
wchar_t c = L'S' ;
//print the character value
cout << "The wide character value 'S' is: " << c << endl ;
//print the size of wide character
cout << "Wide character size is " << sizeof(c) ;
return 0;
}
```

Imagenes: ejemplo código wchar_t

Fuente: https://www.educba.com/c-plus-plus-wchar_t/

1.12.1 Tamaño:

En el compilador de Microsoft, representa un carácter ancho de 16 bits que se usa para almacenar Unicode codificado como UTF-16LE, que es el tipo de carácter nativo en los sistemas operativos Windows.

```
#include <iostream>
using namespace std;
int main()
{
//declare a wide character array string
wchar_t c[] = L"Hope never dies" ;
//print the character value
cout <<"The wide character length of Hope never dies " <<" is : "
return 0 ;
}
```

Imagenes: ejemplo código wchar_t

Fuente: https://www.educba.com/c-plus-plus-wchar_t/

1.12.2 Funciones de Caracteres Anchos:

Funcion: wcslean():

Función que ayuda a obtener la longitud de la cadena de caracteres anchos.

```
wcslen ( const wchar_t* str ) ;
```

Imágenes: ejemplo código wchar_t

Fuente: https://www.educba.com/c-plus-plus-wchar_t/

Función: wcsncpy():

Función que ayuda a copiar los caracteres sn del origen al destino. Si el extremo de origen es más pequeño que el tamaño sn, el destino tendrá algunos caracteres nulos.

```
wchar_t* wcsncpy( wchar_t* dst, const wchar_t* sr, size_t sn) ;
```

Imágenes: ejemplo código wchar_t

Fuente: https://www.educba.com/c-plus-plus-wchar_t/

Función: wcscat():

Función que ayuda a concatenar la cadena de origen con la cadena de destino.

```
wchar_t* wcscat ( wchar_t* dst, const wchar_t* sr) ;
```

Imágenes: ejemplo código wchar_t

Fuente: https://www.educba.com/c-plus-plus-wchar_t/

Función: wcsstr():

Función que ayuda a encontrar la primera aparición de la segunda cadena en la primera cadena. Se devolverá nulo si no está presente.

```
const wchar_t* wcsstr ( const wchar_t* str1, const wchar_t* str2) ;
```

Imágenes: ejemplo código wchar_t

Fuente: https://www.educba.com/c-plus-plus-wchar_t/

1.13 La palabra reservada void:

Es una función que se utiliza para poder colocar el código en este bloque y ya solo mandarlo a llamar en el "Main". La palabra "Void" especifica que no devuelve ningún valor. Cuando se usa una lista de parámetros, void especifica que la función no toma ningún parámetro.

```
#include <stdio.h>

void factorial ():

int main (){

    factorial ();

    system ("pause");
    return 0;
}

void factorial (){
    int x,i,aux;

    aux = 1;

    printf ("Introduce un numero entero: ");
    scanf ("%i",&x);

    for (i = 1; i <= x; i++){
        aux *= i;
    }

    printf ("El factorial del numero %i es %i.\n\n",x,aux);
}
```

Imagenes: ejemplo void

Fuente: <https://www.youtube.com/watch?v=96Lk9lw9y5w>

1.13.1 Utilización en C++:

Se usa principalmente para crear una función. Esta función se coloca en el apartado del Int Main, entonces cuando lee la función, ejecutará toda la parte del código de la función (void ()).

```
17 }
18 void calcular(int a, int b){
19     int s = suma(a,b);
20     int num;
21     cout<<"la suma es:"<<s<<endl;
22     cout<<"la multiplicacion se:"<<a*b<<endl;
23     cout<<"ingrese un numero:";
24     cin>>num;
25     if(num%2== 0){
26         cout<<"es par el numero"<<endl;
27     }else{
28         cout<<"es IMPAR el numero"<<endl;
29     }
30 }
```

The screenshot shows a C++ IDE with a function definition for 'calcular'. The function signature 'void calcular(int a, int b)' is highlighted in blue. The parameter 'a' is circled in red. The variable 's' is assigned the value of 'suma(a,b)', which is also circled in red. A red stick figure is drawn to the right of the code. The IDE interface includes a menu bar, toolbar, and a status bar at the bottom.

Imagenes: ejemplo void

Fuente: <https://www.youtube.com/watch?v=96Lk9lw9y5w>

1.14 La palabra NULL:

La palabra NULL sirve para especificar que un dato es nulo o no tiene contenido en él. Por lo tanto, si todos los punteros no utilizados se les asigna un valor nulo, evitando al mismo tiempo el uso de un puntero nulo, es posible prevenir el mal uso de un puntero no inicializado. En muchos casos, las variables no inicializadas hay algún valor de basura, hace que el programa difícil de depurar.

```

1  #include <iostream>
2  using namespace std;
3  int main () {
4  cout<<"Use of null functions";
5  cout<<"\n";
6
7  int *p_1 = NULL;
8  int *p_2 = NULL;
9  int *p_3 = NULL;
10
11 cout << "The calculated value of the first pointer: " << p_1 ;
12 cout<<"\n";
13 cout << "The calculated value of the second pointer: " << p_2 ;
14 cout<<"\n";
15 cout << "The calculated value of the third pointer: " << p_3 ;
16 return 0;
17 }
18

```

Imágenes: ejemplo suma

Fuente: <https://linuxhint.com/null-cpp/>

1.15 Operadores Aritméticos:

Los operadores aritméticos son aquellos que nos permiten operar las operaciones básicas. Se usan para calcular el valor de 2 o más números según el operador (signo) que se utilice. En programación las operaciones aritméticas son exactamente iguales, con las mismas reglas de jerarquía de signos.

1.15.1 Suma:

El operador lógico de suma sirve para calcular el valor en programación de las sumas implementadas tanto con variables como con dígitos numéricos.

```

in Nombre1
#include <iostream.h>
#include <stdlib.h>

main()
{
    int n1,n2, suma;

    cout<<"introduzca el primer numero"<<endl;
    cin>>n1;
    cout<<"introduzca el segundo numero"<<endl;
    cin>>n2;

    suma=n1+n2;
    cout<<"el resultado es "<<suma

```

Imágenes: ejemplo suma

Fuente: <https://www.youtube.com/watch?v=DoUdHy8uLSU>

1.15.2 Resta:

El operador lógico de resta sirve para calcular el valor en programación de las sustracciones implementadas tanto con variables como con dígitos numéricos.

```

1 #include <iostream>
2
3 int main()
4 {
5     system ("color 6f" );
6
7     int n1,n2,resta;
8
9     printf("Introduzca primer numero: ");
10    scanf("%d", &n1);
11
12    printf("Introduzca segundo numero: ");
13    scanf("%d", &n2);
14
15    resta=n1-n2;
16
17    printf("El resultado de la resta es: %d", resta);
18
19 }
    
```

Imágenes: ejemplo resta

Fuente: <https://www.youtube.com/watch?v=KZyrWXbJ9r4>

1.15.3 Multiplicación:

El operador lógico de multiplicación sirve para calcular el valor en programación de los productos implementados tanto con variables como con dígitos numéricos.

```

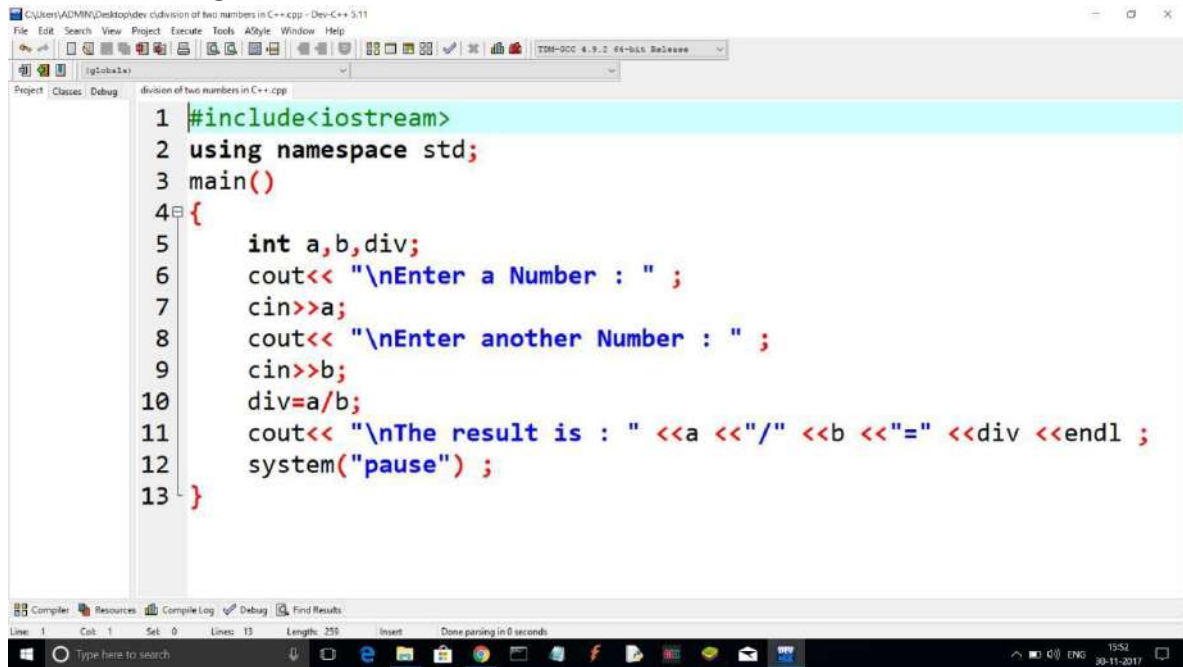
1 #include <iostream>
2 #include <conio.h>
3 using namespace std;
4
5 int main()
6 {
7     int multiplicacion, numero, a=1;
8     cout<<"Ingresa el numero de la tabla : ";
9     cin>>numero;
10
11     cout<<"La tabla del "<<numero<<" es :"<<endl;
12     do{
13
14         multiplicacion = numero * a;
15
16         cout<< numero <<" x "<< a <<" = "<< multiplicacion <<endl;
17         a++;
18     }
19     while(a<=10);
20
21 }
    
```

Imágenes: ejemplo multiplicacion

Fuente: <https://fernandoarciniega.com/crear-las-tablas-de-multiplicar-con-c-dev-5-11/>

1.15.4 División:

El operador lógico de división sirve para calcular el valor en programación de los cocientes implementados tanto con variables como con dígitos numéricos.



```

1 #include<iostream>
2 using namespace std;
3 main()
4 {
5     int a,b,div;
6     cout<< "\nEnter a Number : " ;
7     cin>>a;
8     cout<< "\nEnter another Number : " ;
9     cin>>b;
10    div=a/b;
11    cout<< "\nThe result is : " <<a <<"/" <<b <<"=" <<div <<endl ;
12    system("pause") ;
13 }

```

Imágenes: ejemplo division

Fuente: <https://www.youtube.com/watch?v=GbYUQpca7bY>

1.1 Programación básica en C++ Consola

Por: David Nehemías Bautista Patzán

1.16. Operadores unitarios

Los operadores unitarios son los operadores incrementales (++) y (-) que incrementan o disminuyen en una unidad el valor de la variable a la que afectan. Estos operadores pueden ir inmediatamente delante o detrás de la variable. Si preceden a la variable, ésta es incrementada antes de que el valor de dicha variable sea utilizado en la expresión en la que aparece. Si es la variable la que precede al operador, la variable es incrementada después de ser utilizada en la expresión. A continuación se presenta un ejemplo de estos operadores:

```
i = 2;
j = 2;
m = i++;      /* despues de ejecutarse esta sentencia m=2 e i=3 */
n = ++j;      /* despues de ejecutarse esta sentencia n=3 y j=3 */
```

Estos operadores se aplican a un solo operando y son los siguientes:

!, -, ~, ++ y --.

Operador	Operación
~ o compl	Complemento a 1 (cambiar ceros por unos y unos por ceros). El carácter ~ es el ASCII 126. El operando debe ser un tipo primitivo entero.
-	Cambia el signo al operando (esto es, se calcula el complemento a 2 que es el complemento a 1 más 1). El operador puede ser de un tipo primitivo o entero real.

```
#include <iostream>
using namespace std;

int main()
{
    int a = 2, b = 0, c = 0;

    c = -a;          // resultado c = -2
    cout << "c = -a --> " << -c << endl;

    c = compl b;    // resultado c = -1
    cout << "c = compl --> " << -c << endl;
}
```

Imagen: cómo utilizar los operadores unitarios:

Fuente: <https://baulderasec.wordpress.com/programando-2/cc/elementos-del-lenguaje-c/operadores/operador->

[condicional/#:~:text=Estos%20operadores%20se%20aplican%20a,%2C%20~%20%2C%20%2B%2B%20y%20%2D%2D%20.](https://baulderasec.wordpress.com/programando-2/cc/elementos-del-lenguaje-c/operadores/operador-)

1.17. Operadores aritméticos binarios.

Los operadores aritméticos son los más sencillos de entender y de utilizar. Todos ellos son operadores binarios. En C++ se utilizan los cinco operadores siguientes:

- Suma: +
- Resta: -
- Multiplicación: *
- División: /
- Resto: %

Todos estos operadores se pueden aplicar a constantes, variables y expresiones. El resultado es el que se obtiene de aplicar la operación correspondiente entre los dos operandos.

```
int main()
{
    int a = 10, b = 3, c;
    float x = 2.0F, y;

    y = x + a;
    c = a / b;
    c = a % b;
    y = a / b;
    c = x / y;
}
```

El resultado sería 12.0 de tipo float

El resultado es 3 de tipo int

El resultado es 1 de tipo int

El resultado es 3 de tipo int. Se convierte a float para asignarlo a y.

El resultado es 0.666667 de tipo float. Se convierte a int para asignarlo a c (c = 0).

El único operador que requiere una explicación adicional es el operador resto %. En realidad su nombre completo es resto de la división entera. Este operador se aplica solamente a constantes, variables o expresiones de tipo int. Aclarado esto, su significado es evidente: 23 % 4 es 3, puesto que el resto de dividir 23 por 4 es 3. Si a % b es cero, a es múltiplo de b.

Aclaremos la situación:

```
int main()
{
    int a = 10, b = 3, c;
    float x = 2.0F, y;

    y = x + a;
    c = a / b;
    c = a % b;
    y = a / b;
    c = x / y;
}
```

Para realizar la suma $x + a$ el valor del entero a se convierte a **float**, que es el tipo definido para x . No se modificará a , sino que su valor será convertido a **float** sólo para realizar la suma.


```

int main()
{
    int a = 10, b = 3, c;
    float x = 2.0F, y;

    y = x + a;
    c = a / b;
    c = a % b;
    y = a / b;
    c = x / y;
}

```

En la operación x / y sólo la parte entera es asignada a c , ya que c es de tipo `int`. Esto indica que los reales son convertidos a enteros, truncando la parte fraccionaria (*mantisa*).

Sin embargo observaremos que el compilador mostrará un aviso indicando que una conversión de `float` a `int` producirá una pérdida de precisión, por si se nos había pasado este detalle.

Para evitar que el compilador muestre este aviso tendríamos que especificar explícitamente que deseamos realizar esa conversión así:

```
c = static_cast<int>(x / y);
```

```

#include <iostream>
using namespace std;

/*
 * Area del triangulo
 */

int main()
{
    double base = 11.5, altura = 3.0, area = 0.0;

    area = base * altura / 2;

    cout << "Area = " << area << endl;
}

```

Imagen: código implementado los operadores aritméticos y binarios

Fuente: <https://baulderasec.wordpress.com/programando-2/cc/elementos-del-lenguaje-c/operadores/operadores-aritmeticos/>

1.18. Operador módulo

El operador módulo, % en lenguaje C++, es un operador binario y tiene sentido solo para números enteros. Se utiliza de la siguiente manera; sean A y B dos números enteros, si realizamos la operación A%B el resultado que obtenemos será el resto de la división entre A y B.

A primera vista parece que este operador no va a ser muy utilizado en programación, pero resuelve infinidad de problemas. Un ejemplo de un problema que resuelve este operador es, la devolución del cambio en una máquina expendedora de latas de refresco o de tabaco. Piensa en cómo podríamos calcular el cambio con este operador.

```
#include <iostream>
using namespace std;

float convertir(int gcent)
{
    float gfahr;
    gfahr = 9.0 / 5.0 * gcent + 32;
    return (gfahr);
}

int main()
{
    cout << " 20 °C = " << convertir(20) << " °F" << endl;
}
```

```
#include <iostream>
#include <conio.h>
#include <string.h>

using namespace std;
int main() {
    int numero;
    cout<<"Introduce un numero: ";
    cin>>numero;

    if(numero%2==0) // si el resto es cero el numero es par
        cout<<"PAR";
    else // En otro caso el número es impar
        cout<<"IMPAR";
    getch();
}
```

Imagen: código implementado operador modulo

Fuente: <https://primerospasosparaprogramar.wordpress.com/2010/11/22/el-operador-modulo-en-c/>

1.19. Operadores incremento y decremento

Los operadores de incremento y decremento de prefijo y postfijo afectan a sus operandos. La principal diferencia entre ellos es el orden en que se realiza el incremento o el decremento al evaluar una expresión. (Para más información, vea Operadores de incremento y decremento de postfijos.) En la forma de prefijo, el incremento o decremento tiene lugar antes de que el valor se use en la evaluación de la expresión, por lo que el valor de la expresión es diferente del valor del operando. En la forma de postfijo, el incremento o decremento tiene lugar después de que el valor se use en la evaluación de la expresión, por lo que el valor de la expresión es el mismo que el valor del operando. Por ejemplo, el siguiente programa imprime “++i =

```
// expre_Increment_and_Decrement_Operators.cpp
// compile with: /EHsc
#include <iostream>

using namespace std;

int main() {
    int i = 5;
    cout << "++i = " << ++i <<
}
```

Un operando de tipo entero o flotante aumenta o disminuye en el valor entero 1. El tipo del resultado es el mismo que el tipo del operando. Un operando de tipo de puntero aumenta o disminuye en el tamaño del objeto al que apunta. Un puntero incrementado apunta al siguiente objeto; un puntero disminuido apunta al objeto anterior.

```
//
expre_Increment_and_Decrement_Operators2.cpp
#define max(a,b) ((a)<(b))? (b):(a)

int main()
{
    int i = 0, j = 0, k;
    k = max( ++i, j
}
```

Imagen: código de implementación operadores incremento y decremento

Fuente: <https://learn.microsoft.com/es-es/cpp/cpp/prefix-increment-and-decrement-operators-increment-and-decrement?view=msvc-170>

1.20. Estructuras de decían

C++ es un lenguaje **compilado** que adopta los paradigmas imperativo y orientado a objetos.

Proviene de C, lo cual le brinda características de bajo nivel. Su nombre lleva el operador de incremento que se utiliza en C (++), indicando que es una “mejora” de C.

El punto de inicio de un programa en C++ es la función *main* (que retorna un entero y no recibe parámetros), por lo que esta función es imprescindible para que compile.

Detalles de sintaxis importantes: los bloques de código se definen entre **llaves** y las instrucciones finalizan con **punto y coma**. Los strings se indican con **doble comilla**.

Como detalle semántico, las variables deben declararse (indicando su tipo de datos) antes de utilizarse. También debe indicarse el tipo de valor de retorno de las funciones y el tipo de los parámetros que reciben (ya que C++ no es un lenguaje dinámico, como Python).

Para poder utilizar funcionalidad de entrada/salida de datos, se importa la biblioteca *iostream* colocando al principio `#include <iostream>`. Esto no es obligatorio, pero normalmente será necesario en la mayoría de los programas.

De la misma forma, para utilizar strings se recomienda importar la biblioteca correspondiente, mediante `#include <string>`

```
#include
<iostream> int
main()
{
    std::cout << "Hello,
world!\n"; return 0
}
```

En este ejemplo, la función *main* retorna el valor 0 (aunque esto no es

imprescindible para que el programa compile). Normalmente se utiliza el número 0 para indicar que no hubo error. Es decir, esta función retornará 0 si llega a ejecutarse hasta el final sin ningún error.

```
#include
<iostream> using
namespace std;

int main()
{
    cout << "Hello,
world!\n"; return 0;
}
```

La directiva “using” indica que se use ese espacio de nombres cada vez que sea necesario.

Un espacio de nombres es un ámbito dentro del cual los identificadores de variables, funciones, clases, etc. son únicos. Es decir, podrían existir dos

espacios de nombres diferentes, cada uno con una función llamada *miFuncion()* y con contenidos diferentes. Al no compartir el espacio de nombres, cada función puede llamarse por separado, anteponiendo el espacio de nombres:

```
namespace unEspacio
{
    void miFuncion()
    {
        // cuerpo de la función unEspacio::miFuncion()
    }
}

namespace otroEspacio
{
    void miFuncion()
    {
        // cuerpo de la función otroEspacio::miFuncion()
    }
}
```

Imagen: código de implementación Estructura de decían

Fuente: <http://patriciaemiguel.com/c++/2015/07/03/estructura-de-un-programa-cpp.html>

1.21. Simple y Doble

La estructura de control selectivo simple si evaluará una expresión lógica, y en caso de que sea verdadero se realizarán las instrucciones dentro de su bloque de instrucciones.

```
if (expresión Lógica) {
    //BLOQUE DE INSTRUCCIONES
}
```

La estructura de control selectivo doble if – else (si – sino) evaluará una expresión lógica y en caso de que sea verdadera se realicen las instrucciones dentro del primer bloque (bloque if), y en caso de que sea falsa se realicen las instrucciones dentro del segundo bloque (bloque else).

```
if (expresión lógica) {
    //BLOQUE DE INSTRUCCIONES 1
}
else {
    //BLOQUE DE INSTRUCCIONES 2
}
```

Veamos un ejemplo:

```
#include <iostream>

using namespace std;

int main() {

    int numero;
    cout<<"Ingrese un numero:";
    cin>>numero;
    if (numero%2==0) {
        cout<<"Es un numero par"<<endl;
    }
    else {
        cout<<"Es un numero impar"<<endl;
    }

    return 0;
}
```

```
Ingrese un numero:10
Es un numero par
```

Es posible anidar las estructuras simples y dobles, es decir, se puede poner un if junto con un else. Veamos un programa de ejemplo:

```
#include <iostream>
using namespace std;
int main(){
    int numero;
    cout<<"Ingrese un numero:";
    cin>>numero;
    if(numero<0){
        cout<<"El numero es negativo"<<endl;
    }
    else if(numero>0){
        cout<<"El numero es positivo"<<endl;
    }
    else{
        cout<<"El numero es 0"<<endl;
    }
    return 0;
}
```

```
Ingrese un numero:0
El numero es 0
Ingrese un numero:3
El numero es positivo
Ingrese un numero:-3
El numero es negativo
```

Imagen: código de implementación de simple y doble

Fuente: <https://fisiprogramacion.wordpress.com/2014/08/03/c-7-estructuras-de-control-selectivas-en-c/>

1.22. Múltiples o anidadas

Un único ciclo al interior de otro, funciona como un ciclo normal pero elevado al cuadrado, el ciclo externo comienza su ejecución con normalidad y luego va hasta el ciclo interno, y aquí es donde cambia todo; el ciclo externo no avanza hasta que el ciclo interno termine y una vez el ciclo externo avanza un paso vuelve a esperar al interno y así sucesivamente hasta que el externo termina, es por esto que es tan ineficiente el asunto.

Los ciclos anidados suelen usarse para llenar matrices un vector de varias dimensiones donde un ciclo recorre cada fila y otro cada columna o viceversa depende de lo que necesitemos. De este modo entonces el ciclo externo empieza por la columna cero y el interno llena cada casilla de la columna cero y luego de la uno y las dos y así.

Notemos que entonces el ciclo externo el de las columnas no avanza a la siguiente hasta que el ciclo interno no llene todas las casillas de esa columna (lo cual tiene sentido, pues debe llenarse en orden).

Sintaxis de ciclos anidados:

La sintaxis es sencilla, un ciclo con otro adentro, y lo que nos haga falta, pues podemos poner varias sentencias adicionales al interior de cualquiera de los dos ciclos.

```
for(int i = valor inicial; i < valor final; i++)
{
    /*
    .....
    .....
    Bloque de Instrucciones...
    .....
    .....
    */
    for(int j = valor inicial; j < valor final; j++)
    {
        /*
        .....
        .....
        .....
        .....
        .....
        */
    }
}
```



```
Bloque interno de Instrucciones....
```

```
....
```

```
....
```

```
* /
```

```
//Acá pueden haber más instrucciones
```

```
}
```

Como puedes ver entonces, es bastante simple, solo hay que notar algunas cosas interesantes: la primera y más importante es que la variable de control debe ser distinta en cada ciclo, fíjate que el ciclo externo usa la variable 'i' como variable de control mientras que el ciclo interno usa la 'j'. También debes notar que en el bloque interno de instrucciones podrías poner otro ciclo de cualquier tipo y al interior de este otro y así sucesivamente, cambiando el nombre de la variable de control (usar 'k' por ejemplo) para evitar mal funcionamiento. Finalmente debes saber que aunque son ciclos anidados no dejan de ser ciclos independientes en realidad, así que puedes poner las instrucciones que desees al interior de cualquier de ellos.

Nota: Cabe mencionar que no he hablado de tipos de ciclos, en realidad somos libres de anidar cualquier tipo de ciclo (for, while, do-while) y en cualquier cantidad, por ejemplo:

```
while(condición)
```

```
{
```

```
/*
```

```
Bloque de Instrucciones....
```

```
* /
```

```
for(int i = valor inicial; j < valor final; j++)
```

```
{
```

```
/*
```

```
Bloque interno de Instrucciones....
```

```
* /
```

```
while(condición2)
```

```
{
```

```

/*
Más instrucciones
*/
}
}
//Acá pueden haber más instrucciones
}

```

Como puedes notar, básicamente no tienes restricciones en ello, de hecho al interior de cada ciclo puedes poner condicionales y demás estructuras que requieras.

Ejemplo:

```

#include "iostream"
using namespace std;

int main()
{
    int matriz[10][10];

    for(int i = 0; i < 10; i++) //Ciclo externo
    {
        //Notemos que en ciclo interno no
        //usamos la variable 'i' sino
        //Si usamos i, el
        //comportamiento sería
        //inesperado for(int j =
        //0; j < 10; j++) //Ciclo
        //interno
        {
            //Llenamos la matriz con los números del 1 al 100
            matriz[i][j] = (i+1)*(j+1);
            // (i+1)*(j+1) va desde 1x1 hasta 10x10 (aunque en
            //desorden)
        }
    }

    system("pause");
    return 0;
}

```

Imagen: código de implementación de múltiples o anidadas

Fuente: <https://www.programarya.com/Cursos/C++/Ciclos/Ciclos-anidados>

1.23. Switch Case

La estructura condicional SWITCH – CASE se utiliza cuando queremos evitarnos las llamadas escaleras de decisiones. La estructura IF nos puede proporcionar, únicamente, dos resultados, uno para verdadero y otro para falso. Una estructura SWITCH – CASE, por su parte, nos permite elegir entre muchas opciones.

La estructura SWITCH – CASE nos permite evaluar una condición, la cual puede ser una entrada del usuario o una condición dada por un proceso del programa, y de acuerdo a esta evaluación, se ejecutarán un conjunto de instrucciones.

Uso típico del SWITCH – CASE

Normalmente el uso típico que se le suele dar a este tipo de condición es para la creación de menús, puesto que se introduce un número entero como opción y se ejecuta el código de la selección.

A continuación os muestro un diagrama donde se explica el flujo del SWITCH- CASE.

Sintaxis de SWITCH – CASE

La sintaxis de la estructura SWITCH – CASE es,

```
switch
(opción) case
opción :
    instrucciones-
opción break
```

- SWITCH, palabra reservada que indica el inicio de la estructura de control.
- opción, valor de tipo entero que determina la comparación que se realizará con los casos.
- CASE, palabra reservada, crea una etiqueta (la expresión constante) para una opción que se comparará con el valor opción que recibe SWITCH, es decir, evalúa un caso.
- :, los dos puntos son obligatorios, indica que se pasa el control a la sentencia si la evaluación del CASE es verdadera.
- BREAK, palabra reservada que permite salir del SWITCH una vez se cumpla uno de los casos, es por eso que en DEFAULT no es necesario ponerlo.
- DEFAULT, palabra reservada que se utiliza para los casos en que ninguna de las sentencias case sean verdaderas, es opcional y puede o no existir dentro del SWITCH.

Ejemplo:

```
# include
"iostream" using
namespace std;
int main()
{
    cin >> opcion;
    switch(opcion)
    {
        case 1: cout << "Usted ha seleccionado la opción 1";
        break;
        case 2: cout << "Usted ha seleccionado la opción 2";
        break;
        case 3: cout << "Usted ha seleccionado la opción 3";
        break;
        default: cout << "Usted ha ingresado una opción incorrecta";
    }
    // system("PAUSE"); //Solo ponla si no te da error
    return 0;
}
```

Imagen: código de implementación de Switch Case

Fuente: <https://www.programarya.com/Cursos/C++/Condicionales/Condicional-switch>

(Programación básica en C++ Consola)

1.24 Estructuras de repetición

Por: Cristian Coromac

Una **Estructura de Repetición en C++** le permite al programador especificar que se repita una acción, en tanto cierta condición se mantenga verdadera.

Las estructuras repetitivas nos servirán para repetir un bloque de instrucciones dada una condición. Existen 3 estructuras de control: while (mientras), do – while (hacer – mientras) y for(para).

Fuente:

<https://fisiprogramacion.wordpress.com/2014/08/04/c-8-estructuras-de-control-repetitivas-en-c/>

1.25 While

Por: Cristian Coromac

secuencialmente mientras la **condición** de entrada al bucle sea cierta.

La condición se evalúa al comienzo de la estructura. Esto supone que el bloque de instrucciones puede no ejecutarse ninguna vez si la condición es inicialmente falsa.

Un bucle while tiene asociado un bloque de **sentencias** que se ejecutarán

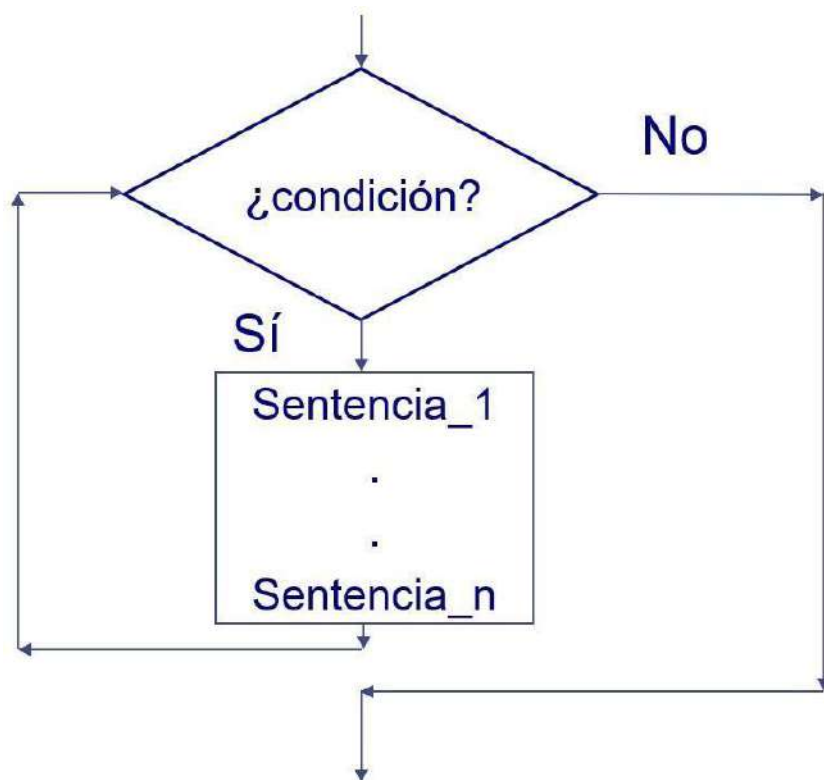


Imagen: Diagrama de flujo de la sentencia While

1.25.1 Sintaxis del Ciclo While en C++

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Dame el mes: (1, 2, ..., 12): ";
    int mes;
    cin >> mes;
    while (mes < 1 || mes > 12)
    {
        cout << "El valor introducido no es válido.\n";
        cout << "Dame el mes: (1, 2, ..., 12): ";
        cin >> mes;
    }
    if (mes == 1 || mes == 3 || mes == 5 || mes == 7 ||
        mes == 8 || mes == 10 || mes == 12)
        cout << "El mes tiene 31 días.\n";
    else if (mes == 2)
        cout << "El mes tiene 28 o 29 días.\n";
    else if (mes == 4 || mes == 6 || mes == 9 || mes == 11)
        cout << "El mes tiene 30 días.\n";
    else
        cout << ";Imposible!\n";
}
```

Imagen: Código implementado de C++

Fuente:

https://www2.eii.uva.es/fund_inf/cpp/temas/6_control_flujo_iterativo/while.html#:~:text=U n%20bucle%20while%20tiene%20asociado,la%20condici%C3%B3n%20es%20inicialmente%20falsa.

1.26 Do While

Por: Cristian Coromac

En resumen, un ciclo do-while, es una estructura de control cíclica que permite ejecutar de manera repetitiva un bloque de instrucciones sin evaluar de forma inmediata una condición específica, sino evaluándola justo después de ejecutar por primera vez el bloque de instrucciones

1.26.1 Sintaxis del Ciclo Do-While en C++

Por: Cristian Coromac

La sintaxis de un ciclo do-while es un tanto más largo que la del ciclo while en C++, sin embargo, no se hace más complicado, de hecho, con tan solo tener bien clara una condición de finalización para el ciclo tendremos prácticamente todo terminado.

Como podrás comprobar, ambos códigos hacen lo mismo, haciendo uso del ciclo while o del ciclo do while, sin embargo, con el ciclo do-while, hemos obtenido un beneficio interesante, primero que todo hemos podido evitar tener que inicializar la variable numero, segundo también pudimos eliminar las líneas repetidas donde se pedía el numero por fuera del ciclo y luego al interior del ciclo, aquí lo hicimos solo al interior, pues la condición se evaluaba mucho más abajo.

```

#include "iostream"

using namespace std;
int main()
{
    int numero;
    do
    {
        cout << "Ingrese un numero ";
        cin >> numero;
    }
    while(numero <= 100);
    system("PAUSE");
    return 0;
}

```

Imagen: Código implementado de c++

Fuente:

<https://www.programarya.com/Cursos/C++/Ciclos/Ciclo-do-while>

1.27 For

Por: Cristian Coromac

En resumen, un ciclo for es una estructura de control iterativa, que nos permite ejecutar de manera repetitiva un bloque de instrucciones, conociendo previamente un valor de inicio, un tamaño de paso y un valor final para el ciclo.

1.27.1 Como funciona un Ciclo For

Por: Cristian Coromac

Para comprender mejor el funcionamiento del ciclo for, pongamos un ejemplo, supongamos que queremos mostrar los números pares entre el 50 y el 100, si imaginamos un poco como sería esto, podremos darnos cuenta que nuestro ciclo deberá mostrar una serie de números como la siguiente: 50 52 54 56 58 60 ... 96 98 100.

Vamos a ver ahora como es la sintaxis de un ciclo for en C++.

1.27.2 Sintaxis del Ciclo For en C++

Por: Cristian Coromac

La sintaxis de un ciclo for es simple en C++, en realidad en la mayoría de los lenguajes de alto nivel es incluso muy similar, de hecho, con tan solo tener bien claros los 3 componentes del ciclo for (inicio, final y tamaño de paso) tenemos prácticamente todo hecho.


```

#include "iostream"
#include "stdlib.h"

using namespace std;
int main()
{

    for(int i=50;i<=100;i+=2)
    { //Notemos que escribir i+=2 es similar a escribir i = i + 2
        cout << i << endl;
    }
    system("PAUSE");
    return 0;
}

```

Imagen: Código Implementado en C++

Fuente:

<https://www.programarya.com/Cursos/C++/Ciclos/Ciclo-for#:~:text=En%20resumen%2C%20un%20ciclo%20for,valor%20final%20para%20el%20ciclo.>

1.28 Estructuras repetitivas anidadas

Por: Cristian Coromac

Los ciclos anidados NO son en sí una estructura de control, son de hecho un conjunto de estructuras de control anidadas, con anidadas me refiero a que una está dentro de la otra.

Uno de los mayores problemas en cuanto a eficiencia que hay en la programación es tener ciclos anidados, son simplemente terribles para el rendimiento, sin embargo, hay ocasiones en las que son indispensables.

En conclusión, los ciclos anidados no son una estructura de control por sí mismos, son un indicativo de que quizá deberíamos plantear la solución a algún problema si nos vemos obligados a usar ciclos anidados y más aún si es más de uno, sin embargo, debemos saber que a veces son indispensables.

1.28.1 Como funciona un Ciclo anidado

Por: Cristian Coromac

Un único ciclo al interior de otro, funciona como un ciclo normal pero elevado al cuadrado, el ciclo externo comienza su ejecución con normalidad y luego va hasta el ciclo interno, y aquí es donde cambia todo; el ciclo externo no avanza hasta que el ciclo interno termine y una vez el ciclo externo avanza un paso vuelve a esperar al interno y así sucesivamente hasta que el externo termina, es por esto que es tan ineficiente el asunto.

Los ciclos anidados suelen usarse para llenar matrices (un vector de varias dimensiones) donde un ciclo recorre cada fila y otro cada columna o viceversa (depende de lo que necesitemos).

1.28.2 Sintaxis del Ciclo For en C++

Por: Cristian Coromac

La sintaxis es sencilla, un ciclo con otro adentro, y lo que nos haga falta, pues podemos poner varias sentencias adicionales al interior de cualquiera de los dos ciclos.

Cabe mencionar que no he hablado de tipos de ciclos, en realidad somos libres de anidar cualquier tipo de ciclo (for, while, do-while) y en cualquier cantidad, por ejemplo:

```
while(condición)
{
    /*
     Bloque de Instrucciones....
    */

    for(int i = valor inicial; j < valor final; j++)
    {
        /*
         Bloque interno de Instrucciones....
        */
        while(condición2)
        {
            /*
             Más instrucciones
            */
        }
    }
    //Acá pueden haber más instrucciones
}
```

Imagen: Código implementado en C++

Fuente: <https://www.programarya.com/Cursos/C++/Ciclos/Ciclos-anidados>

Como puedes notar, básicamente no tienes restricciones en ello, de hecho, al interior de cada ciclo puedes poner condicionales y demás estructuras que requieras.

1.29 Operadores Lógicos

Por: Cristian Coromac

Una operación lógica utiliza como operandos datos booleanos. La mayoría de ellas compara dos expresiones o datos booleanos y indica una posibilidad de lo que puede ocurrir cuando se juntan. El resultado es otra expresión booleana.

Fuente: http://cpp.aprende-web.net/basico/basico3/basico3_3.php

1.30 Operador &&

Por: Cristian Coromac

Esta operación compara dos datos booleanos, de manera que el resultado será "true" (verdadero) solamente si los dos operandos son "true" (verdadero); en los demás casos el resultado será "false" (falso).

El signo para realizar la operación es && (doble signo ampersand) pero también podemos poner la palabra and. Por ejemplo, declaramos las siguientes variables booleanas.

```
bool v1=true, v2=true, f1=false, f2=false, r;
```

Las variables que tenemos inicializadas las utilizaremos como operandos, mientras que la variable r, sin inicializar, será donde obtengamos el resultado, operamos así:

```
r = v1 && v2;
```

El resultado será true. La misma operación la podemos escribir así:

```
r = v1 and v2;
```

Aunque es preferible ponerlo como en el primer ejemplo.

Los posibles valores que puede tener el resultado son los siguientes:

```
r = v1 && v2; // r: true
r = v1 && f1; // r: false
r = f1 && v1; // r: false
r = f1 && f2; // r: false
```

Aquí el resultado solo es verdadero si los dos operandos son verdaderos.

Fuente: http://cpp.aprende-web.net/basico/basico3/basico3_3.php

1.30 Operador | |

Por: Cristian Coromac

Este operador compara dos datos booleanos, de manera que el resultado será "false" (falso) solamente si los dos operadores son falsos; en los demás casos el resultado es "true" (verdadero).

El signo para realizar la operación consiste en la doble barra vertical | | (en la mayoría de los teclados se escribe con las teclas "Alt Gr" + 1). También tenemos la opción de escribir la palabra or entre los operandos.

Los posibles valores que puede tener el resultado son los siguientes: (utilizaremos aquí las mismas variables que en el ejemplo con and)

```
r = v1 | | v2; // r: true
r = v1 | | f1; // r: true
r = f1 | | v1; // r: true
r = f1 | | f2; // r: false
```

Aquí el resultado solo es falso si los dos operandos son falsos.

Fuente: http://cpp.aprende-web.net/basico/basico3/basico3_3.php

1.32 Operador “!”

Por: Willian Estuardo Cuevas Rivas

El Operador “!” es una Negación Lógica que es denominado como un Operador “No Lógico” y que se representa en el texto escrito como NOT, este operador es convertido en tipo bool, por lo que nos puede mostrar valores tanto falsos como verdaderos y viceversa. Este operador puede realizar conversiones, como por ejemplo un operador convierte un operando 0 en 1, devolviendo así el operador de tipo “True”, pero si se ejecuta distinto a 0 este operador se devuelve a “False”.

```
Option Explicit

Sub Negacion()
Dim bValor As Boolean
'Valor original Verdadero

bValor = True
MsgBox bValor

'Le aplicamos el operador
bValor = Not bValor
MsgBox bValor

'Valor original Falso
bValor = False
MsgBox bValor

'Le aplicamos el operador
bValor = Not bValor
MsgBox bValor

End Sub
```

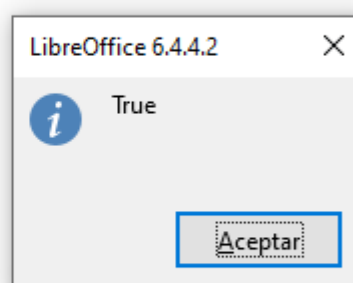


Imagen: Ejemplo del uso de Negaciones Lógicas.

Fuente: <http://profesores.fi-b.unam.mx/carlos/lcpi/p09/OPERADORES%20EN%20%20C++.pdf>

1.33 Arreglos Y Matrices

Por: Willian Estuardo Cuevas Rivas

Los Arreglos son estructuras de datos de un mismo tipo, por lo general los Arreglos tienen un tamaño que es la cantidad de objetos que pueden almacenar, por otro lado, los Arreglos son entidades estáticas debido a que se declaran en un cierto tamaño y se conservan a lo largo de la ejecución del programa que se está realizando. Los elementos de un Arreglo pueden ser inicializados por medio de una declaración con signo igual y una lista entre llaves de valores separados por comas, si en tal caso se declaran más inicializadores dentro del Arreglo que se está ejecutando dentro del programa, ocurrirá un error de sintaxis.

Los Arreglos también pueden formar parte de una función dentro de C++, donde estos almacenen datos y lean estos registros ingresados dentro del programa que se realice.

```

#include <iostream>
#include <stdlib.h>

using namespace std;

int main(int argc, char *argv[])
{
    int x=0,y=0;
    int m[33];
    do(
    if(x%3==0 && x!=100){
    m[y++] =x;
    }
    }while(x++<100);
    for(y=0;y<33;y++){
    cout << m[y] << endl;
    }
    system("PAUSE");
    return 0;
}

```

Imagen: Ejemplo del uso de Arreglos dentro de C++.

Fuente: <https://www.fing.edu.uy/tecnoinf/mvd/cursos/prinprog/material/teo/prinprog-teorico07.pdf>

Las Matrices son estructuras de datos similares a los Vectores ya que poseen la misma estructura, una secuencia de Vectores contenidos uno en el otro, en otras palabras, una Matriz es un Vector cuyas posiciones son de otros Vectores.

Una Matriz constituye de filas y columnas en donde se ingresan diferentes tipos de datos y/o valores idealmente estructurada por más de dos filas y columnas, la información que cada casilla puede almacenar puede ser simple o compleja. Para poder declarar una Matriz primero se necesita saber la cantidad de números o información que se tendrá que almacenar, por ejemplo, si son valores de tipo entero, decimales, cadenas de texto, entre otros, también necesitaremos el nombre de nuestra matriz y un tamaño máximo en las filas y columnas que vamos a crear dentro de nuestro programa, por último, vamos a declarar los valores dentro de nuestra Matriz utilizando corchetes "[]".

```

#include <stdio.h> /* for printf */
#include <algorithm> /* for std::swap() */

template<unsigned int R, unsigned int C, typename T>
struct AMatrix {
    T Data[R][C];
    operator T*(){
        return &Data[0][0];
    }
    T* operator[](unsigned int row){
        return Data[row];
    }
    T& operator()(unsigned int row, unsigned int col){
        return Data[row][col];
    }
};

```

Imagen: Ejemplo del uso de Matrices dentro de C++.

Fuente: <https://www.programarya.com/Cursos/C++/Estructuras-de-Datos/Matrices>

1.35 Unidimensionales y Bidimensionales

Por: Willian Estuardo Cuevas Rivas

Los Unidimensionales o mejor conocidos como Arrays, son variables estructuradas formadas a partir de un número de variables simples del mismo tipo, estos tipos de variables se le conocen como componentes o elementos del Array, dependiendo de la cantidad de elementos que se ingresen dentro de una variable será la dimensión del Array. Un Array en C/C++ puede tener más de dos dimensiones, dependiendo de lo que se le solicita en el programa estructurado.

Para declarar un Array Unidimensional solo basta con colocar el tipo de variable, acompañado de corchetes "[]", cerrado de punto y coma ";". Otra manera de declarar un Array es asignarle valores entre llaves e ir agregando cada uno de los elementos del Array.

```

4  #include <iostream>
5  #include <stdlib.h>
6  #include "conio.h"
7  using namespace std;
8
9  int main() {
10
11     int vec[] = {30, 50, 78, 95, 140};
12     bool band = false;
13     int valor, i = 0;
14
15     cout << "Favor ingresar un valor: ";
16     cin >> valor;
17
18     while( !band && i < 5 ){
19         if(vec[i] > valor){
20             cout << "Se debe insertar en la posicion " << (i+1);
21             band = true;
22         }
23         i++;
24     }
25
26     _getch();
27 }

```

Imagen: Ejemplo del uso de Arrays Unidimensionales dentro de C++.

Fuente: <https://www.uco.es/grupos/eatco/informatica/metodologia/cadenasyarrays.pdf>

Los Bidimensionales o mejor conocidos como Arrays Bidimensionales o dos dimensiones, son un tipo de dimensiones que podemos crear a base de tablas o matrices, este tipo de Array se pueden representar de una sola dimensión como por ejemplo insertando valores de tipo entero "1, 2, 3, 4..." o también representado por dos dimensiones "[4][8]", en este Array podemos ingresar cualquier tipo de números, pero hay que tener en cuenta que sus valores ya no serán una lista como en el Array Unidimensional ya que ahora pasaran a formar multiplicaciones por su mismo valor, por ejemplo si nuestro Array posee dos dimensiones "[4] y [8]" el total de valores que podemos ingresar serían 32, resultado de la multiplicación de las tablas y columnas.

Podemos declarar los Arrays Bidimensionales de manera como lo hacemos en el Array Unidimensional, colocando el tipo de valores seguido de las tablas y filas que ingresaremos dentro de nuestro programa.

```

5 | int main() {
6 |     int numeros[4]; //Declaramos nuestro vector que
7 |     int contador=0, sumatoria=0; //Declaramos varia
8 |
9 |     //Ciclo que solicitara introducir 5 numeros
10 | for(contador=0;contador<5;contador++){
11 |     printf("Introduce un numero: ");
12 |     scanf("%d", &numeros[contador]); //Guardamo
13 | }
14 | printf("\n");
15 |
16 | //Ciclo que hara la sumatoria de todos los nume
17 | for(contador=0;contador<5;contador++){
18 |     sumatoria = sumatoria + numeros[contador];
19 | }

```

Imagen: Ejemplo del uso de Arrays Bidimensionales dentro de C++.
 Fuente: <https://www.aprenderaprogramar.pro/2017/10/arrays-bidimensionales.html>

1.36 Algoritmos de Ordenamiento

Por: Willian Estuardo Cuevas Rivas

Los Algoritmos de Ordenamiento son operaciones que se basan en un método de orden secuencial de acuerdo a un criterio de ordenamiento, este se efectú con base al valor de algún campo que ingresemos dentro de nuestro registro. Esto nos ayuda a facilitar los valores de búsqueda en nuestros conjuntos de datos para que de tal forma queden secuenciados tanto numéricamente, alfabéticamente, alfanuméricamente, ascendente o descendente.

Existen diferentes tipos de ordenamiento, por ejemplo, entre estos están: El ordenamiento por inserción directa, Ordenamiento por selección, Por inserción binaria, Por el método de Shell, Ordenamiento Quick Sort y Ordenamiento Heap Sort.

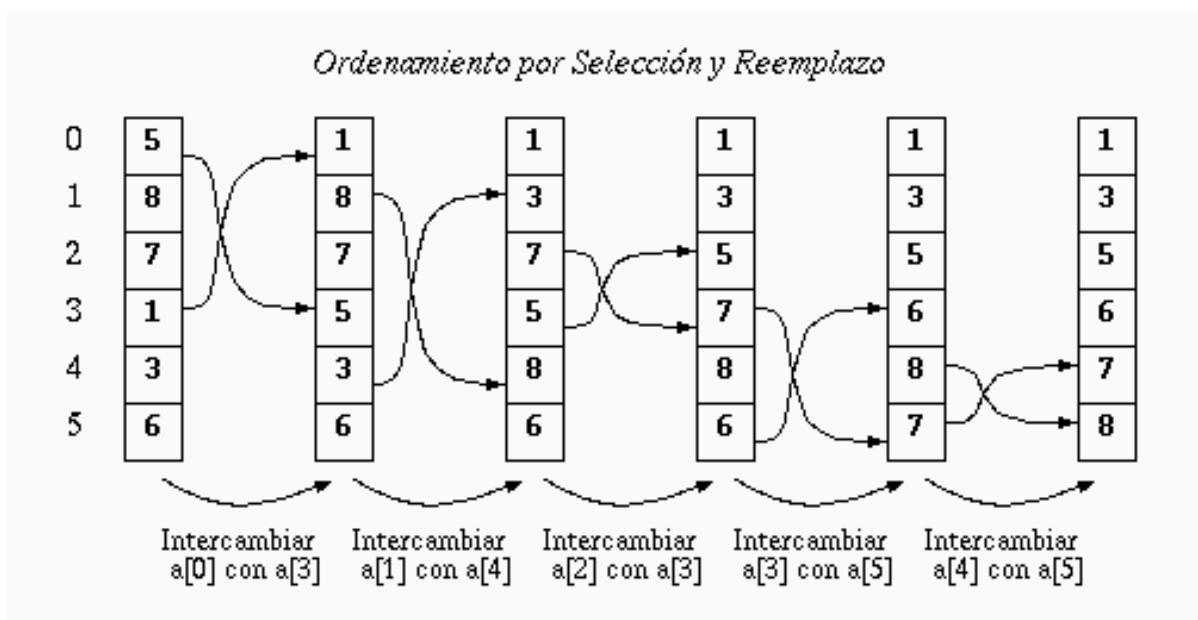


Imagen: Clasificación De los Algoritmos de Ordenamiento en C++.
 Fuente: <https://www.slideshare.net/geoXnet/ordenamiento-en-c>

1.37 Métodos y Procedimientos

Por: Willian Estuardo Cuevas Rivas

Los Métodos de una Clase en C++ son funciones que nos ayudan a manipular una gran cantidad de variables dentro de nuestra clase, esta encapsula todas las variables dentro de nuestro programa. Los Métodos se declaran y se definen de la misma manera que una función convencional, ya que estos explican el retorno, un nombre para el método y una lista de parámetros o datos de entrada, generalmente estos Métodos se hacen al interior de la clase mientras que la definición se hace por fuera de la clase, haciendo así que el método en cuestión sea definido por el mismo programa.

```
struct Point3s
{
    float x;
    float y;
    float z;
};

std::vector<Point3s> inputDataSet(count);
std::vector<Point3s> outputDataSet(count);

for (int i=0; i < count; ++i)
{
    Point3s inputElement = inputDataSet[i];
    Point3s result = // algoritmo independiente de la iteración
                   // que transforma inputElement
                   // utilizando objetos y métodos auxiliares.
    outputDataSet[i] = result;
}
```

Imagen: Ejemplo del Uso de Métodos en C++.

Fuente: <https://www.codingame.com/playgrounds/50557/clases-y-objetos-en-c-practica-1/miembros-de-clase-en-c-variables-y-metodos>

Los Procedimientos son un conjunto de instrucciones que se ejecutan sin tener algún retorno de ningún valor, existen situaciones en donde los Procedimientos no reciben valores o argumentos en absoluto, en otras palabras, un Procedimiento es mejor conocido como una función "Void" donde debemos utilizar la sentencia "Return" para poder ejecutar nuestro Procedimiento.

```
#include <stdio.h>
//declarar la función o el procedimiento
void imprimir();
void imprimir2(char mensaje[]);
int
//implementación del procedimiento o de la función
void imprimir(){
    printf("Hola Mundo\n");
}
void imprimir2(char mensaje[]){
    printf("%s\n",mensaje);
}
```

Imagen: Ejemplo sobre el uso de Procedimientos en C++.

Fuente: <https://www.programarya.com/Cursos/C++/Funciones#:~:text=Los%20procedimientos%20son%20b%20sicamente%20lo,nada%20que%20se%20lo%20impida.>

1.38 Funciones

Por: Willian Estuardo Cuevas Rivas

Una Función es un bloque de código que realiza alguna operación, una Función se puede definir como un parámetro de entrada que nos permite pasar argumentos a nuestra Función. Las funciones son útiles también para devolver valores como salida, para encapsular operaciones comunes en un solo programa, cada Función posee un nombre distinto para cada operación que se realiza en la ejecución, por ejemplo, funciones que aceptan dos enteros y devuelven la suma de "a" y "b" que son parámetros de tipo entero.

La Función puede ser llamada desde cualquier lugar del programa y de esta manera los tipos de parámetros devolverán la definición de la Función.

```
[*] new.cpp |
1 #include <iostream>
2 #include <cmath>
3 #include <cstdlib>
4
5 int main() {
6     using namespace std;
7     int edad;
8     cout << "Cual es tu edad? ";
9     cin >> edad;
10    imprimirEdad(edad);
11    system("pause");
12    return 0;
13 }
14 void imprimirEdad(int miEdad) {
15     using namespace std;
16     cout << "Tu edad es " << miEdad << endl;
17 }
```

Imagen: Ejemplo sobre el uso de Funciones Básicas en C++.

Fuente: <https://learn.microsoft.com/es-es/cpp/cpp/functions-cpp?view=msvc-170>

1.39 Parámetros y Parámetros por Valor

Por: Willian Estuardo Cuevas Rivas

Las Funciones al momento de ser llamadas, necesitan tener una cantidad de Parámetros que se necesitan para proporcionar los valores dentro de nuestro programa, de esta manera los Parámetros pasarán de la misma forma que fueron establecidos en un principio con la cantidad exacta que se le solicite al programa. Los Parámetros que se envían a una Función pueden ser viables, constantes o una expresión.

Los Parámetros por Valor o también conocidos como Paso por Copia, cuando se utiliza un Parámetro por Valor, se envían las copias introducidas de valores a nuestro mensaje de salida, esto significa que los cambios que le ocurran a estos valores serán afectados desde la Función,

Por ejemplo, queremos realizar una Suma con la Función if/else, pues primero declaramos todas las variables y al momento de realizar las salidas de texto, únicamente cambiamos las variables dentro de nuestra función, de esta manera solo afecta dentro de nuestro código y no dentro de nuestra ejecución del programa.

```
#include <iostream>
using namespace std;

//Prototipo de la función.
void MiFuncion(int n);

int main()
{
    int x = 7;
    cout<<"Variable antes de utilizar la funcion = "<<x<<endl;

    //Llamada de la función, como recibe un sólo parámetro, enviamos
    MiFuncion(x);

    cout<<"Variable despues de la llamada de la funcion = "<<x<<endl;
    return 0;
}

//Definición de la función.
void MiFuncion(int n)
{
    cout<<"Variable dentro de la funcion = "<<n<<endl;
    n = 100;
    cout<<"Variable luego de ser modificada en la funcion = "<<n<<endl;
}
```

Imagen: Ejemplo del uso de Parámetros con Funciones en C++.

Fuente: <https://www.aprenderaprogramar.pro/2017/08/parametros-de-funciones.html>

TECNOLOGÍA

Bimestre II

2. Programación en C#

2.1 Introducción

Por: Adolfo Morales

¿Qué es C#?

C# es un lenguaje de programación desarrollado por Microsoft, orientado a objetos, que ha sido diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. Se trata de un lenguaje simple, eficaz y con seguridad de tipos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C.

La sintaxis viene derivada de C y C++ y utiliza el modelo de objetos de la plataforma .NET, muy parecido al de Java, aunque incluye mejoras propias de otros lenguajes. Como curiosidad, el nombre de este lenguaje fue inspirado por la escala musical. En ella, la letra C equivale a la nota musical do y el símbolo # significa sostenido, lo que indica que es un semitono más alto. Así, C# sugiere que es superior a C y C++.

Variables en C#

Las variables son identificadores asociados a valores. Se declaran indicando el tipo de dato que almacenará y su identificador. Un identificador puede:

empezar por “_”.

contener caracteres Unicode en mayúsculas y minúsculas (sensible a mayúsculas y minúsculas).

Un identificador no puede:

-empezar por un número.

-empezar por un símbolo, ni, aunque sea una palabra clave.

-contener más de 511 caracteres.

2.2 Conceptos

Operadores en C#

Los operadores aritméticos funcionan igual que en C y C++. El resultado de los operadores relacionales y lógicos es un valor tipo booleano. Los operadores de cortocircuito evalúan el segundo operando solo cuando es necesario. Por su parte, los operadores a nivel de bits no se pueden aplicar a tipos bool, float, double o decimal.

Categoría	Operadores
Aritméticos	+ - * / %
Lógicos	! &&
A nivel de bits	& ^ ~
Concatenación	+
Incremento, decremento	++ --
Desplazamiento	<< >>
Relacional	== != < > <= >=
Asignación	= ^= <<= >>=
Acceso a miembro	.
Indexación	[]
Conversión	()
Condicional	? : ??
Creación de objeto	new
Información de tipo	as is sizeof typeof

Fuente: <https://openwebinars.net/blog/que-es-c-introduccion/>

Instrucciones de control en C#

Las instrucciones if-else, for, while, do-while, switch, return, break y continue son, básicamente, iguales que en C, C++ y Java. La instrucción foreach, al igual que en Java, realiza un ciclo a través de los elementos de una matriz o colección. En este ciclo se recorre la colección y la variable recibe un elemento de dicha colección en cada iteración. La instrucción goto sigue utilizando en C# a pesar de la polémica sobre su uso.

"Hola mundo" en C

Por último, pueden probar C# para recrear el famoso ejemplo de "Hola mundo" con el siguiente código:

```
using System;

public class Ejemplo
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Hola mundo");
    }
}
```

Fuente: <https://openwebinars.net/blog/que-es-c-introduccion/>

2.3 Entornos de trabajo

C# dónde se utiliza

C# es uno de los lenguajes más usados hoy en día por grandes y pequeñas compañías debido a su versatilidad y sencillez para el desarrollo de aplicaciones de cualquier índole que resultan eficientes y seguras.

Es importante decir que C# está diseñado para la plataforma .NET de Microsoft, pero puede permitir escribir programas para otras plataformas como Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

C# se puede utilizar para desarrollar todo tipo de aplicaciones: desde videojuegos hasta aplicaciones web como Stack Overflow. Algunas de los usos de C# más destacados son:

- Aplicaciones de escritorio, en consola o con interfaz gráfica usando -WinForms, WPF, UWP o la nueva .NET MAUI.
- Aplicaciones para dispositivos móviles con Xamarin.
- Servicios de API y páginas web con ASP.NET.
- Microservicios con ASP.NET Core.
- Software embebido con .NET Nanoframework

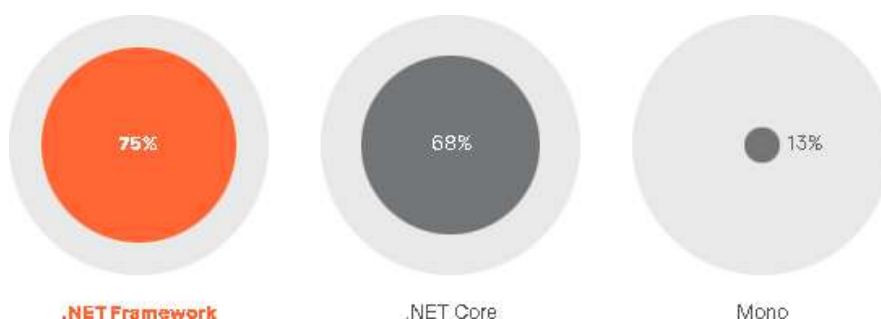
-Servicios Web XML.

-Desarrollo de videojuegos con Unity

¿Qué versiones de C# utiliza habitualmente?

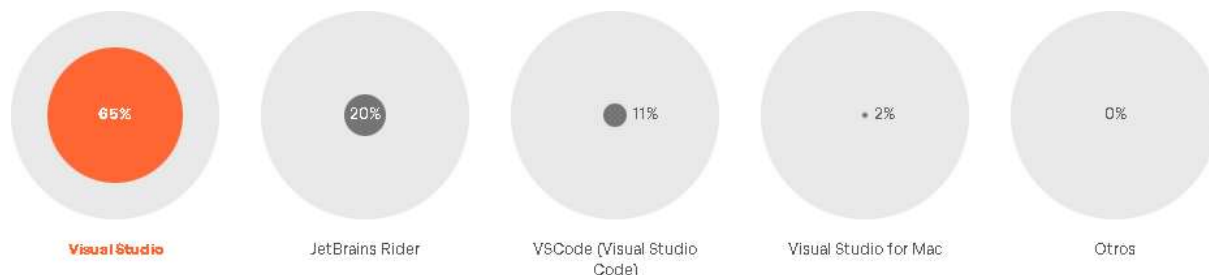
Los desarrolladores C# adoptan las nuevas versiones rápidamente. Más de la mitad de ellos usan C# 8. Parece que muchos empiezan a usar las nuevas funcionalidades del lenguaje en cuanto se publican.

La adopción de .NET Core sigue aumentando. Veremos si el año que viene consiguesuperar a.NETFramework.



Fuente:<https://www.jetbrains.com/es-es/lp/devecosystem-2020/csharp/>

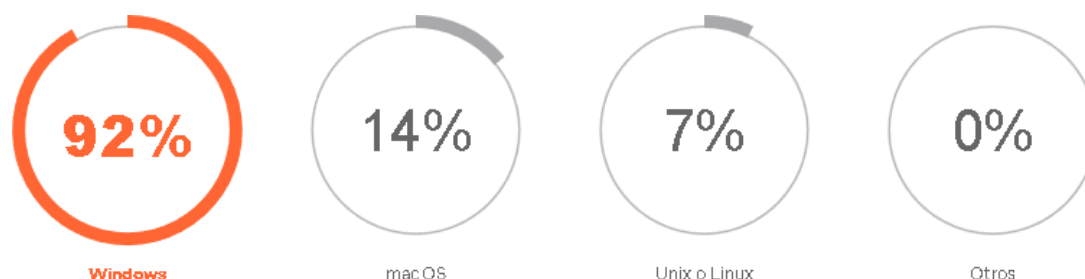
¿Qué editor o entorno de desarrollo integrado utiliza con mayor frecuencia para el desarrollo en C#?



Fuentes: <https://www.jetbrains.com/es-es/lp/devecosystem-2020/csharp/>

¿Qué sistema operativo usa su entorno de desarrollo en C#?

Parece que algunos desarrolladores aprovechan que .NET Core sea multiplataforma y lo utilizan en sistemas operativos que no son Windows. No obstante, aún se trata de un número bajo.



Fuentes: <https://www.jetbrains.com/es-es/lp/devecosystem-2020/csharp/>

2.4 Barra de herramientas

PRINCIPALES PROPIEDADES:

Nombre	Descripción
BottomToolStripPanel	Obtiene la parte inferior ToolStripPanel de este control.
BottomToolStripPanelVisible	Especifica si el fondo ToolStripPanel es visible.
ContentPanel	Obtiene el panel central de este control.
LeftToolStripPanel	Obtiene la izquierda ToolStripPanel de este control.
LeftToolStripPanelVisible	Especifica si el fondo ToolStripPanel es visible.
RightToolStripPanel	Obtiene el derecho ToolStripPanel de este control.
RightToolStripPanelVisible	Especifica si el derecho ToolStripPanel es visible.
TopToolStripPanel	Obtiene la parte superior ToolStripPanel de este control.
TopToolStripPanelVisible	Especifica si la parte superior ToolStripPanel es visible.

Fuente: <https://csharmaniacos.wordpress.com/category/menu-y-barra-de-herramientas/>

EJEMPLO DE USO:



Fuente: <https://csharmaniacos.wordpress.com/category/menu-y-barra-de-herramientas/>

2.5 Nombramiento de cada herramienta

Proporciona barras de herramientas y otros elementos de la interfaz de usuario que admiten varias opciones de apariencia, así como el desbordamiento y la reordenación de elementos en tiempo de ejecución.

PRINCIPALES PROPIEDADES:

Nombre	Descripción
AccessibilityObject	Obtiene AccessibleObject asignado al control.
AccessibleName	Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan.
Font	Obtiene o establece la fuente utilizada para mostrar texto en el control.
ForeColor	Obtiene o establece el color de primer plano del control ToolStrip.
Height	Obtiene o establece el alto del control.
Visible	Obtiene o establece un valor que indica si se muestran el control y todos sus controles secundarios.

Fuentes: <https://csharmaniacos.wordpress.com/category/menu-y-barra-de-herramientas/>

EJEMPLO DE USO:



Fuente: <https://csharmaniacos.wordpress.com/category/menu-y-barra-de-herramientas/>

IDENTIFICANDO EL ELEMENTO:



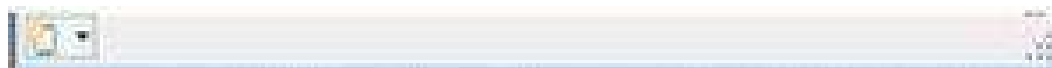
Fuente: <https://csharmaniacos.wordpress.com/category/menu-y-barra-de-herramientas/>

PRINCIPALES PROPIEDADES:

Nombre	Descripción
Locked	La propiedad locked determina si se puede mover o cambiar el tamaño del control.
SizingGrip	Determina si StatusStrip tiene un control de tamaño.
Size	Tamaño de control, en píxeles.
AllowDrop	Obtiene o establece un valor que indica si las operaciones de arrastrar y colocar, y la reordenación de elementos se controlan mediante eventos que implementa.
AutoSize	Obtiene o establece un valor que indica si el control cambia automáticamente de tamaño para mostrar todo su contenido.

Fuentes: <https://csharmaniacos.wordpress.com/category/menu-y-barra-de-herramientas/>

EJEMPLO DE USO:



Fuentes: <https://csharmaniacos.wordpress.com/category/menu-y-barra-de-herramientas/>

2.5 Creación de un cuadro de texto

Este es el código XAML para un cuadro de texto sencillo con un encabezado y texto de marcador de posición.

XML:

```
<TextBox Width="500" Header="Notes" PlaceholderText="Type your notes here"/>
```

Fuente: <https://learn.microsoft.com/es-es/windows/apps/design/controls/text-box>

C#:

```
TextBox textBox = new TextBox();
textBox.Width = 300;
textBox.Header = "Notes";
textBox.PlaceholderText = "Type your notes here";
// Add the TextBox to the visual tree.
rootGrid.Children.Add(textBox);
```

Fuentes: <https://learn.microsoft.com/es-es/windows/apps/design/controls/text-box>

Este es el cuadro de texto que se obtiene de este código XAML.

Notes

Type your notes here

Fuentes:

<https://learn.microsoft.com/es-es/windows/apps/design/controls/text-box>

2.5.1 Instrucciones de control

if-else:

```
if (i ==
```

```
2)
```

```
{
```

```
    // ...
```

```
}
```

```
else if (i == 3)
```

```
{
```

```
    // ...
```

```
}
```

```
else
```

```
{
```

```
    // ...
```

```
}
```

Switch:

```
switch (i)
```

```
{
```

```
    case 1:
```

```
        ...
```

```
        brea
```

```
    k; case
```

```
    2:
```

```
    case 3:
```

...



```
    brea  
k;  
default:  
    ...  
break;  
}
```

For:

```
for (int i = 0; i < 10; ++i)  
{  
    // ...  
}
```

While:

```
while (i <  
10)  
{  
    // ...  
}
```

do-while:do

```
{  
    // ...  
} while (true);foreach:  
foreach (char c in  
charList)  
{  
    // ...  
}
```

2.8 Cuadros de imagen

Por: Erick Estuardo Moran Pech

Normalmente PictureBox, se usa para mostrar gráficos de un mapa de bits, metarchivo, icono, JPEG, GIF o ARCHIVO PNG.

Establezca la Imagen propiedad en el Imagen que desea mostrar, ya sea en tiempo de diseño o en tiempo de ejecución. También puede especificar la imagen estableciendo la ImageLocation propiedad y cargando la imagen de forma sincrónica mediante el Load método o de forma asincrónica mediante el LoadAsync método.

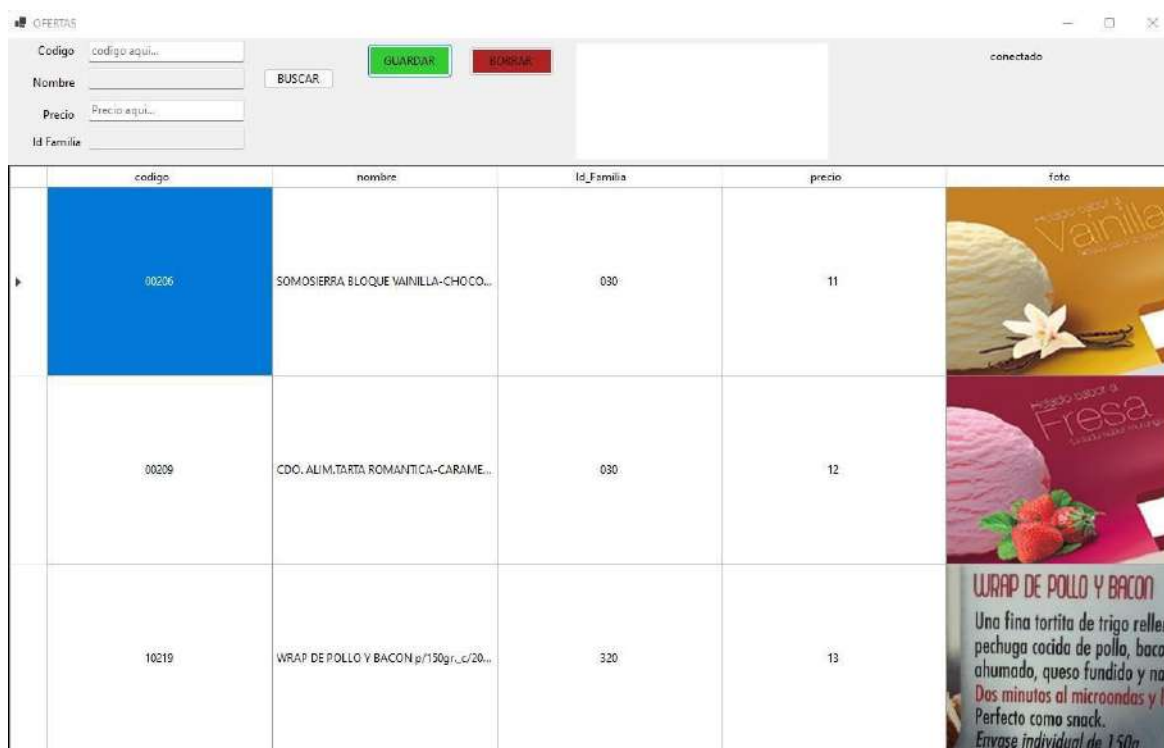


Imagen: cuadros de imagen

Fuente: <https://i.stack.imgur.com/8d1DM.png>

2.9 Botones

El control Button de Windows Forms permite al usuario hacer clic en él para llevar a cabo una acción. Al hacer clic en el botón, parece como si se hubiera presionado y soltado. Cada vez que el usuario hace clic en un botón, se invoca el controlador de eventos Click.

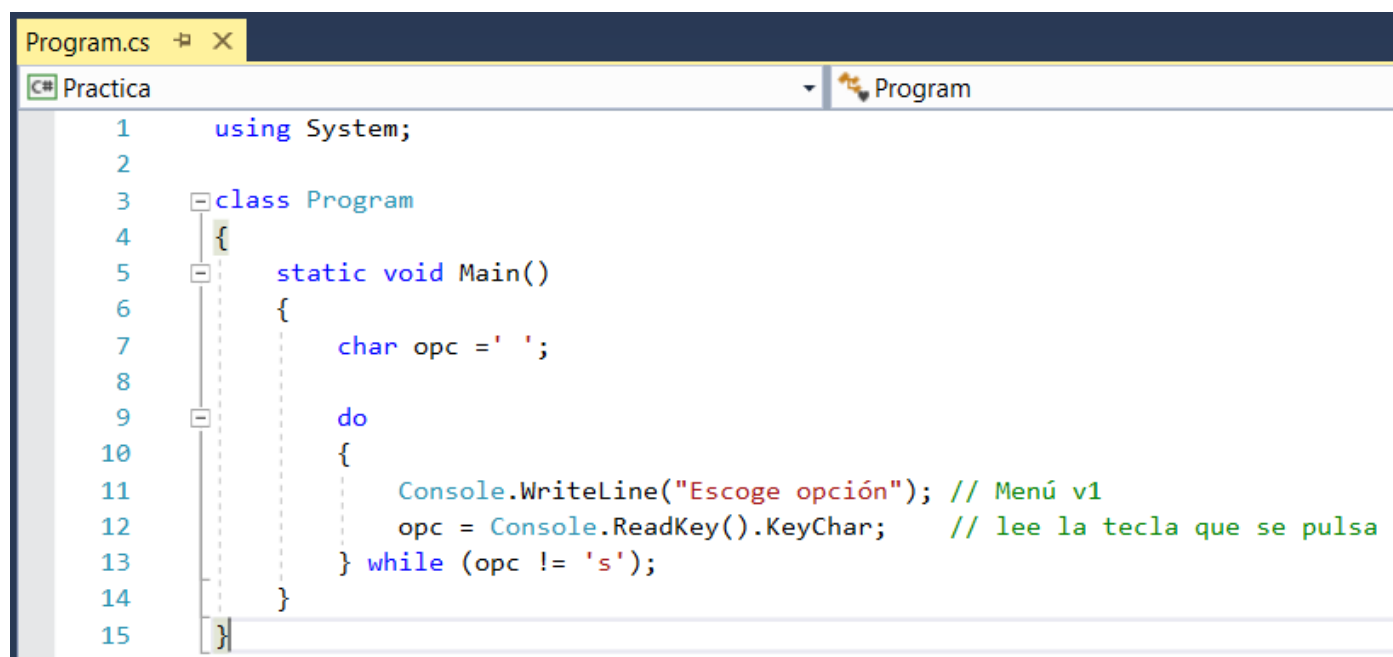


Imagen: botones

Fuente: https://www.elguille.info/colabora/puntonet/miliuco_cerrar_desactivado.jpg

2.10 Menús

Los menús suelen usar una estructura de bucle, es decir, de repetir lo mismo (el menú) hasta que se escoja la opción de salir "s" en nuestro caso. Hay tres bucles, el for, el while, y el do-while, este último es el adecuado para el menú.



```

1  using System;
2
3  class Program
4  {
5      static void Main()
6      {
7          char opc = ' ';
8
9          do
10         {
11             Console.WriteLine("Escoge opción"); // Menú v1
12             opc = Console.ReadKey().KeyChar; // lee la tecla que se pulsa
13         } while (opc != 's');
14     }
15 }

```

Imagen: programación de un menu

Fuente: <https://r2apuntes.es/wp-content/uploads/2018/11/MenuCSharpv1>

2.11 Video

Agregue el control Reproductor multimedia de Windows ActiveX a un formulario. Cambie el tamaño del control y colóquelo donde quiera que aparezca la ventana de vídeo.

Seleccione el control Reproductor multimedia de Windows y cambie la propiedad uiMode a "none". Esta configuración oculta los controles de la interfaz de usuario. Cuando el usuario reproduce un vídeo, aparecerá en la ventana. En el caso del contenido de solo audio, aparecerá una visualización.



Fuente: <https://i.ytimg.com/vi/Vdy6TC9uYcY/hqdefault.jpg>

2.12 Sonido

Para reproducir un sonido en C# utilizaremos la variable soundplayer la cual nos servirá para reproducir archivos en formato wav desde una ruta de acceso local o un identificador uniforme de recursos (URI).

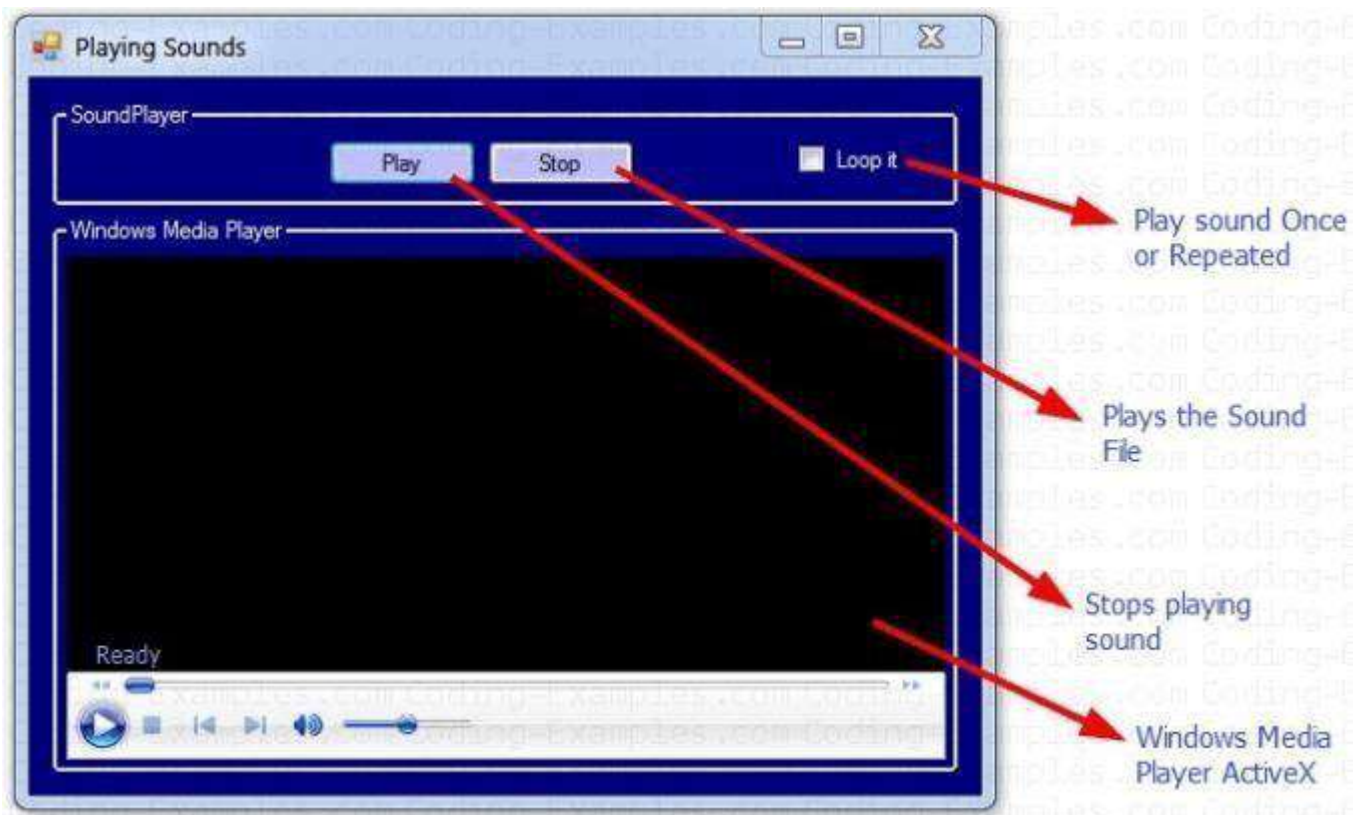


Imagen: reproductor de sonido

Fuente: <https://i0.wp.com/coding-examples.com/wp-content/uploads/2020/04/Windows-Media-Player-Active-X-Control.png?w=691&ssl=1>

2.13 Combo box

El control ComboBox de Windows Forms se usa para mostrar datos en un cuadro combinado desplegable. De forma predeterminada, el control ComboBox aparece en dos partes: la parte superior es un cuadro de texto que permite al usuario escribir un elemento de lista

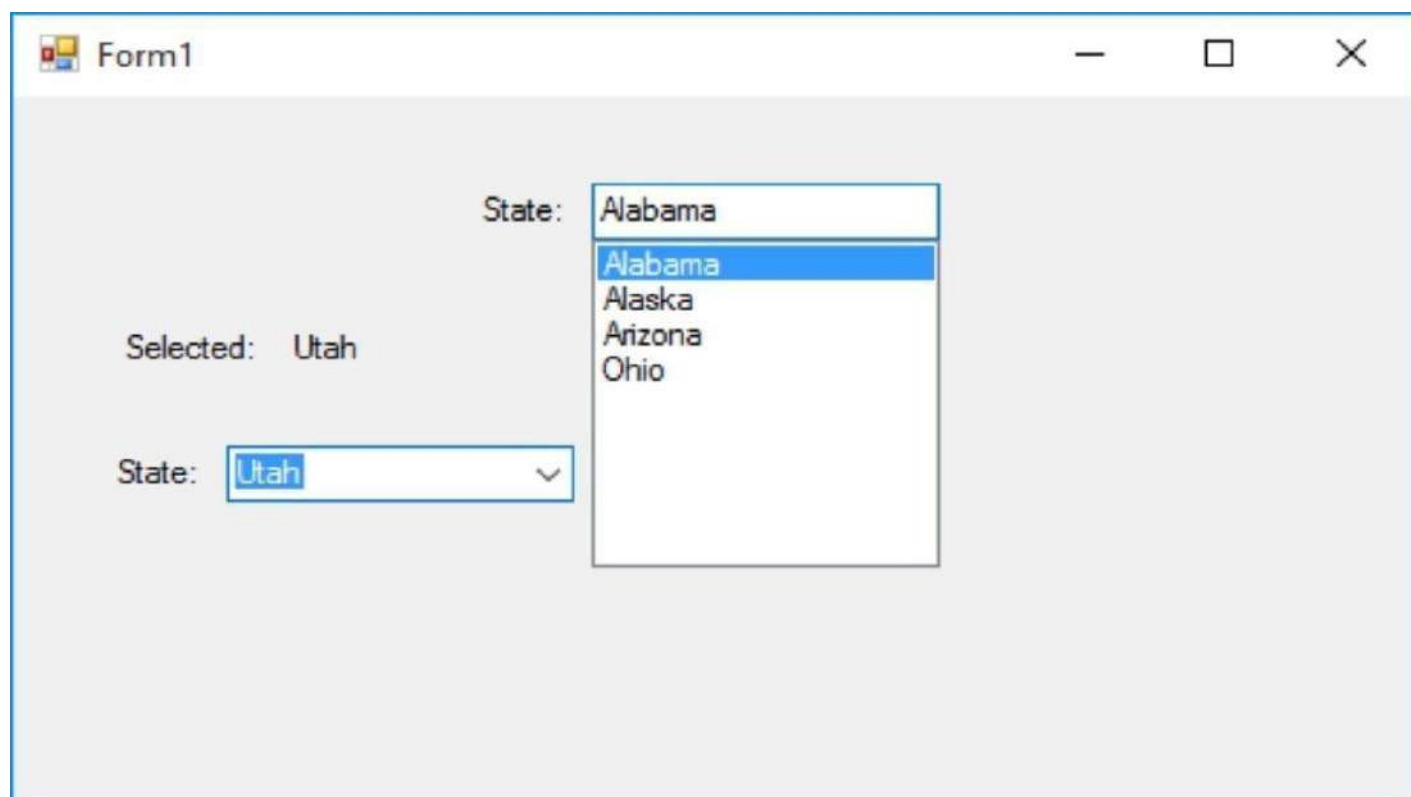


Imagen: Combo box

Fuente: <https://i.ytimg.com/vi/uUdmtgJBjk4/maxresdefault.jpg>

2.14 List box

El ListBox control permite mostrar una lista de elementos al usuario que el usuario puede seleccionar haciendo clic. Un ListBox control puede proporcionar una o varias selecciones mediante la SelectionMode propiedad .

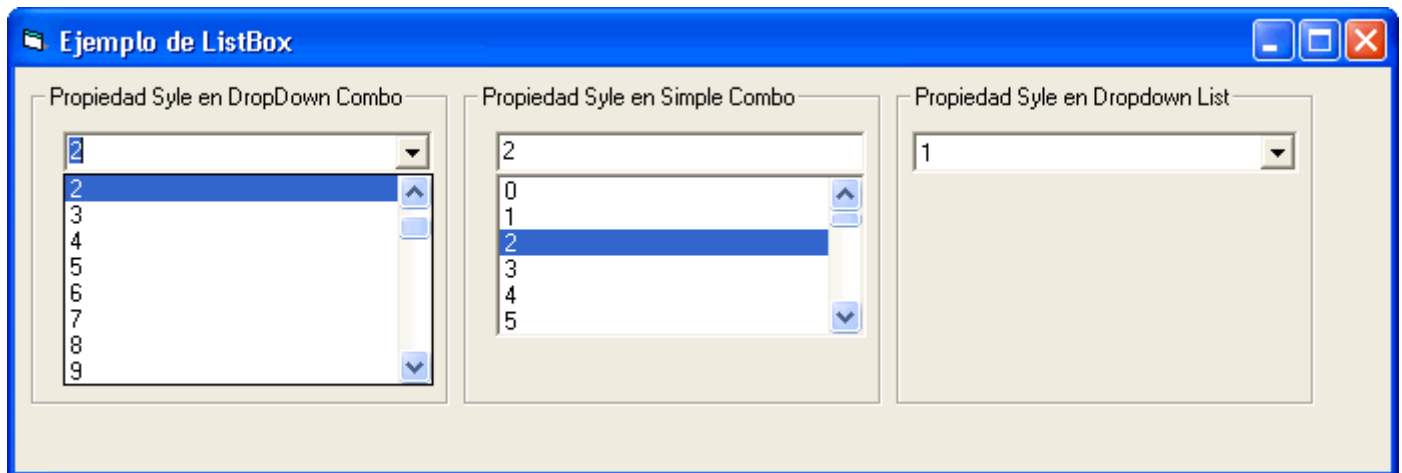


Imagen: list

box Fuente:

http://2.bp.blogspot.com/_FH0pCFRCdo/S_Fv0VvpjWI/AAAAAAAAAIE/adZ053ocrds/s320/combobox_propiedad_style_5.gif

2. Programación en C#

Por: Maria Jose Noj Sor

Microsoft. Visual C#, C# es un lenguaje de programación de alto nivel de propósito general que admite múltiples paradigmas. En ese momento, Microsoft no tenía productos de código

abierto. Para el lenguaje de programación C#..

2.15 Calendarios

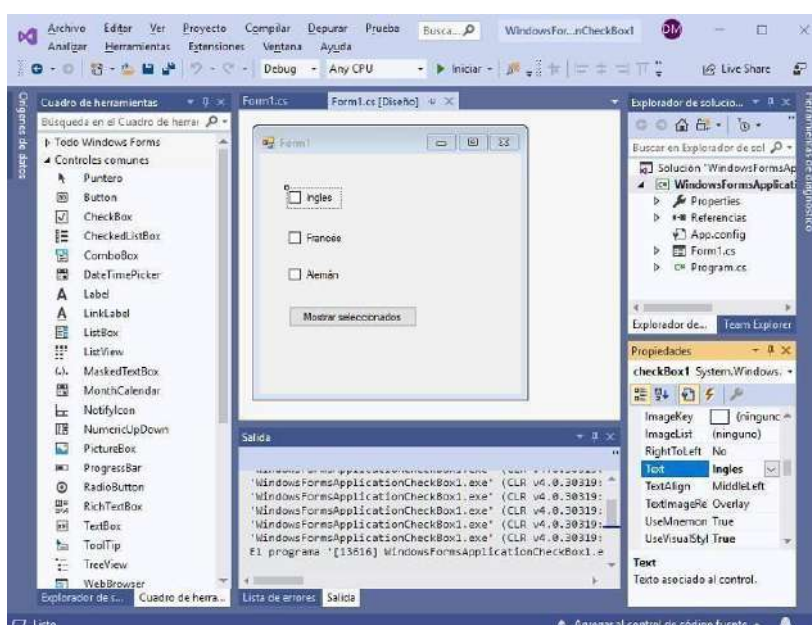
Desde C#, tenemos la posibilidad de **manejar fechas y horas con facilidad**. Para ello, tenemos el tipo de datos DateTime. Por ejemplo, podemos hallar la fecha (y hora) actual con: DateTime fecha = DateTime.Now;

2.16 Radio Buttons

La primera, es la más sencilla, usando la **propiedad** Checked. Por ejemplo, si tuviésemos dos **RadioButton**, uno llamado ConTarjeta y otro ConEfectivo. Bastaría, con chequear cual de estos posee la **propiedad** Checked en true. `if(ConTarjeta.Checked) { MessageBox.Show("Pagó con tarjeta!"); } else { MessageBox.Show("Pagó con efectivo!"); }`

2.17 Check Box

El control CheckBox permite implementar un **cuadro de selección** (básicamente un botón de dos estados: seleccionado o no seleccionado) Problema 1: Confeccionar un programa que muestre 3 objetos de la clase CheckBox con etiquetas de tres idiomas.



Check box en visual studio Fuente:

<https://www.tutorialesprogramacionya.com/csharpya/detalleconcepto.php?codigo=162&inicio=20>

2.18 Timer

La clase Timer se utiliza para **crear un evento después de un intervalo de tiempo establecido** en C#. Podemos generar un evento que se repite después de un período de tiempo específico con la clase Timer en C#. El evento Timer.Elapsed se utiliza para especificar el evento que se repetirá.

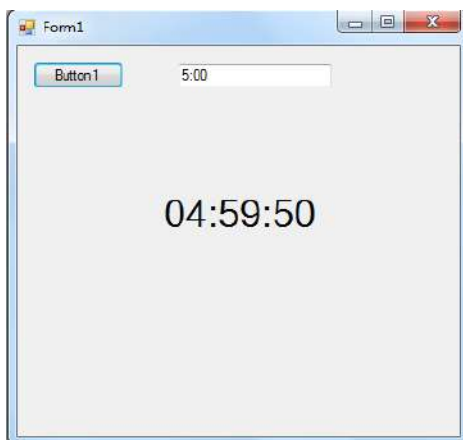


Imagen: Timer en visual studio

Fuente: <https://social.msdn.microsoft.com/Forums/vstudio/en-US/4178d110-fb2f-4ef7-a754-34449c371d65/countdown-timer-in-vb-any-help?forum=vbgeneral>

2.19 Form

Tras seleccionar la plantilla de proyecto de **C#** y asignar un nombre al archivo, Visual Studio abre un formulario automáticamente. Un formulario es una interfaz de usuario de Windows.

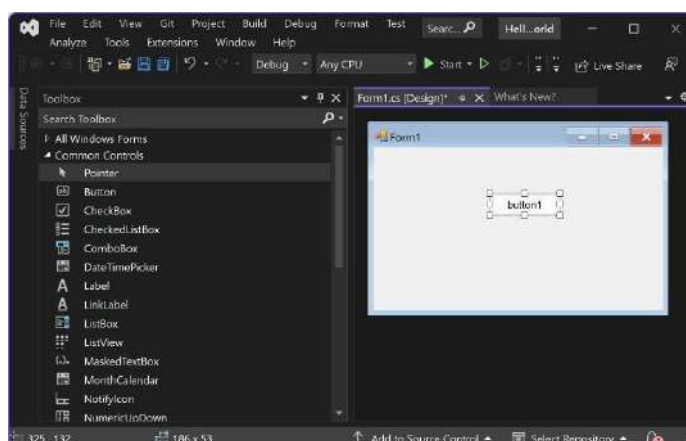


Imagen : form desde visual

studio Fuente: capturado con recortes

2.20 Origen De La Base De Datos

Abre **C#** o VB.NET, crea un nuevo proyecto (Menú Archivo → Nuevo Proyecto Ctrl + Mayús. + N), selecciona Aplicación de Windows Forms y de nombre le pongo "EjemploSencillo". Pulsa el botón Aceptar. Ve al menú Datos → Agregar nuevo **origen** de **datos** y se abrirá un asistente para la configuración de **orígenes** de **datos**.



imagen: Origen de la base de datos

fuelle: <https://karmany.net/images/2013/01-06/tipo-origen-datos.png>

2. Programación en C#

Por: José Antonio Osoy Crespo

2.22 Creación de cadenas

El operador += crea una nueva cadena que contiene el contenido combinado.

```
C#

string s1 = "A string is more ";
string s2 = "than the sum of its chars.";

// Concatenate s1 and s2. This actually creates a new
// string object and stores it in s1, releasing the
// reference to the original object.
s1 += s2;

System.Console.WriteLine(s1);
// Output: A string is more than the sum of its chars.
```

Fuente: <https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/strings/>

2.23 Búsqueda

La búsqueda en C# es un método para encontrar un valor objetivo dentro de una lista. Ésta comprueba secuencialmente cada elemento de la lista para el valor objetivo hasta que es encontrado hasta que todos los elementos hayan sido comparados.

```
using System;
using System.Windows.Forms;
using Microsoft.VisualBasic;

namespace HolaMundo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Humano ingresa el numero a buscar de la lista");
            int numeroBuscar = int.Parse(Console.ReadLine());
            int[] listaNumeros = new int[7];
            for (int i = 0; i < 7; i++)
            {
                Console.WriteLine("Humano ingresa el elemento "+(i+1)+" de la lista");
                int numero = int.Parse(Console.ReadLine());
                listaNumeros[i] = numero;
            }
            int pos = 0;
            bool bandera = false;
            while (pos < 7 && bandera == false) {
                if (numeroBuscar == listaNumeros[pos]) {
                    bandera = true;
                    break;
                }
                pos++;
            }
            if (bandera == true)
            {
                Console.WriteLine("Felicidades humano se encontro el numero " + numeroBuscar + " en la posicion " + (pos+1));
            }
            else {
                Console.WriteLine("Lo siento humano no se encontro el numero "+numeroBuscar);
            }
            Console.Read();
        }
    }
}
```

```

C:\Users\augencia\source\repos\HolaMundo\HolaMundo\bin\Debug\HolaMundo.exe
Humano ingresa el numero a buscar de la lista
4
Humano ingresa el elemento 1 de la lista
6
Humano ingresa el elemento 2 de la lista
5
Humano ingresa el elemento 3 de la lista
4
Humano ingresa el elemento 4 de la lista
3
Humano ingresa el elemento 5 de la lista
2
Humano ingresa el elemento 6 de la lista
1
Humano ingresa el elemento 7 de la lista
1
Felicidades humano se encontro el numero 4 en la posicion 3
    
```

Fuente:<https://learn.microsoft.com/es-es/azure/cognitive-services/bing-web-search/quickstarts/csharp>

2.24 Concatenación

Concatenación es la acción de encadenar conceptos para crear una secuencia, en programación es unir o anexar cadenas, una después de otra. Al programar en varias ocasiones vamos a necesitar unir cadenas, strings, o diferentes tipos de variables, strings con numeros, fechas, etc.

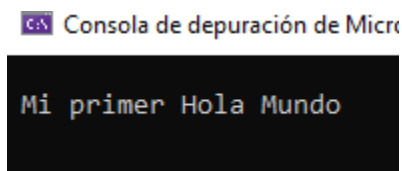
En C# tenemos varias opciones para realizar concatenaciones.

- 2.24.1.1 String.Format
- 2.24.1.2 String.Concat
- 2.24.1.3 String.Join o
- 2.24.1.4 StringBuilder.Append
- 2.24.1.5 El operador "+"
- 2.24.1.6 Interpolación

1. Operador "+"

```

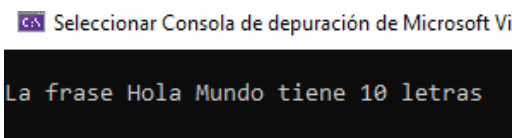
class Program
{
    static void Main(string[] args)
    {
        string variable = "Hola Mundo";
        Console.WriteLine(" Mi primer " + variable);
    }
}
    
```



2. String.Format

```

class Program
{
    static void Main(string[] args)
    {
        string variable = "Hola Mundo";
        Console.WriteLine($"La frase {variable} tiene {variable.Length} letras");
    }
}
    
```



3. Interpolación

```
class Program
{
    static void Main(string[] args)
    {
        string variable = "Hola Mundo";
        Console.WriteLine($"La frase {variable} tiene {variable.Length} letras");
    }
}
```

Seleccionar Consola de depuración de Microsoft \

La frase Hola Mundo tiene 10 letras

Fuente: <https://dev.to/veronicaguamann/3-formas-de-concatenar-string-con-c-d2c>

2.25 Comparación

En C# tenemos dos formas básicas de comparar objetos. Por un lado el operador de igualdad (==) y por otro el método *Equals* que se define en *Object* y por lo tanto está disponible en cualquier objeto.

```
1. class A
2. {
3.     public int Value { get; set; }
4. }
5.
6. class Program
7. {
8.     static void Main(string[] args)
9.     {
10.         var a1 = new A() { Value = 42 };
11.         var a2 = new A() { Value = 42 };
12.         var equals = a1.Equals(a2);
13.     }
14. }

1. public override bool Equals(object obj)
2. {
3.     return obj is A a ? a.Value == Value : base.Equals(obj);
4. }
```

Fuente: <https://geeks.ms/etomas/2018/12/17/comparaciones-en-c/#:~:text=Sabemos%20que%20en%20C%23%20tenemos,est%C3%A1%20disponible%20en%20cualquier%20objeto.>

2.26 Generación de subcadenas

Una subcadena es cualquier secuencia de caracteres que se encuentra en una cadena. Use el método *Substring* para crear una nueva cadena de una parte de la cadena original.

```

string s3 = "Visual C# Express";
System.Console.WriteLine(s3.Substring(7, 2));
// Output: "C#"

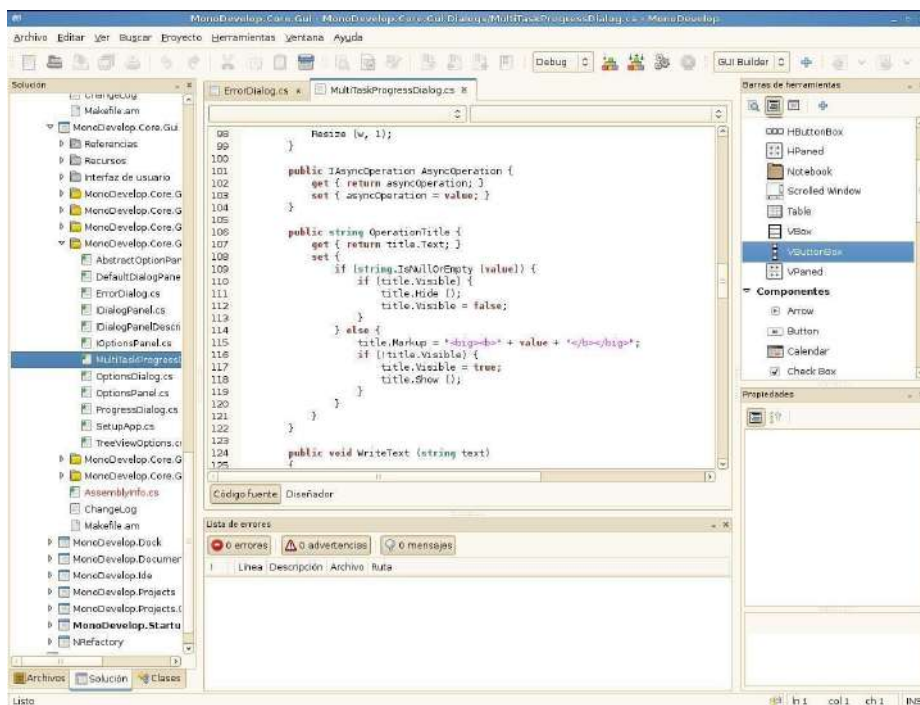
System.Console.WriteLine(s3.Replace("C#", "Basic"));
// Output: "Visual Basic Express"

// Index values are zero-based
int index = s3.IndexOf("C");
// index = 7
    
```

Fuente: <https://learn.microsoft.com/es-es/dotnet/api/system.string.substring?view=net-7.0>

2.27 Implementación de entorno

Permiten a los desarrolladores extraer credenciales confidenciales de su código fuente y utilizar diferentes variables de configuración en función de su entorno de trabajo.



Fuente: learn.microsoft.com/es-es/aspnet/web-forms/overview/deployment/configuring-server-environments-for-web-deployment/configuring-deployment-properties-for-a-target-environment

2.29 Creación de base de datos en Access

Introducción

Por: Diego Racanac

El programa Microsoft Access, es un programa que se encarga de la administración de las bases de datos. El mismo te ayudará a organizar y administrar un buen número de registros. Con esta herramienta también se podrá diseñar aplicaciones que se puedan emplear en Access, en la modalidad Backend. Ya que es un software de gestión de bases de datos desarrollado por Microsoft. Es una herramienta poderosa que permitela creación, el mantenimiento y el análisis de bases de datos de manera fácil y eficiente.

Pasos para crear una base de datos en Access

- 3.1. Abrir Microsoft Access y seleccionar "Crear una base de datos nueva".
- 3.2. Elegir el tipo de base de datos que se desea crear (por ejemplo, una base de datos de tablas simples o una base de datos de clientes/proveedores) y seleccionar la plantilla apropiada.
- 3.3. Nombrar la base de datos y elegir una ubicación en la que se almacenará.
- 3.4. Crear tablas en la base de datos para almacenar los datos. Esto implica definir los campos que se requieren y establecer relaciones entre las tablas.
- 3.5. Ingresar datos en las tablas creadas.
- 3.6. Crear consultas para recuperar los datos de la base de datos de manera eficiente.
- 3.7. Crear formularios y informes para presentar los datos de manera atractiva y fácil de entender. Guardar y cerrar la base de datos.

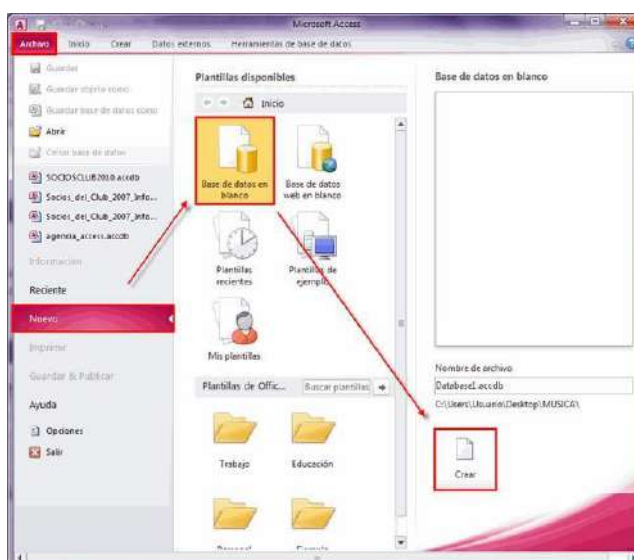


Imagen: Creación de base de datos

Fuente: <https://ti-h5.jimdofree.com/unidad-3-herramientas-de-ofimática/3-4-1-crear-y-usar-una-base-de-datos/>

2.30 Relaciones entre tablas de Microsoft Access

Las relaciones entre tablas son un aspecto fundamental de la creación y gestión de bases de datos en Microsoft Access. Las relaciones permiten vincular tablas entre sí para crear una estructura de base de datos coherente y eficiente.

- 2.1. ¿Qué son las relaciones entre tablas? Las relaciones entre tablas son vínculos que se establecen entre dos o más tablas para indicar cómo se relacionan los datos almacenados en ellas. Por ejemplo, puede haber una relación entre una tabla que contiene información sobre clientes y otra tabla que contiene información sobre pedidos realizados por esos clientes.
- 2.2. ¿Por qué son importantes las relaciones entre tablas? Las relaciones entre tablas son importantes porque permiten evitar la redundancia de datos y garantizar la integridad de los datos. Además, facilitan la consulta y el análisis de la información de la base de datos.
- 2.3. ¿Cómo se establecen las relaciones entre tablas en Access? Para establecer una relación entre tablas en Access, debe identificar un campo común en ambas tablas que pueda utilizarse como clave primaria y clave foránea. Luego, debe seleccionar ambas tablas en la pestaña "Base de datos" y hacer clic en "Relaciones". Finalmente, arrastre el campo clave primaria a la tabla que contiene el campo clave foránea y haga clic en "Crear relación".

En resumen, las relaciones entre tablas son un aspecto fundamental para la creación y gestión de bases de datos eficientes en Microsoft Access. Al establecer relaciones entre tablas, se puede garantizar la integridad de los datos, evitar la redundancia y facilitar la consulta y el análisis de la información.

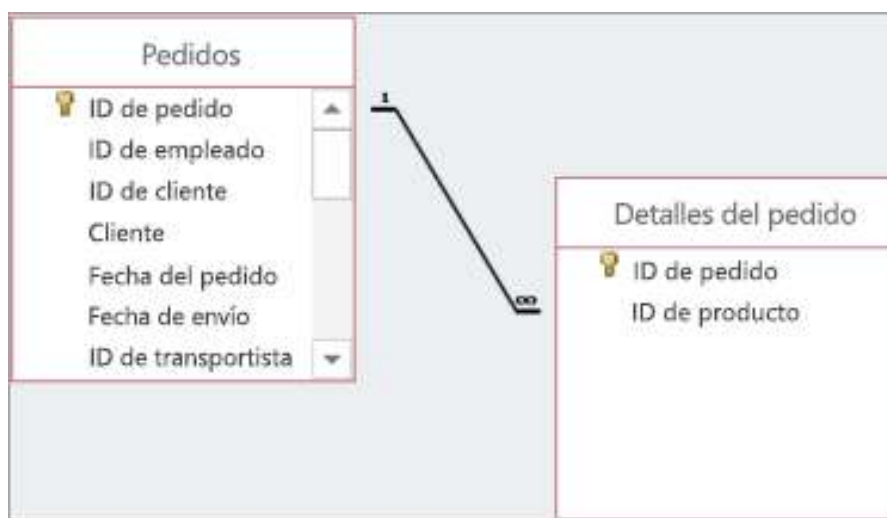


Imagen: Relaciones entre tablas

Fuente: <https://support.microsoft.com/es-es/office/vídeo-introducción-a-las-relaciones-de-tabla-728d53ff-f332-4ac6-9382-574ee271500a>

2.31 Implementación de llaves primarias en Access

Las llaves primarias son un aspecto fundamental de la creación y gestión de bases de datos en Microsoft Access. Las llaves primarias se utilizan para identificar de manera única cada registro en una tabla. Ya que una clave principal es un campo o un conjunto de campos con valores únicos en una tabla. Los valores de la clave pueden usarse para hacer referencia a registros completos, porque cada registro tiene un valor diferente para la clave.

¿Cómo funciona?

- 2.1. ¿Qué son las llaves primarias? Las llaves primarias son campos o conjuntos de campos en una tabla que se utilizan para identificar de manera única cada registro en la tabla. Cada registro en una tabla debe tener un valor único en el campo de la llave primaria.
- 2.2. ¿Por qué son importantes las llaves primarias? Las llaves primarias son importantes porque garantizan la integridad de los datos al asegurar que cada registro en una tabla sea único e identificable. Además, las llaves primarias se utilizan para establecer relaciones entre tablas y para garantizar que los datos sean consistentes y coherentes.
- 2.3. ¿Cómo se implementan las llaves primarias en Access? Para implementar una llave primaria en Access, primero debe crear una tabla o seleccionar una tabla existente. Luego, debe seleccionar el campo o conjunto de campos que desea utilizar como llave primaria y hacer clic en la pestaña "Diseño de tabla". Finalmente, haga clic en la opción "Marcar como llave primaria" o seleccione la opción "Definir llave primaria" en el menú "Tabla".

En resumen, las llaves primarias son un aspecto fundamental de la creación y gestión de bases de datos en Microsoft Access. Al implementar llaves primarias, se puede garantizar la integridad de los datos y facilitar la creación de relaciones entre tablas para crear una estructura de base de datos coherente y eficiente.

2.32 Implementación de llaves foráneas en Access

En el contexto de bases de datos relacionales, una clave foránea o llave foránea o clave ajena (o Foreign Key FK) es una limitación referencial entre dos tablas. La clave foránea identifica una columna o grupo de columnas en una tabla (tabla hija o referendo) que se refiere a una columna o grupo de columnas en otra tabla (tabla maestra o referenciada). Las columnas en la tabla referendo deben ser la clave primaria u otra clave candidata en la tabla referenciada. La clave foránea identifica una columna(s) en una TABLA REFERENCIANTE a una columna(s) en la TABLA REFERENCIADA. Ahora una explicación de que son y como implementarlas:

- 2.1. ¿Qué son las llaves foráneas? Las llaves foráneas son campos o conjuntos de campos en una tabla que se relacionan con un campo o conjunto de campos en otra tabla. Esta relación se establece para garantizar la integridad de los datos y para asegurar que los datos en diferentes tablas estén sincronizados.
- 2.2. ¿Por qué son importantes las llaves foráneas? Las llaves foráneas son importantes porque ayudan a garantizar la integridad de los datos y a asegurar que los datos en diferentes tablas estén sincronizados. Además, las llaves foráneas facilitan la creación de consultas y informes que involucren datos de varias tablas.
- 2.3. ¿Cómo se implementan las llaves foráneas en Access? Para implementar una llave foránea en Access, primero debe crear dos tablas con los datos relacionados. Luego, seleccione la tabla en la que desea establecer la llave foránea y abra la pestaña "Diseño de tabla". A continuación, seleccione el campo o conjunto de campos que desea utilizar como llave foránea y haga clic en la opción "Relaciones". Finalmente, seleccione la tabla relacionada y el campo de la llave primaria y haga clic en "Aceptar".

Al implementar llaves foráneas, se puede garantizar la integridad de los datos y establecer relaciones entre tablas para crear una estructura de base de datos coherente y eficiente.

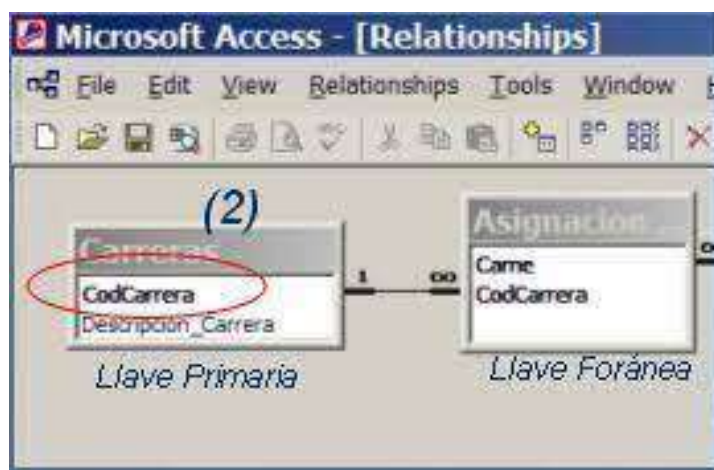


Imagen: Llave foranea y primaria

2.33 Conexión a base de datos en Access

La conexión a una base de datos en Microsoft Access permite acceder y utilizar los datos almacenados en una base de datos desde la aplicación Access. Esto significa que se puede acceder a los datos de forma más rápida y eficiente, compartir y utilizar los datos con múltiples usuarios y aplicaciones, y mejorar el control y la gestión de los datos.

¿Cómo crear conexión?

Además, la conexión a una base de datos en Access también permite realizar tareas avanzadas, como la creación de informes, consultas y formularios con los datos almacenados en la base de datos. Esto mejora la presentación y el análisis de los datos. La conexión a una base de datos en Microsoft Access es una herramienta clave para acceder, utilizar, compartir y analizar los datos almacenados en una base de datos.

- 2.1 ¿Qué es una conexión a una base de datos? Una conexión a una base de datos es un canal de comunicación entre una aplicación y una base de datos. La conexión permite a la aplicación acceder y utilizar los datos almacenados en la base de datos.
- 2.2 ¿Cómo se crea una conexión a una base de datos en Access? Para crear una conexión a una base de datos en Access, siga los siguientes pasos:
- 2.3 Abra Access y haga clic en "Archivo" y "Nuevo".
- 2.4 Seleccione "Base de datos en blanco" y haga clic en "Crear".
- 2.5 Haga clic en "Externa de datos" y seleccione "SQL Server".
- 2.6 Ingrese la información de conexión, como el nombre del servidor, el nombre de la base de datos, el nombre de usuario y la contraseña.
- 2.7 Haga clic en "Conectar".
- 2.8 ¿Qué ventajas ofrece una conexión a una base de datos en Access? Una conexión a una base de datos en Access ofrece las siguientes ventajas:
- 2.9 Acceso a datos más rápido y eficiente.
- 2.10 Posibilidad de compartir y utilizar los datos con múltiples usuarios y aplicaciones.
- 2.11 Mejor control y gestión de los datos.

En resumen, la conexión a una base de datos en Microsoft Access es un paso esencial para acceder y utilizar los datos almacenados en una base de datos. Al crear una conexión, se pueden acceder y utilizar los datos de forma más rápida y eficiente, compartir y utilizar los datos con múltiples usuarios y aplicaciones, y mejorar el control y la gestión de los datos.

2.34 Implementación de consultas en Access

La implementación de consultas en Microsoft Access permite acceder y utilizar los datos almacenados en una base de datos de manera más eficiente y efectiva. Las consultas permiten filtrar, ordenar y resumir los datos según las necesidades específicas de un usuario o aplicación, lo que mejora la accesibilidad y la utilidad de los datos.

Creación de una implementación de consulta

Además, la implementación de consultas también permite realizar análisis más avanzados de los datos, como la agrupación y la sumación, lo que mejora la capacidad de tomar decisiones informadas basadas en los datos almacenados en la base de datos. La implementación de consultas en Microsoft Access es una herramienta clave para mejorar la accesibilidad, la utilidad y la utilización de los datos almacenados en una base de datos.

- 2.1 ¿Qué es una consulta en Access? Una consulta en Access es una solicitud específica a la base de datos que permite seleccionar, filtrar, ordenar y resumir los datos almacenados en una base de datos.
- 2.2 ¿Cómo se crea una consulta en Access? Para crear una consulta en Access, siga los siguientes pasos:
- 2.3 Abra Access y haga clic en "Crear" y "Consulta en diseño".
- 2.4 Seleccione las tablas y los campos que desea incluir en la consulta.
- 2.5 Agregue cualquier filtro o condición para limitar los resultados de la consulta.
- 2.6 Especifique cómo desea ordenar los resultados.
- 2.7 Guarde la consulta y ejecútela.
- 2.8 ¿Qué ventajas ofrece la implementación de consultas en Access? La implementación de consultas en Access ofrece las siguientes ventajas:
- 2.9 Fácil acceso a los datos específicos que se necesitan.
- 2.10 Posibilidad de resumir y analizar los datos de forma más eficiente.
- 2.11 Mejora la presentación de los datos.

En resumen, las consultas en Microsoft Access son una herramienta esencial para filtrar, ordenar y resumir los datos almacenados en una base de datos. Al crear y utilizar consultas en Access, se pueden acceder a los datos específicos que se necesitan, resumir y analizar los datos de forma más eficiente, y mejorar la presentación de los datos.

2.35 Formularios de registros en Access

Los formularios de registro en Microsoft Access sirven para capturar y editar los datos en una base de datos de manera más eficiente y amigable para los usuarios. Los formularios de registro incluyen controles, como cuadros de texto, listas desplegables y botones de opción, que permiten a los usuarios ingresar y editar los datos de manera intuitiva y precisa. Además, los formularios de registro pueden ayudar a mejorar la calidad y la integridad de los datos al proporcionar controles para validar y verificar la información ingresada. En resumen, los formularios de registro en Access son una herramienta clave para administrar y mantener la información en una base de datos.

¿Qué son los formularios?

- 2.1 ¿Qué son los formularios de registro en Access? Los formularios de registro en Access son una interfaz de usuario que permite a los usuarios ingresar y editar los datos en una base de datos. Los formularios de registro pueden incluir controles, como cuadros de texto, listas desplegables y botones de opción, que ayudan a los usuarios a ingresar y editar los datos de manera eficiente.
- 2.2 ¿Cómo se crea un formulario de registro en Access? Para crear un formulario de registro en Access, siga los siguientes pasos:
- 2.3 Abra Access y haga clic en "Crear" y "Formulario en diseño".
- 2.4 Seleccione las tablas y los campos que desea incluir en el formulario.
- 2.5 Agregue cualquier control adicional, como cuadros de texto, listas desplegables o botones de opción, para ayudar a los usuarios a ingresar y editar los datos.
- 2.6 Guarde el formulario y cierre el diseñador.
- 2.7 ¿Qué ventajas ofrece la implementación de formularios de registro en Access? La implementación de formularios de registro en Access ofrece las siguientes ventajas:
- 2.8 Permite a los usuarios ingresar y editar los datos de manera eficiente.
- 2.9 Proporciona una interfaz de usuario amigable para los usuarios que no están familiarizados con la base de datos.
- 2.10 Mejora la calidad y la integridad de los datos al proporcionar controles para asegurar la precisión y la consistencia de los datos ingresados.

En resumen, los formularios de registro en Microsoft Access son una herramienta útil para capturar y editar los datos en una base de datos. Al crear y utilizar formularios de registro, se pueden ingresar y editar los datos de manera eficiente, proporcionar una interfaz de usuario amigable y mejorar la calidad y la integridad de los datos.

TECNOLOGÍA

BIMESTRE 3



3. Programación Básica V.B (consola).

Por: Nohelia Guadalupe de los Ángeles Culajay Oxcal.

3.1.1 Definiciones

Visual Basic es un lenguaje de programación con seguridad de tipos de diseño para aprender. Una aplicación de consola toma la entrada y muestra salida en una ventana de líneas de comandos, también conocida como consola.

Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo

Visual Basic es un **lenguaje de programación orientado a objetos** desarrollado por Microsoft. El uso de Visual Basic agiliza y simplifica la creación de aplicaciones . NET con seguridad de tipos.

Es una aplicación que se desarrolla en una ventana **MS-DOS** y en donde no hay un formulario ni una interfaz de usuario por lo que los programas se construyen y ejecutan usando línea de comandos.

3.1.2 Sintaxis.

Es el conjunto de reglas que define las combinaciones de símbolos que se consideran declaraciones o expresiones correctamente estructuradas en ese lenguaje. Esto se aplica tanto a los lenguajes de programación, donde el documento representa el código fuente, como a los lenguajes de marcado, donde el documento representa los datos.

La sintaxis de un tema de Ayuda de Visual Basic para un método, instrucción o función **muestra todos los elementos necesarios para usar la función, instrucción o método correctamente.**

Los comandos de consola **Red, ReadLine, Write, WriteLine** que sirven para leer valores del teclado y mostrarlos en pantalla.

La sintaxis del método activar, la palabra cursiva "objeto" es un marcador de posición para la información que suministre. Deben escribirse las palabras que estén en negrita exactamente tal y como aparecen.

```
VB
Sub MakeActive()
    Windows(2).Activate
End Sub
```

En la sintaxis de instrucción Dim , la palabra Dim es la palabra clave obligatoria. El único elemento obligatorio es varname. (el nombre de la variable).

VB

```
Dim myAnswer As String
```

Fuente: <https://learn.microsoft.com/es-es/dotnet/visualbasic/programming-guide/language-features/statements>

3.1.3 Palabras reservadas.

AddHandler	Char	Default
AddressOf	CInt	Delegate
Alias	Class (Restricción)	Dim
And	Class (Instrucción)	DirectCas
AndAlso	CLng	† Do
As	CObj	Double
Boolean	Const	Each
ByRef Byte	Continue	Else
ByVal	CByte	Elseif
Call	CShort	End (Instrucción)
Case	C\$ng	End<keyword>
Catch	CStr	EndIf
CBool	CType	Enum
CByte	CUInt	Erase
CChar	CULng	Error
CDate	CUShort	Event
Cdbl	Date	Exit
CDec	Decimal	False
	Declare	
Finally	Friend	GetType
For (en For...Next)	Function	GetXMLNamespace
For Each...Next	Get	Global

GoSub	On	<<=
GoTo	Operator	Throw
If()	Option	To
Implements	Optional	True
Implements (Instrucción)	Or	Try
Imports (Espacio de nombres y tipo .NET)	OrElse	TryCast
Imports (Espacio de nombres XML)	Out (Modificador genérico)	TypeOf...Is
In	Overloads	UInteger
In (Modificador)	Overridable	ULong
Inherits	Overrides	UShort
Integer	ParamArray	Using
Interface	Partial	Variant
Is	Private	Wend
IsNot	Property	When
Let	Protected	While
Lib	Public	Widening
Like	RaiseEvent	With
Long	ReadOnly	WithEvents
Loop	ReDim	WriteOnly
Me	REM	Xor
Mod	RemoveHandler	#Const
Module	Resume	#Else
Module (Instrucción)	Return	#Elseif
MustInherit	SByte	#End
MustOverride	Select	#If
MyBase	Set	=
MyClass	Shadows	&
	Shared	&=
	Short	

NameOf	Single	*
Namespace	Static	*=
Narrowing	Step	/
New (Restricción)	Stop	/=
Operador New	String	\
Next	Structure (Restricción	\=
Next (en Resume))	^
Not	Structure (Instrucción	^=
Nothing)	+
NotInheritable	Sub	+=
NotOverridable	SyncLock	--=
Object	Then	Operador >>
Of		Operador >>=
		<<

Fuente: <https://learn.microsoft.com/es-es/dotnet/visual-basic/language>

3.1.4 Variables.

Las variables son usadas dentro de ámbitos, el ámbito indica en qué lugar de la aplicación desea utilizar la variable y puede ser una de las siguientes palabras claves. **Dim, Static, Private o Public.**

Dim: Se puede utilizar a nivel de procedimiento y a nivel del formulario. Si se declara una variable a nivel de procedimiento con la palabra Dim, **Visual Basic** reinicializa (borra su contenido) la **variable** cada vez que ejecuta el procedimiento.

Ejemplo:

Di valor As Integer

Static: Se utiliza solo a nivel del procedimiento. Cuando una variable se declara con la palabra **Static, Visual Basic** no reinicializa la variable cada vez que se ejecuta el procedimiento, solo lo hace la primera vez. Esto quiere decir que la variable conserva su valor entre una llamada al procedimiento y otra.

Ejemplo: Static x As Double

Private: Se utiliza solo a nivel del módulo e indica que la variable es privada; es decir, solo va a ser reconocida dentro del módulo donde ha sido declarada.

Ejemplo:

Private c As String

Public: Se utiliza solo a nivel de módulo e indica que la variable es pública; es decir, va a ser reconocida en toda la aplicación.

Public d As Date

Fuente: <https://www.aulafacil.com/cursos/programacion/visual-basic>



3.1.5 Constantes.

Es un elemento con un nombre determinado que mantiene un valor constante durante el desarrollo de todo el programa el cual no se puede cambiar. Las constantes se definen con la instrucción **Const** por ejemplo:

Const e = 10

Nota: La definición de las constantes puede empezar con las palabras **Public** o **Private**. Para que sean públicas o privadas. Por defecto son privadas.

Comandos:

COMANDO	FUNCION	FORMA DE USO
Read	Captura o lee el dato que el usuario ingresa.	VariableIngresada = Console.Read
ReadLine	Captura o lee el dato que el usuario ingresa, pero a la vez hace un cambio de línea al terminar de leer el dato.	Variable = Console.ReadLine
Write	Para mostrar texto y datos calculados o capturados	Console.Write(.Texto a Mostrar {0},{1}, {2}, ..., {n}., Var0, Var1, Var2,..., Varn) Console.Write(.Texto a Mostrar.)
WriteLine	Permite mostrar ya sea una cadena texto o un dato numérico, y a la vez hace un cambio de línea, después de haber mostrado el dato.	Console.WriteLine(.Texto a Mostrar {0},{1}, {2}, .. {n}., Var0, Var1, Var2,..., Varn) Console.WriteLine(.Texto a Mostrar.)

Fuente: <https://www.microsoft-visualstudio.com/2011/07/glosario-determinos-de-visual>

3.2 Tipos de datos

Por: Kevin Alexander De Leon Andrade

Los tipos de datos son categorías que describen el tipo de información que puede ser almacenada y manipulada por un programa de computadora. Algunos de los tipos de datos más comunes son enteros, decimales y caracteres.

3.2.1 Entero

Representa números enteros con o sin signo, que estarán compuestos por los dígitos del 0 al 9, pudiendo ser adelantados por los signos + o -. Algún ejemplo de dato enteros son: 0, 23, -176, -1, etc. Para definir un tipo de dato se escribe lo siguiente:

```
int nombre_variable = valor;
```

3.2.2 Decimal

Se emplean para representar números reales (con decimales). Para definir datos reales se antepone la palabra reservada *float* al identificador de la variable.

```
float identificador = valor;
```

```
Por ejemplo: float numero1, numero2; float numero3  
= 123.43;  
float numero3;
```

3.2.3 Caracter

Este tipo de datos se emplea para representar un carácter perteneciente a un determinado código utilizado por el ordenador (normalmente el código ASCII). Para representar este tipo de dato se adelanta la palabra reservada *char* al identificador de la variable.

```
Char identificador = 'valor';
```

Un constante tipo char se representa como un solo carácter encerrado entre comillas simples.

```
Por ejemplo: char letra, letra2; char letra='a';
```




Tipo de dato	Palabra reservada	Ejemplo
Entero	int	int numero=0;
Real	float	float numero=12.2;
Carácter	char	char letra = 'a';

Imagen: Tipos de datos + palabra reservada + ejemplo

Fuente: <http://recursostic.educacion.es/observatorio/web/en/software/programacion/972-tiposdedatos#:~:text=Tipo%20entero%3A%20representa%20n%C3%BAmeros%20enteros,al%20identificador%20de%20la%20variable.>



3.3 Operadores Aritméticos

Por: **Marta Alicia Del Carmen Díaz Cifuentes.**

Permiten realizar operaciones aritméticas sobre los operando. Cada operación realizada con los operando genera un resultado que se almacenara en alguna posición de memoria. El tipo de valor generado y por tanto su representación interna, dependerá de los tipos de los operando.

En C++ y otros muchos lenguajes, se sigue el orden normal establecido en la aritmética, realizándose las operaciones de izquierda a derecha siguiendo el siguiente orden:

- Se evalúan las expresiones entre paréntesis.
- Potencias (C++ no dispone de un operador nativo, pero en otros lenguajes como Python sí)
- Multiplicación, división y resto de la división entera
- Suma y resta.

3.3.1 Suma, Resta, Multiplicación, División

Operador de Suma

Calcular el total de dos o más números. Su símbolo es “+”

📄 Código C++ 📄

```
1 int numero1 = 10; LF
2 int numero2 = 10; LF
3 int suma = numero1 + numero2; LF
4 LF
5 cout << suma;
```

📄 Resultado en la consola 📄

```
1 | 20
```

Imagen: Como usar la suma.

Fuente: <https://www.programacionfacil.org/cursos/c++/capitulo-8-operadores-aritmeticos.html>

Operador de Resta

Se usa para encontrar la diferencia de dos números. Su signo es “-”

Código C++

```
1 int numero1 = 10; LF
2 int numero2 = 10; LF
3 int resta = numero1 - numero2; LF
4 LF
5 cout << resta;
```

Resultado en la consola

```
1 | 0
```

Imagen: Como usar la resta.

Fuente: <https://www.programacionfacil.org/cursos/c++/capitulo-8-operadores-aritmeticos.html>

Operador de Multiplicación

Hallar el resultado de sumar un número tantas veces como indique el otro. Su signo es “*”

Código C++

```
1 int numero1 = 10; LF
2 int numero2 = 10; LF
3 int multiplicacion = numero1 * numero2; LF
4 LF
5 cout << multiplicacion;
```

Resultado en la consola

```
1 | 100
```

Imagen: Como usar la multiplicación.

Fuente: <https://www.programacionfacil.org/cursos/c++/capitulo-8-operadores-aritmeticos.html>

Operador de División

Permite distribuir en partes iguales. Su signo es “/”

Código C++

```
1 int numero1 = 10; LF
2 int numero2 = 3; LF
3 int division = numero1 / numero2; LF
4 LF
5 cout << division;
```

Resultado en la consola

El resultado de la división es 3. El resto no me lo da.

```
1 | 3
```

Imagen: Como usar la división.

Fuente: <https://www.programacionfacil.org/cursos/c++/capitulo-8-operadores-aritmeticos.html>

3.3.2 Operadores de igualdad y relacionales.

Los operadores relacionales, también denominados operadores binarios lógicos y de comparación, se utilizan para comprobar la veracidad o falsedad de determinadas propuestas de relación. Las expresiones que los contienen se denominan expresiones relacionales. Aceptan diversos tipos de argumentos, y el resultado, que es la respuesta a la pregunta, es siempre del tipo cierto/falso, es decir, producen un resultado booleano. Si la propuesta es cierta, el resultado es true (un valor distinto de cero), si es falsa será false (cero). C++ dispone de los siguientes operadores relacionales:

Operador	Relación	Ejemplo	Resultado
<	Menor	X = 5; Y = 3; if(x < y) x+1;	X vale 5 Y vale 3
>	Mayor	X = 5; Y = 3; if(x > y) x+1;	X vale 6 Y vale 3
<=	Menor o igual	X = 2; Y = 3; if(x <= y) x+1;	X vale 3 Y vale 3
>=	Mayor o igual	X = 5; Y = 3; if(x >= y) x+1;	X vale 6 Y vale 3
==	Igual	X = 5; Y = 5; if(x == y) x+1;	X vale 6 Y vale 5
!=	Diferente	X = 5; Y = 3; if(x != y) y+1;	X vale 5 Y vale 4

Imagen: Operadores de igualdad y relacionales.

Fuente: <http://profesores.fi-b.unam.mx/carlos/lcpi/p09/OPERADORES%20EN%20C++.pdf>

Operador Igual y desigualdad

Los operadores de igualdad, igual a (==) tienen prioridad inferior que los operadores relacionales, pero se comportan de igual forma. El tipo de resultado de estos operadores es bool. El operador igual a (==) devuelve true si ambos operandos tienen

el mismo valor; de lo contrario, devuelve false.



Para determinar si una variable A tiene distinto contenido que una variable B debemos usar $A \neq B$. En C no se admite la sintaxis $A <> B$ que usamos en pseudocódigo.

```
C++

// expre_Equality_Operators.cpp
// compile with: /EHsc
#include <iostream>

using namespace std;

int main() {
    cout << boolalpha
         << "The true expression 3 != 2 yields: "
         << (3 != 2) << endl
         << "The false expression 20 == 10 yields: "
         << (20 == 10) << endl;
}
```

Imagen: Operadores igual.

Fuente: <https://learn.microsoft.com/es-es/cpp/cpp/equality-operators-equal-equal-and-exclptequal?view=msvc-170>

3.3.3 Igual, menor que, menor o igual que, mayor que, mayor o igual que, desigualdad.

Menor que

El signo menor que generalmente se coloca entre dos valores que se comparan y significa que el primer número es menor que el segundo número o viceversa. Su signo es "<"

Mayor que

El signo mayor que generalmente se coloca entre dos valores que se comparan y significa que el primer número es mayor que el segundo número o viceversa. Su signo es ">"

```
C++

// expre_Relational_Operators.cpp
// compile with: /EHsc
#include <iostream>

using namespace std;

int main() {
    cout << "The true expression 3 > 2 yields: "
         << (3 > 2) << endl
         << "The false expression 20 < 10 yields: "
         << (20 < 10) << endl;
}
```

Imagen: Menor que y Mayor que.

Fuente: <https://learn.microsoft.com/es-es/cpp/cpp/relational-operators-equal-and-equal?view=msvc-170>

Menor igual

Se usa para una aproximación del signo menor que o igual. Su signo es " \leq "

Mayor igual se utiliza para una aproximación del signo mayor que o igual Su signo es " \geq "

Suponiendo que la variable A tiene un valor $A = 5$ y la variable B un valor $B = 7$ podríamos plantear las siguientes expresiones:

Expresión	Resultado	Ejemplo código
$A == B$	0 (falso)	<code>printf ("A == B vale %d\n", (A == B));</code>
$A != B$	1 (verdadero)	<code>printf ("A != B vale %d\n", (A != B));</code>
$A > B$	0 (falso)	<code>printf ("A > B vale %d\n", (A > B));</code>
$A \geq B$	0 (falso)	<code>printf ("A \geq B vale %d\n", (A \geq B));</code>
$A < B$	1 (verdadero)	<code>printf ("A < B vale %d\n", (A < B));</code>
$A \leq B$	1 (verdadero)	<code>printf ("A \leq B vale %d\n", (A \leq B));</code>

Imagen: Operadores de igualdad y relacionales

Fuente:https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=908:operadores-logicos-y-relacionales-en-lenguaje-c-igual-que-distinto-and-oa-or-not-ejemploscu00519f&catid=82&Itemid=210



3.4 Estructuras de decisión

Por: Oscar Alexander Echeverría

Se le llama estructura de decisión a la función de realizar una acción en nuestro código desde el punto de vista lógico de una decisión.

La estructura de decisión la podemos dividir en: Simples, anidadas y dobles.

Simple: Es la acción dentro de nuestro código basándose en el resultado de las decisiones las cuales ejecutamos con una sentencia por condiciones utilizando (cierto o falso) por ejemplo.

Estructura de Decisión Simple

Diagrama de flujo.



Pseudocódigo en español.

```

Si (<condición>) entonces
    S1
    ...
    Sn
fin_si
    
```

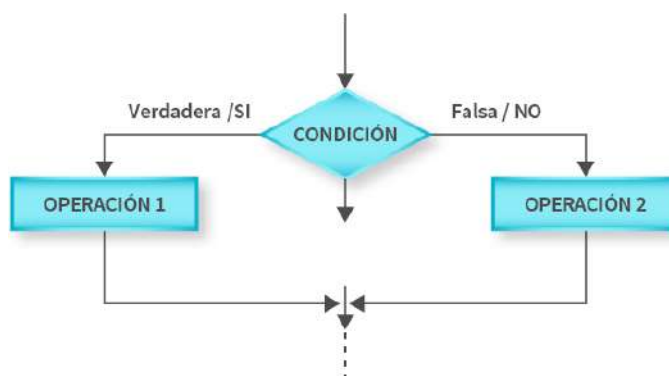
Código en C

```

if (<condición>)
{
    S1
    ...
    Sn
}
    
```

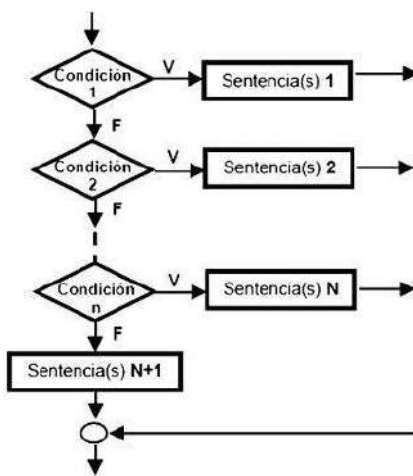
Fuente: https://lh3.googleusercontent.com/proxy/HsxpOEeGrmQiMD3PzRpdMVcW8Pmh7hNM8tshZtAaFYIVB6aB5yxHEGbjhQ3TAXyDnlwIM6OF_4z6hqab0EgYrGDmb7KHjJpN288bEcEnkVmSpOj5i7_X8=w1200-h630-p-k-no-nu

Dobles: Es la acción de escoger entre opciones el cumplimiento o no de nuestra condición.



Fuente: <https://www.uacj.mx/CGTI/CDTE/JPM/Documents/IIT/selectivas/images/iit-es-09.png>

Anidadas: Es la acción de ingresar una de nuestras condiciones dentro de otra condición para poder tener multiples opciones a ejecutar.



Fuente:http://img3.wikia.nocookie.net/_cb20130509020621/fundamentos/es/images/4/43/Estructuras_de_decisi%C3%B3n_anidadas.jpg

3.4.1 If simple

El if es una sintaxis la cual su funcionamiento es el de realizar la opción de una condición dentro de nuestro programa el cual si es (verdadero) ejecutara la opción asignada, pero si es (falso) realizara otra opción.

Ejemplos:

```

1)
  if (condicion) instruccion;

2)
  if (condicion)
{
  instruccion1
;
  instruccion2
;
  instruccionX
;
}

3)
  if (a > 10)
{
  instruccion1
; if (b < 10)
}
  
```

3.4.2 If...else

El if es una sintaxis la cual su funcionamiento es el de realizar la opción de una condición dentro de nuestro programa el cual si es (verdadero) ejecutara la opción asignada, pero si es (falso) realizara otra opción y el else es la manera en la cual podemos ejecutar un bloque para luego desencadenar una acción evaluando si es (verdadero o falso).

Ejemplos:

```
1)
if (condicion)
{
    instruccion1
    ;
    instruccion2
    ;
}
else
{
    instruccion3
    ;
    instruccion4
    ;
}

2)
if
(condicion1
)
instruccion1
;
```

3.4.3 Switch

Esta sintaxis nos ayuda a ordenar parte de nuestro código ramificando y simplificando nuestro código, esto también nos ayuda a estructurar múltiples condiciones nos sirve para crear un menú, por ejemplo.

Ejemplos:

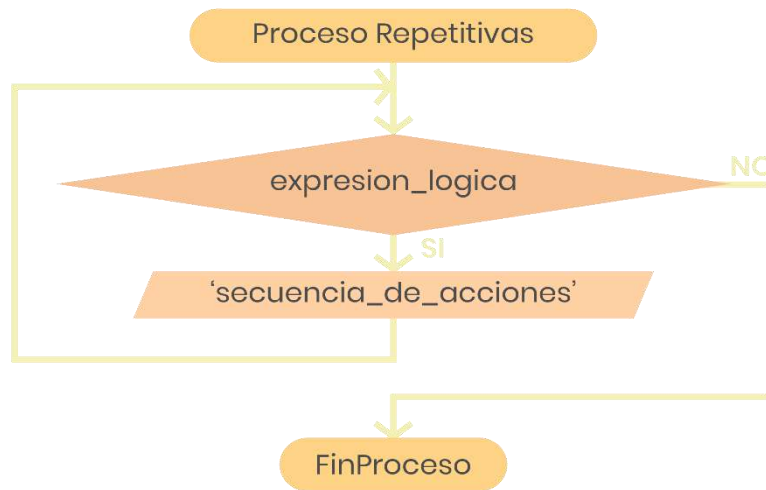
```
1)
switch (variable)
{
    case
    valor1:
    instruccion1
    ; break;
    case
    valor2:
    instruccion2
    ; break;
    [default:]
    instruccion_default
    ;
}
```

```
2)
switch(mes)
{
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        print("Hay 31 días en este
        mes"); break;
    case 2:
        print("Hay 28 días en este
        mes"); break;
    case 4:
    case 6:
    case 9:
    case 11:
        print("Hay 30 días en este
        mes"); break;
    default:
        print("Mes
```

3.5 Estructura de Repetición

Por: Aroldo Escobar

Una estructura de repetición (también conocida como bucle) es una estructura de control que permite la ejecución repetida de una o más instrucciones.



Fuente de la

<https://lumen.uv.mx/recursoseducativos/EstructuraMientras/contenido.html>

REPITA PARA

Se utiliza para repetir una sentencia o grupo de sentencias un número fijo de veces.
REPITA PARA variable_de_control:= valor_inicial HASTA valor_final

```
Sentencia 1;  
Sentencia 2;  
FIN RP
```

• REPETIR HASTA

Se utiliza para repetir una sentencia o grupo de sentencias hasta que una condición especificada sea verdadera.
Repetir

```
Sentencia 1;  
Sentencia 2;  
.....  
Sentencia n;  
hasta Condición;
```

• REPITA MIENTRAS

Se utiliza para repetir una sentencia o grupo de sentencias mientras una condición especificada sea verdadera.
Repita Mientras Condición se cumpla haga

```
Sentencia 1;  
Sentencia 2;  
.....  
Sentencia n;  
fin rm
```

Fuente de la imagen: <https://sites.google.com/site/algorithmicai/estructuras-de-repeticion>

Estas estructuras describen procesos que se deben realizar un cierto número finito de veces.

Por ejemplo, en el lenguaje C++, el bucle "for" se utiliza para iterar sobre un conjunto de datos y realizar una acción determinada.

Información proporcionada por las siguientes páginas:

1. <https://lumen.uv.mx/recursoseducativos/EstructuraMientras/contenido.html>

2. <https://sites.google.com/site/algorithmicai/estructuras-de-repeticion>
3. <http://courseware.url.edu.gt/Facultades/Facultad%20de%20Ingenier%C3%ADa/Ingenier%C3%ADa%20en%20Inform%C3%A1tica%20y%20Sistemas/Segundo%20Ciclo%202011/Introducci%C3%B3n%20a%20la%2>



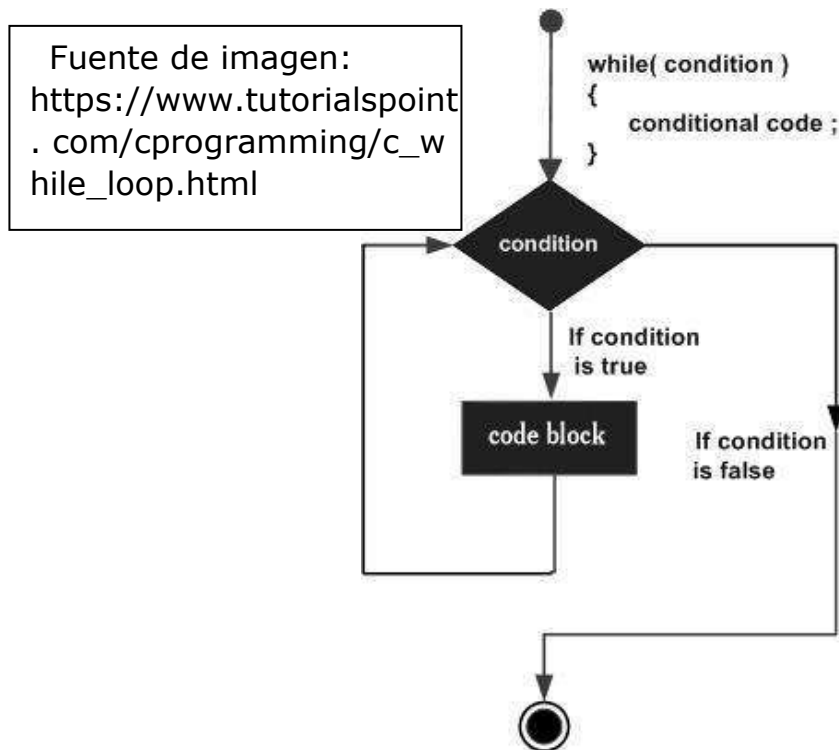
[0Programaci%C3%B3n/Objetos%20de%20aprendizaje/Unidad%203B/Unidad-3B/qu_es_una_estructura_repetitiva.html](https://www.tutorialspoint.com/cprogramming/c_while_loop.html)

4. <https://galanistaprogramando.wordpress.com/estructuras-repetitivas/>

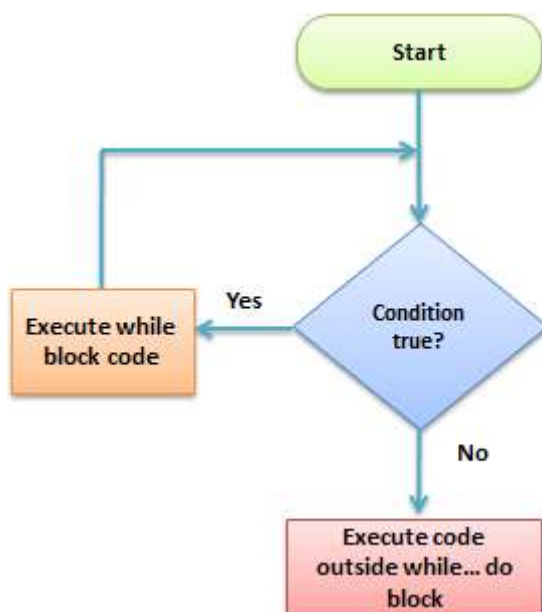
5. <https://aprendiendoarduino.wordpress.com/category/estructuras-de-repeticion/>

3.5.1 While

Un bucle while es una estructura de control que se utiliza para repetir una sección de código un número desconocido de veces hasta



Por ejemplo, si queremos saber cuántas veces se necesita iterar sobre un conjunto de datos para realizar una acción determinada, podemos usar el bucle "while" en lenguaje C.



Fuente de imagen:
<https://www.guru99.com/c-loop-statement.html>

En Python, el bucle "while" se utiliza para ejecutar un bloque de código hasta que se cumpla una condición dada. En C++, el bucle "while" itera a través del código especificado en su cuerpo (llamado declaración while)

mientras se cumpla una condición predeterminada.

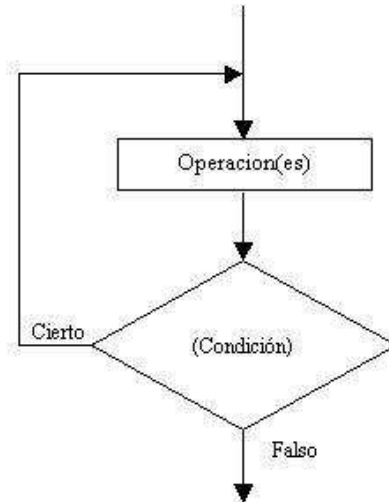
Información proporcionada por las siguientes páginas:

1. https://users.cs.utah.edu/~germain/PPS/Topics/while_loops.html

2. https://www.tutorialspoint.com/cprogramming/c_while_loop.htm
3. <https://www.guru99.com/c-loop-statement.html>
4. <https://www.programiz.com/python-programming/while-loop>
5. <https://study.com/academy/lesson/while-loop-definition-example-results.html>

3.5.2 Do... While

La estructura de control do while es un bucle que ejecuta una sentencia especificada hasta que la condición de comprobación se evalúa como verdadera.



Fuente de la imagen:
<https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=85>

```

JavaScript Demo: Statement - Do...While
1 let result = '';
2 let i = 0;
3
4 do {
5   i = i + 1;
6   result = result + i;
7 } while (i < 5);
8
9 console.log(result);
10 // Expected output: "12345"
11

```

Fuente de la imagen:
<https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statement>

Por ejemplo, se puede usar para sumar los 7 primeros números enteros o para ejecutar de manera repetitiva un bloque de instrucciones sin conocer el número exacto de veces que se va a iterar. En C, la instrucción do-while crea un bucle que ejecuta una sentencia especificada mientras se cumpla una condición dada.

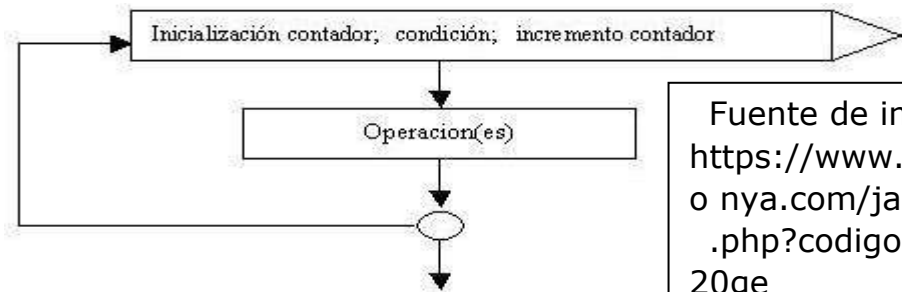
Información proporcionada por las siguientes páginas:

1. <https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=85>
2. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/do...while>
3. <https://ccia.ugr.es/~jfv/ed1/c/cdrom/cap4/cap45.htm>
4. <https://learn.microsoft.com/es-es/cpp/c-language/do-while-statement-c?view=msvc-170>

3.5.3 For



La estructura de control for es un bucle que se usa para repetir un bloque de instrucciones un número determinado de veces.



Fuente de imagen:
<https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=84#:~:text=En%20general%2C%20la%20estructura%20de%20repetici%20n%20de%20veces%20se%20usa%20para%20repetir%20un%20bloque%20de%20instrucciones%20un%20n%C3%BAmero%20determinado%20de%20veces.>

Por ejemplo, se puede usar para cargar 10 números, ingresar 5 notas de alumnos o realizar la suma de los 8 primeros números enteros.

Información proporcionada por las siguientes páginas:

1. <https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=84#:~:text=En%20general%2C%20la%20estructura%20de%20repetici%20n%20de%20veces%20se%20usa%20para%20repetir%20un%20bloque%20de%20instrucciones%20un%20n%C3%BAmero%20determinado%20de%20veces.>
2. <https://aprendiendoarduino.wordpress.com/category/estructuras-de-repeticion/>
3. <https://ccia.ugr.es/~jfv/ed1/c/cdrom/cap4/cap43.htm>
4. http://odin.fi-b.unam.mx/salac/practicafp/fp_p9.pdf

3.5.4 Operadores Lógicos

Los operadores lógicos AND, OR y NOT se usan para combinar relaciones lógicas [1][2]. El orden de prioridad es "not" antes que "and" y "and" antes que "or".

```
- and, significa "y",  
- or, significa "y/o",  
- not, significa "no", invierte el resultado  
- (), paréntesis
```

Fuente de imagen:
<https://sites.google.com/site/tumonitororg/postgresql/operadores-logicos-and-or-not>

Los operadores incluyen la negación lógica unaria (!), AND lógico binario (&), OR (|) y OR exclusivo (^) y el AND lógico condicional.

Información proporcionada por las siguientes páginas:

```
bool SecondOperand()  
{  
    Console.WriteLine("Second operand is evaluated.");  
    return true;  
}  
  
bool a = false & SecondOperand();  
Console.WriteLine(a);  
// Output:  
// Second operand is evaluated.  
// False  
  
bool b = true & SecondOperand();  
Console.WriteLine(b);  
// Output:  
// Second operand is evaluated.  
// True
```

Fuente de imagen:
<https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/operators/boolean>

Información proporcionada por las siguientes páginas:

1. [http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/operadores1.htm#:~:text=Los%20operadores%20l%C3%B3gicos,-Los%20operadores%20l%C3%B3gicos&text=%26%26%20AND%20\(e%20resultado%20es,la%20condici%C3%B3n%20de%20la%20expresi%C3%B3n\)](http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/operadores1.htm#:~:text=Los%20operadores%20l%C3%B3gicos,-Los%20operadores%20l%C3%B3gicos&text=%26%26%20AND%20(e%20resultado%20es,la%20condici%C3%B3n%20de%20la%20expresi%C3%B3n))
2. <https://www.ibm.com/docs/es/spss-statistics/saas?topic=expressions-logical-operators>
3. <https://sites.google.com/site/tumentororg/postgresql/operadores-logicos-and-or-not>
4. <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/operators/boolean-logical-operators>
5. https://fmhelp.filemaker.com/help/18/fmp/es/index.html#page/FMP_Help/logical-operators.html

3.5.5 And, Or, Not

Los operadores lógicos AND, OR y NOT se usan para combinar relaciones lógicas.

SÍMBOLO	SIGNIFICADO	EJEMPLO
=	Igual que	A = B
>	Mayor que	A > B
<	Menor que	A < B
>=	Mayor o igual que	A >= B
<=	Menor o igual que	A <= B
<>	Distinto que	A <> B

Fuente de imagen:

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=272:operadores-logicos-y-de-comparacion-en-programacion-not-and-or-ejemplos-cu00132a&catid=28&Itemid=59

El operador AND (&) devuelve verdadero si ambas expresiones son verdaderas, el operador OR (|) devuelve verdadero si alguna expresión es verdadera, y el operador NOT (!) invierte la condición de la expresión.

Operador	Nombre	Ejemplo	Significado
<	menor que	$a < b$	a es menor que b
>	mayor que	$a > b$	a es mayor que b
==	igual a	$a == b$	a es igual a b
~=	no igual a	$a \sim b$	a no es igual a b
<=	menor que o igual a	$a \leq 5$	a es menor que o igual a b
>=	mayor que o igual a	$a \geq b$	a es menor que o igual a b

Información proporcionada por las siguientes páginas:

1. https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=272:operadores-logicos-y-de-comparacion-en-programacion-not-and-or-ejemplos-cu00132a&catid=28&Itemid=59
2. <https://es.javascript.info/logical-operators>
3. https://www.ciberaula.com/cursos/java/operadores_logicos.php
4. <http://www.sc.ehu.es/sbweb/fisica3/basico/condicional/condicional.html>
5. <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/operators/boolean-logical-operators>

3.5.6 Vectores y Matrices

Un vector es una secuencia de objetos del mismo tipo almacenados consecutivamente en memoria. Una matriz es una estructura de objetos con filas y columnas. Los vectores y matrices se usan para realizar operaciones básicas como la concatenación, lectura de componentes, definición a partir de rangos y linspace().

```
octave:11> v = [ 1 2 3 4 5]
v =
   1   2   3   4   5

octave:12> w = [6; 7; 8]
w =
   6
   7
   8

octave:13> |
```

Información proporcionada por las siguientes páginas:

1. http://informatica.uv.es/ieguia/II/prac6_CAS.pdf
2. <https://www.monografias.com/trabajos-pdf5/vectores-y-matrices/vectores-y-matrices>
3. https://eva.fing.edu.uy/pluginfile.php/307994/mod_resource/content/1/Vectores%20y%20matrices.pdf
4. [https://espanol.libretexts.org/Ingenieria/Ingenier%C3%ADa_Mec%C3%A1nica/Matem%C3%A1ticas%2C_Numerics_y_Programaci%C3%B3n_\(para_Ingenieros_Mec%C3%A1nicos\)/03%3A_Unidad_III_-_%C3%81lgebra_lineal_1_-](https://espanol.libretexts.org/Ingenieria/Ingenier%C3%ADa_Mec%C3%A1nica/Matem%C3%A1ticas%2C_Numerics_y_Programaci%C3%B3n_(para_Ingenieros_Mec%C3%A1nicos)/03%3A_Unidad_III_-_%C3%81lgebra_lineal_1_-)

[_Matrices%2C m%C3%ADnimos cuadrados y regresi%C3%B3n/16%3A Matrices y Vectores -
_Definiciones y Operaciones/16.01%3A Operaciones B%C3%A1sicas
_de Vectores y Matrices](#)



5. <https://octaveintro.readthedocs.io/en/latest/vectores.html>



TECNOLOGÍA

Bimestre IV

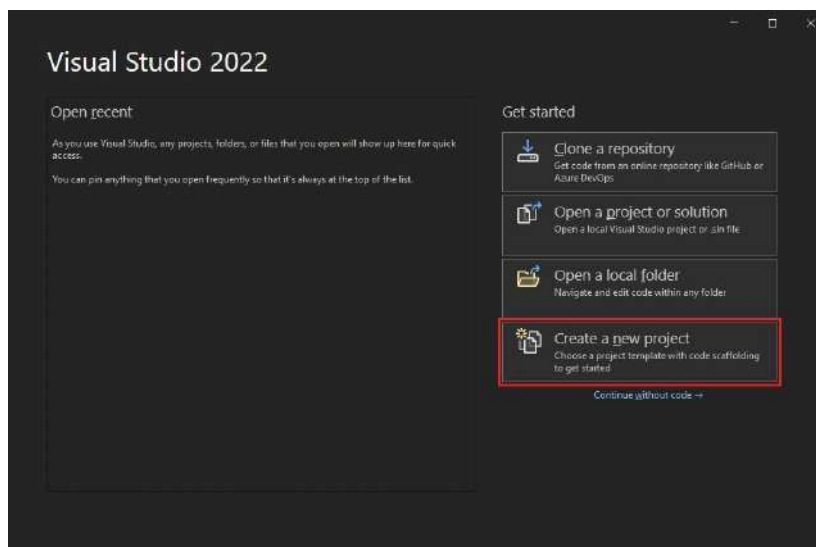


4.1 Windows Form App

Por: Pavel Palma

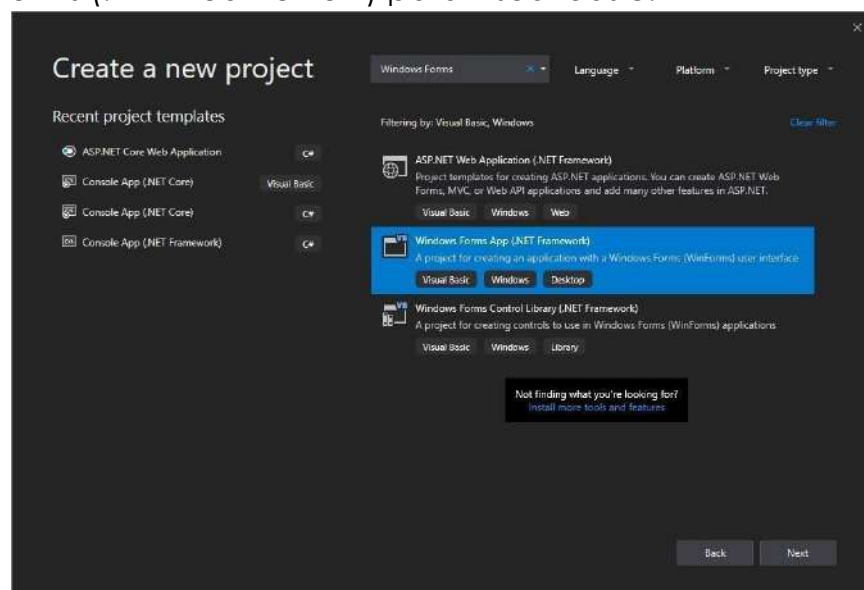
4.1.1 Creacion del Proyecto

- 1) Abrimos nuestro visual studio
- 2) En la ventana de inicio, elegimos la opcion de Crear un nuevo proyecto



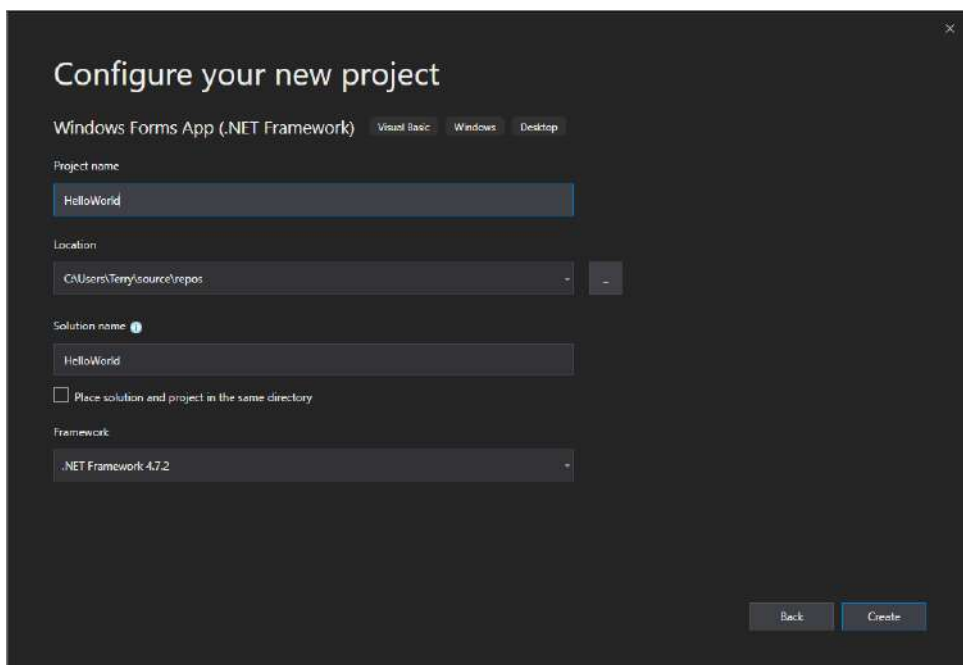
Fuente: [Tutorial: Crear una aplicación de Windows Forms con Visual Basic - Visual Studio \(Windows\) | Microsoft Learn](#)

- 3) En la ventana de crear un nuevo proyecto, seleccionamos la plantilla aplicación de Windows Forms (.NET Framework) para visual basic.



Fuente: [Tutorial: Crear una aplicación de Windows Forms con Visual Basic - Visual Studio \(Windows\) | Microsoft Learn](#)

- 4) En la ventana de Configure your new Project, escriba el nombre que desee como el nombre del proyecto. y a continuación, seleccione crear.



Fuente: [Tutorial: Crear una aplicación de Windows Forms con Visual Basic - Visual Studio \(Windows\) | Microsoft Learn](#)

5) Y con esto se crearía nuestro proyecto listo para usarlo.

4.2 .NET framework

Un framework consiste en una serie de estructuras y tecnologías que básicamente nos facilita la programación, cuando hablamos del .NET Framework estamos hablando de este conjunto de estructuras que proporcionan para una programación mas sencilla orientada a las redes e internet. Para programar en .NET existen mas de 20 lenguajes de programación, pero C# y Visual Basic son los mas populares.

4.2.1 Para que sirve .NET framework

Este es un componente software que da soluciones de códigos que son utilizadas por los programas y que se gestionan programas escritos para este Framework.

.NET Framework proporciona un entorno de desarrollo que utiliza un software sencillo, aumentando la seguridad de los programas y reduciendo las vulnerabilidades. Esta plataforma entra dentro del campo de servicios Web al XML.

4.2.2 Versiones de .NET Framework

Cada una de las versiones de .NET Framework contiene algunas características de la versión anterior y incorpora nuevas. Dentro del .NET Framework se incorpora en CLR (Common Language Runtime) que incluye componentes básicos y algunos adicionales.

La primera versión de .NET Framework fue la 1.0 que contenía también la primera versión de CLR y la primera versión de las bibliotecas de clases. Luego empiezan a salir las nuevas actualizaciones y versiones hasta llegar a la actual que sería la .NET Framework 4.8.1.





Fuente: [Microsoft anuncia .NET Framework 4.8 – Jorge Serrano \(geeks.ms\)](#)

4.2.3 Como descargar .NET framework

Para descargarlo existen 2 manera, que son por la pagina oficial de Microsoft, eligiendo la versión que nosotros deseamos o por medio de Windows 10, dentro de las opciones de configuración, en actualizaciones y seguridad.

**Download .NET Framework
4.7.2**

Fuente: [Download .NET Framework 4.7.2 | Free official downloads \(microsoft.com\)](#)

4.1 Visual Studio .NET POO

Por: Erwin Patzàn

Visual Basic proporciona compatibilidad total con la programación orientada a objetos, incluidos el encapsulado, la herencia y el polimorfismo.

La encapsulación significa que un grupo de propiedades, métodos y otros elementos relacionados se trata como una sola entidad u objeto. La herencia describe la capacidad de crear nuevas clases basadas en una clase existente.

El polimorfismo significa que puede tener varias clases que son intercambiables, incluso si cada clase implementa las mismas propiedades o métodos de manera diferente.

1.1.1 Clases y Objetos

Los términos "clase" y "objeto" a veces se usan indistintamente, pero en realidad, las clases describen un tipo de objeto y los objetos son instancias de clases que se pueden usar. Por lo tanto, el acto de crear un objeto se denomina instancia. Usando la analogía del plano, el salón de clases es un plano y la base es un edificio construido sobre ese plano.

Ejemplo:

```
VB Copiar
Class SampleClass
End Class
```

Imagen: Ejemplo de una clase

Fuente: <https://learn.microsoft.com/es-es/dotnet/visual-basic/programming-guide/concepts/object-oriented-programming>

4.1.3 Controles.Net

Definición del controlador:

Los controles son objetos y los objetos son combinaciones de datos y código que se tratan como una sola entidad.

Ejemplo de controles:

Todos los objetos de la barra de herramientas son controles, como: formulario, etiqueta, botón, cuadro de texto, imagen, temporizador, etc. Propiedades, métodos y eventos:

Todos los controles tienen las siguientes características:

Propiedades: Define sus características como color, tamaño, posición, nombre, etc.
Métodos: Son conjuntos de procedimientos que permiten al controlador realizar una

acción. Los métodos de ejemplo son el método Show y el método Hide, que hacen que el control sea visible e invisible según el método al que se crea o llama.

Eventos: Estas son acciones que el control reconoce y responde. Ejemplo de evento: evento de clic de botón; hacer que devuelva una respuesta programada, como cerrar un formulario o mostrar un mensaje. El evento Mouse Enter en la etiqueta puede hacer que se muestre un mensaje cuando el puntero del mouse está sobre el control. Hay más de 80 eventos disponibles en el formulario en el que puede reaccionar.

Los objetos son los elementos que colocaremos en el formulario y este también es un objeto. En la imagen de abajo puedes ver muchos ejemplos de controles de la barra de herramientas que usaremos para desarrollar nuestros proyectos.

Ejemplo de controles;

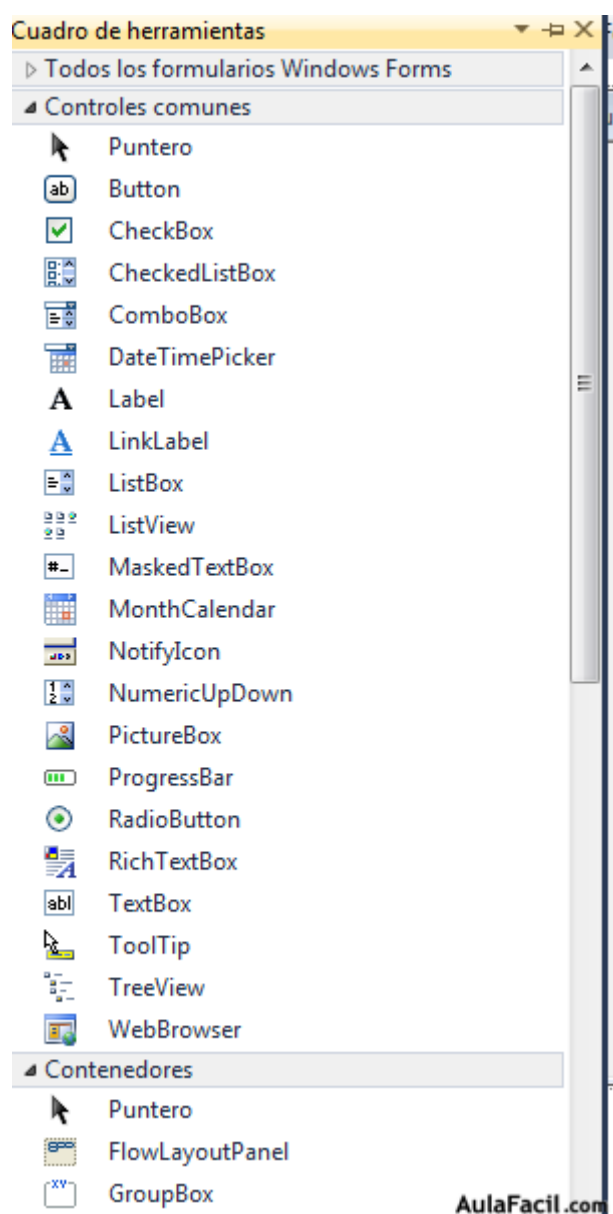


Imagen: controles

Fuente: <https://www.aulafacil.com/cursos/programacion/visual-basic-paso-a-paso/cuadro-de-herramientas-y- controles-142107#:~:text=Los%20controles%20son%20objetos%20y,%2C%20PictureBox%2C%20Timer%2C%20etc.>

4.1.4 Codificación avanzada

La codificación de archivos, también conocida como codificación de caracteres,

especifica cómo se representan los caracteres durante el procesamiento de texto. Aunque generalmente se prefiere Unicode, se puede preferir una codificación sobre otra en función de los caracteres del idioma que puede o no manejar.

Unicode es la codificación preferida cuando se trabaja con archivos. Unicode es un

estándar mundial de codificación de caracteres que utiliza valores de código de 16 bits para representar todos los caracteres utilizados en las computadoras modernas, incluidos los símbolos técnicos y los caracteres especiales utilizados en las publicaciones.

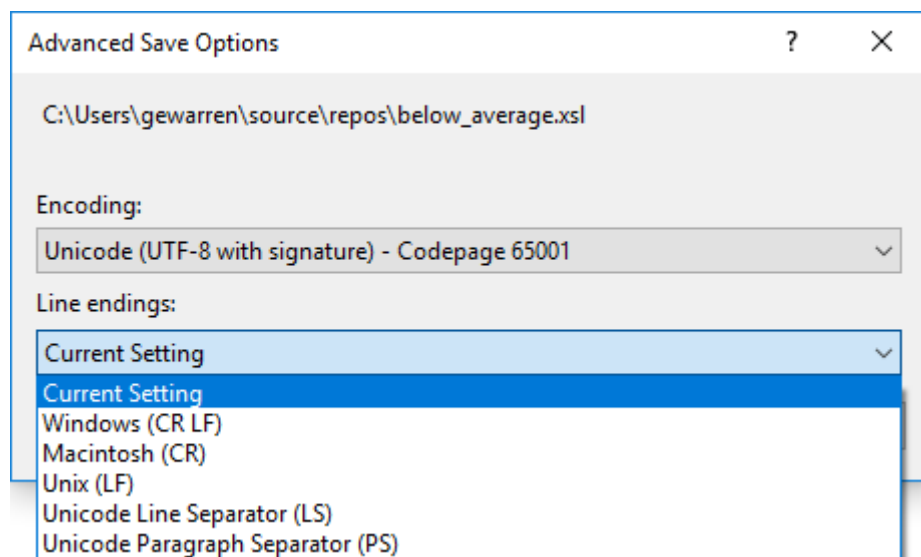


Imagen: Codificación

Fuente: <https://learn.microsoft.com/es-es/visualstudio/ide>

4.1 Visual Studio .NET POO

Por: Wilmer Estuardo Patzan Vail

Un entorno de desarrollo integrado (IDE) es un programa con numerosas características que respalde muchos aspectos del desarrollo de software. El IDE de visual studio es un panel de inicio creativo que se puede usar para editar, depurar y compilar código y después publicas una aplicación.

4.1.5 Formulario .NET

Net es una plataforma de código abierto para crear aplicaciones de escritorio web y móviles que se pueden ejecutar de forma nativa en cualquier sistema operativo. El sistema .NET incluye herramientas bibliotecas y lenguajes que admiten el desarrollo de software moderno, escalable y de alto rendimiento, ya que en una comunidad de desarrolladores activa mantiene y apoya la plataforma .NET.

4.1.5.1 Formulario de Registro de Sesión

EL primerísimo paso de un usuario aterriza en una web o app que se identifique, son los que necesitas para obtener datos personales básicos para identificar a todos esos usuarios anónimos. La mayoría de sitios web no le dan suficiente importancia a este primer contacto entre el usuario y la plataforma y muestran formularios de registro básicos.

4.1.6 Medidas de Seguridad

La mayoría de los errores de codificación que tienen como resultado la aparición de puntos vulnerables de seguridad se producen por que los programadores no hacen la susposiciones validas cuando trabajan con datos que proporciona el usuario por que no acaban de entender la plataforma para la que están desarrollando. Las directrices de codificación segura describen la diferente formas en que se pueden diseñar código.NET para que funcione con el sistema de seguridad.



<https://www.incanatoit.com/2014/11/formularios-controles-programacion-visual-basic-net.html>

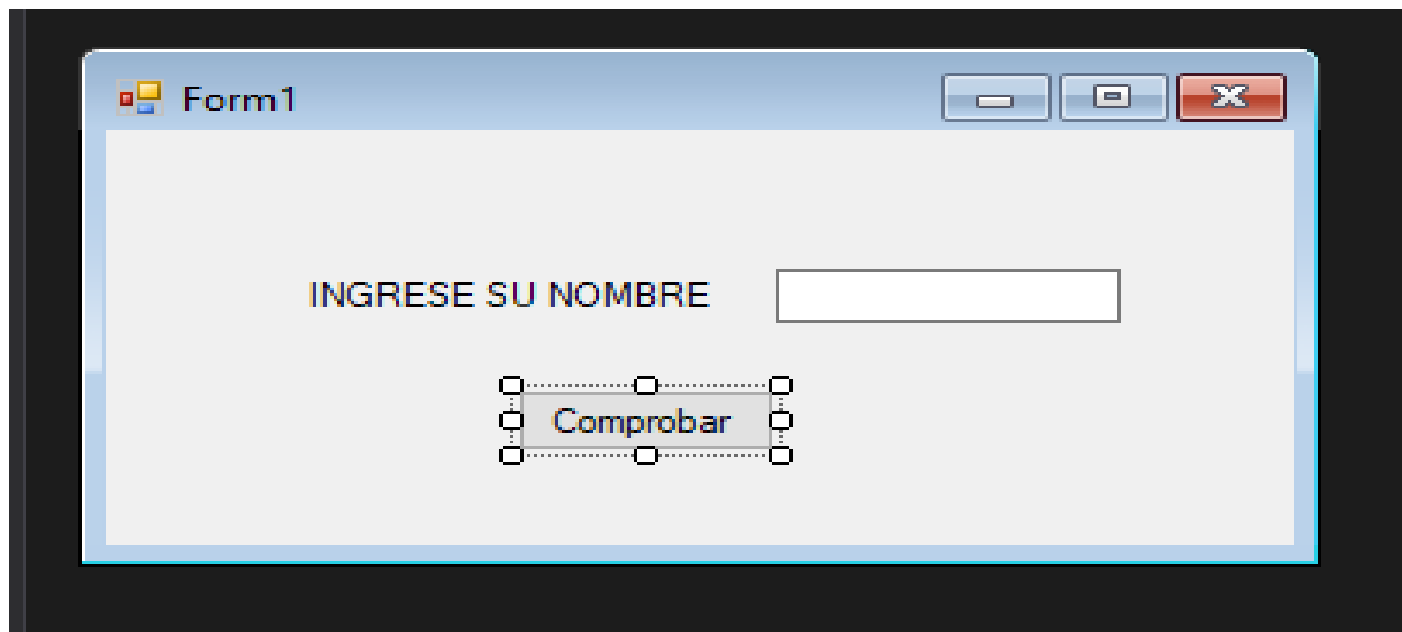


<https://karmany.net/proteccion-software/29-vb6-desde-programacion/16-protege-tu-programa-vb6-i>

4.1.7 ErrorProvider

Por: Leonel Perez

ErrorProvider Sirve para muestra una notificación de error al usuario cuando estemos trabajando en nuestra interfaz gráfica. Si el usuario ha ingresado un dato erróneo el ErrorProvider nos será de utilidad, podemos usar este control para notificar la existencia de un error.



Form utilizado para demostrar ErrorProvider

Fuente: Capturado con Snipping

Creamos un Form donde podamos realizar nuestro ejemplo práctico, creamos un Label, TextBox y un Button, entramos al código de nuestro TextBox. Y Escribimos el siguiente código.

```
Private Sub TxtNombre_TextChanged(sender As Object, e As EventArgs) Handles
TxtNombre.TextChanged
```

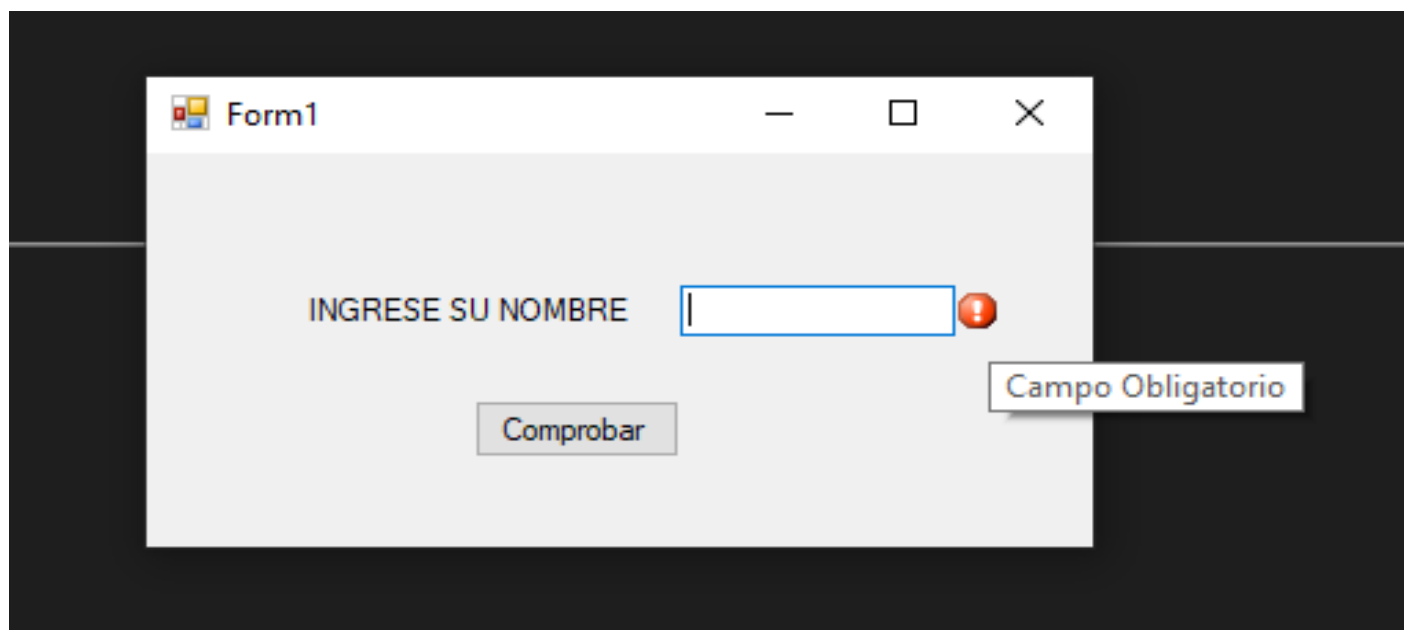
```
    If TxtNombre.Text.Length <= 0 Then
        ErrorProvider1.SetError(TxtNombre, "Campo Obligatorio")
    Else
        ErrorProvider1.SetError(TxtNombre, "
    ") End If
```

```
End Sub
```

TxtNombre.Text.Length ← Este Código devuelve la longitud de la cadena de texto contenida dentro del TextBox.

ErrorProvider1.**SetError(TxtNombre, "Campo Obligatorio")** ← Toma dos argumentos: el primer argumento es el control al que se asocia el error, y el segundo argumento es el mensaje de error que se mostrará al usuario.

ErrorProvider1.SetError(TxtNombre, "") ← Este Código se ejecuta si la cadena contiene algo, esto hará que no muestre ningún mensaje de error.



Resultado del programa en
ejecución Fuente: Capturado con
Snipping

4.1.8 AntiErrores

En Visual Basic existen diversas formas de manejar errores entre las más comunes pueden ser como los Try Catch Finally. En resumen, existen diversas formas de prevenir y manejar errores en Visual Basic, y es importante tener en cuenta buenas prácticas de programación para garantizar un código de alta calidad y sin errores.

If char .IsNumber(e.KeyChar) then

e.Handled = false

End If

Con este código podemos habilitar o deshabilitar números, letras, caracteres, etc. Para poder deshabilitar espacios el comando a usar sería.

If char .IsSeparator(e.KeyChar) then

e.Handled = false

End If

Ureferencias

```
Private Sub txtedad_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtedad.KeyPress
    If Char.IsNumber(e.KeyChar) Then
        e.Handled = False
    ElseIf Char.IsControl(e.KeyChar) Then
        e.Handled = False
    ElseIf Char.IsSeparator(e.KeyChar) Then
        e.Handled = False
    Else
        e.Handled = True
        MsgBox("Solo se permite números")
    End If
End Sub
```

Uso para deshabilitar Numeros
Fuente: Capturado con
Snipping

1.1 Try Catch

Por: Krystel Popa

Se utiliza para manejar fragmentos de código que son propensos a fallar, como puede ser: recibir un valor nulo, convertir un tipo de dato a otro o en otro caso, la lectura de un archivo.

1.1.1 Función

La función que realiza es que al leer un archivo que está almacenado en la memoria del dispositivo, se genera un flujo de bits, desde dicha memoria a la memoria RAM.

1.1.2 Try

El "try" contiene el código protegido que puede producir la excepción. El bloque hasta que se produce una excepción o hasta que se completa correctamente.

1.1.3 Catch

El "catch" se puede utilizar para detectar cualquier tipo de excepción, no se recomienda este uso. Este solo debe de convertir las excepciones que sabe cómo recuperar.

```
try {  
  
    // Convierte un valor de tipo String a int  
    int numero = Integer.parseInt("a456");  
  
} catch(Exception e) {  
  
    System.out.println("ERROR: el valor de tipo String contiene caracteres no válidos");  
  
}
```

Fuente: <https://es.stackoverflow.com/questions/96278/para-que-sirve-el-try-y-catch-en-java>

1.1.4 Beneficios de su uso

Este nos permite crear programas mucho más robustos y tolerantes a fallas que ocurren en escasas situaciones, pero en caso que se presenten disponemos de un algoritmo alternativo para reaccionar a dicha situación evitando que el programa finalice su ejecución.

```
static void Main()
{
    int n;
    try
    {
        // Do not initialize this variable here.
        n = 123;
    }
    catch
    {
    }
    // Error: Use of unassigned local variable 'n'.
    Console.Write(n);
}
```

Fuente: <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/try-catch>



ANÁLISIS BIMESTRE 1



1.1.1PSeInt

Por: Thania García

Es una herramienta educativa para ayudar a los estudiantes en los primeros pasos dentro de la programación. Este software utiliza un pseudolenguaje complementado con diagramas de flujos, lo que hace que el estudiante centre su atención en los conceptos principales del algoritmo computacional con numerosas ayudas y recursos didácticos.

1.1.2Lógica de programación

Se podrían decir muchas cosas acerca de la lógica de la programación, resolución de problemas, de la programación en sí misma, etc. Voy a tratar de saltar toda la introducción pesada e ir directamente a la acción, ya que cuando yo aprendí no necesité toda esa teoría de entrada, sino que lo que me facilitó el camino, aparte del hecho de que me gustaba mucho la programación, fue practicar y intentar programar cualquier tipo de cosa que se me ocurría.

Creo que lo único que necesitan saber para empezar son cosas sencillas:

- Los programas ejecutan instrucciones PASO A PASO, línea tras línea, una debajo de la otra.
- Para almacenar información necesitamos crear pequeños espacios de memoria.
- Podemos hacer que tome un camino u otro dependiendo de CONDICIONES.
- Pueden repetir una o varias líneas de código MIENTRAS una condición sea cumplida.

El resto de cosas se pueden aprender literalmente a la pasada, a medida que se van metiendo en el tema. Bien... Vamos a empezar.



1.1.3 Conociendo el compilador

El compilador es la herramienta que vamos a usar para programar nuestras aplicaciones. Se encarga de transformar esto:

Proceso ejemplo1

Definir num1 **Como Entero**;

Escribir "Ingrese un número: ";

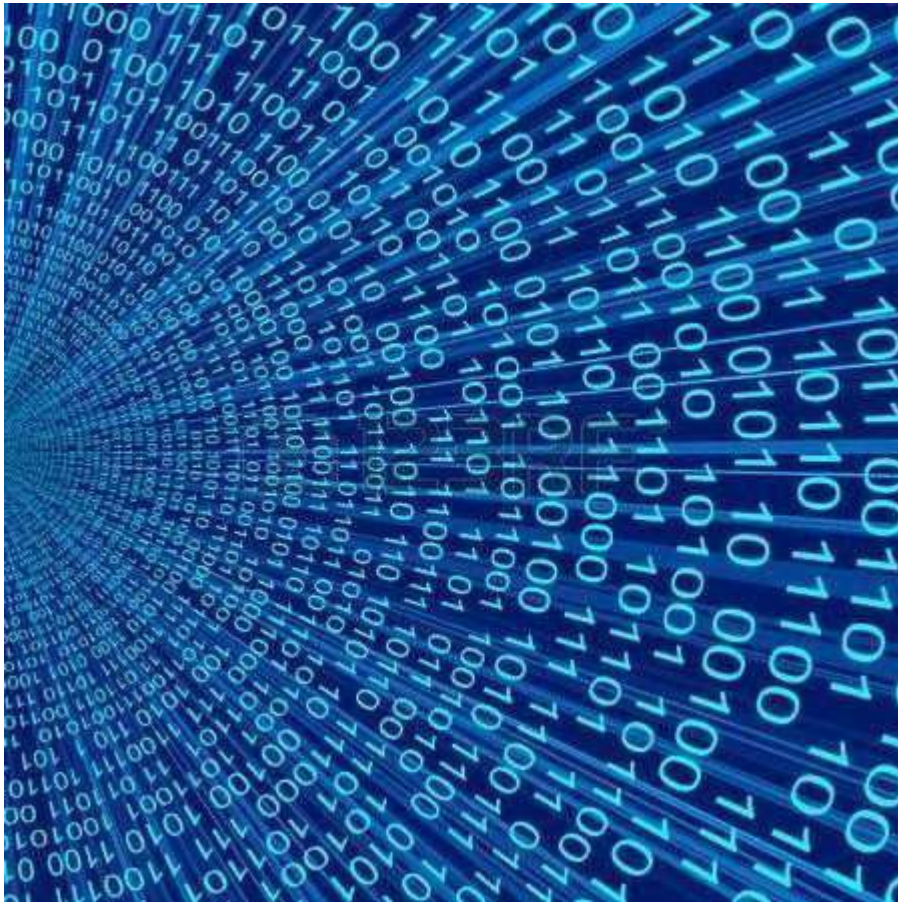
Leer num1;

Escribir "El doble de ese número es: ", num1 * 2;

FinProceso

Imágenes: PSEINT

En esto: <https://www.ighniz.com/2017/04/20/pseint-logica-de-la-programacion/>



Imágenes: PSEINT

<https://www.ighniz.com/2017/04/20/pseint-logica-de-la-programacion/>

O sea que, de manera más corta, se encarga de crear la parte ejecutable de nuestro programa. Esto en la gran mayoría de los compiladores se puede hacer con una tecla, como, por ejemplo, en el caso de PSeInt: F9.



1.1.4 Declaración de variables

Podríamos empezar con la pregunta ¿Qué es una variable?

Básicamente es un espacio en la memoria para guardar la información que necesitemos, ya sea un número, un texto, o un valor cualquiera sea.

```
Proceso miprograma
  Definir num1 Como Entero;
FinProceso
```

Imágenes: PSEINT

<https://www.ighniz.com/2017/04/20/pseint-logica-de-la-programacion/>

Tener en cuenta que las palabras “Proceso” y “FinProceso” ya vienen creadas con el programa al crear un archivo nuevo. Básicamente son los comandos que indican dónde inicia y dónde termina el programa respectivamente.

Con respecto a la línea 2, lo que se está haciendo con esto es crear un espacio en la memoria para guardar SOLAMENTE números enteros, y le estamos poniendo un nombre a ESE espacio, llamado “num1”. Siempre es recomendable que las variables que se crean tengan un nombre descriptivo acorde a lo que se va a guardar en ellas. Por ejemplo, si se va a guardar el puntaje del usuario, quizás se debería llamar “puntaje” o “score”.

1.1.5 Sentencias iniciales

Vamos a aprender algunas sentencias iniciales antes de programar nuestra primera aplicación. Todo lenguaje tiene comandos, sentencias, procesos, como quieran decirlo, que básicamente son instrucciones ya predefinidas que cumplen una tarea determinada.

Las primeras dos que vamos a ver se llaman “Escribir” y “Leer”. Ambas, como la gran mayoría de los comandos, reciben algún “parámetro”. Los parámetros son datos que uno le da a cada instrucción como para que se haga algo con él. Por ejemplo, si queremos escribir algo en la pantalla, usaremos el comando “Escribir” y le pasaremos por parámetro “qué cosa” vamos a escribir.

```
Proceso ejemplo
  Escribir “Hola mundo!”;
FinProceso
```

Imágenes: PSEINT

<https://www.ighniz.com/2017/04/20/pseint-logica-de-la-programacion/>



1.1.6 Poniendo Valores a Nuestras variables

Siempre que vayamos a hacer algún juego/programa vamos a necesitar ponerle un valor a nuestras variables, en lugar de dejar que el usuario lo haga (como vimos en el ejemplo del comando "Leer "). Para esos casos, vamos a usar algo llamado "OPERADOR". Hay muchos tipos de operadores, cada uno con una función específica. Vamos a empezar usando el operador "=" (igual), que sirve para guardar un dato en una variable. Veamos un ejemplo:

Proceso

Definir num1 Como Entero;

num1 = 5;

FinProceso

Imágenes: PSEINT

<https://www.ighniz.com/2017/04/20/pseint-logica-de-la-programacion/>

1.1.7 Variables y tipos de datos de los algoritmos con PSEINT

Una variable es un espacio de la memoria donde guardar información. La información que se guarda en la variable puede ser de diversos tipos y puede ir cambiando a lo largo del programa. A la variable hay que darle un nombre para identificarla, y ese nombre estará formado solo por letras, números y el guion bajo únicamente; no debe contener espacios ni operadores, ni palabras claves del lenguaje. Identificadores de variable correctos podrían ser: *nombre_usuario*, *fecha_2018*, *edad*, *A*, *X*, *suma*.

```
// Sintaxis para declarar variables
Definir (var1, var2, ..., varN) Como (tipo_de_dato);

// Ejemplos
Definir nombre Como Texto;
Definir edad Como Entero;
Definir respuesta Como Logico;
```

Imágenes: PSEINT

<https://terepebernal.com/blog/pseudocodigo/algoritmos-pseint-lo-mas-basico/>

Una vez declarada la variable se le puede **asignar un valor y también pedir por teclado ese valor (leer)**. También existe una palabra clave para **mostrar datos por pantalla**, esta es Escribir, si se utiliza Sin Saltar, no se hace un salto de línea después, sino se utiliza se hará un salto de línea.

```
// Asignación de variables
var1 <- valor;

// Lectura de variables
Leer var1;
Leer var2 Sin Saltar;

// Mostrar datos por pantalla
Escribir (cadena_texto);
Escribir var1;

// Ejemplos
nombre <- "Antonio";

Escribir "Introduce tu nombre: ";
Leer nombre;
Escribir "¿Edad?: " Sin Saltar;
Leer edad;
```

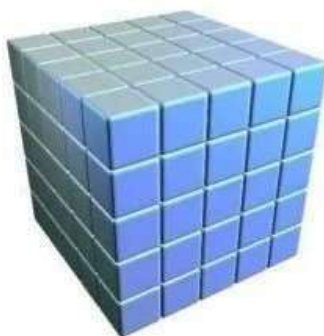
Imágenes: PSEINT

<https://terepebernal.com/blog/pseudocodigo/algoritmos-pseint-lo-mas-basico/>

[basico/](#)



Los tipos de datos anteriores son simples, además existen los **arreglos**, que son algo más complejos, son las matrices matemáticas, estructuras de datos homogéneos del mismo tipo que pueden tener varias dimensiones. También se suelen llamar *arrays*.



Imágenes: PSEINT

<https://terepebernal.com/blog/pseudocodigo/algoritmos-pseint-lo-mas-basico/>

Para declarar un arreglo se utiliza la palabra clave **Dimension**, puedes verlo en los dos ejercicios básicos que siguen:

```
// Sintaxis declaración de un arreglo
Dimension (nombre_arreglo) [(max1), (max2), ..., (maxN)];

// Arreglo para almacenar las notas de 8 alumnos para 5 asignaturas.
Dimension notas[8,5];
notas[1,1]<-8;
notas[1,2]<-4.5;
notas[1,3]<-9.2;
notas[1,4]<-7;
notas[1,5]<-3;

// Para mostrarlo por pantalla
Escribir "Las notas del alumno 1 son:";
Escribir "Asignatura 1: " Sin Saltar;
Escribir notas[1,1];
Escribir "Asignatura 2: " Sin Saltar;
Escribir notas[1,2];
Escribir "Asignatura 3: " Sin Saltar;
Escribir notas[1,3];
Escribir "Asignatura 4: " Sin Saltar;
Escribir notas[1,4];
Escribir "Asignatura 5: " Sin Saltar;
Escribir notas[1,5];
```

Imágenes: PSEINT

<https://terepebernal.com/blog/pseudocodigo/algoritmos-pseint-lo-mas-basico/>

El resultado sería:

```
PSeInt - Ejecutando proceso NOTAS_ALUMNOS
*** Ejecución Iniciada. ***
Las notas del alumno 1 son:
Asignatura 1: 8
Asignatura 2: 4.5
Asignatura 3: 9.2
Asignatura 4: 7
Asignatura 5: 3
*** Ejecución Finalizada. ***
```

Imagen: Algoritmos de PSeINT

<https://terepebernal.com/wp-content/uploads/2019/11/ejemplo-notas.jpg>



1.1.8 Algunas instrucciones o comandos simples de los algoritmos con PSeInt

Existen otras palabras claves que realizan acciones simples, todas ellas junto con las anteriores, son las llamadas secuenciales. Borrar Pantalla se utiliza para limpiar la pantalla, dejarla vacía como el que limpia una pizarra. Esperar se utiliza para parar el algoritmo un intervalo de tiempo, que se indica a continuación junto a la unidad de dicho intervalo, que puede ser *segundos* y *milisegundos*, por ejemplo: *Esperar 5 Segundos*; Esperar Tecla se utiliza también para parar el algoritmo hasta que el usuario pulsa una tecla.

1.1.9 Expresiones y operadores de los algoritmos con PSeInt

Las expresiones son combinaciones de constantes, variables y operadores que nos permiten trabajar con los datos. Dependiendo de los operadores utilizados en ellas, pueden ser de varios tipos: aritméticas, relacionales, lógicas, alfanuméricas y de asignación.

<p>Expresión aritmética Aquella en la que se utilizan operadores aritméticos y como operandos datos numéricos. +, -, *, /, ^, % o MOD</p>
<p>Expresión relacional Aquella en la que se utilizan operadores relacionales y el resultado de esta expresión siempre será verdadero o falso. >, <, >=, <=, =, <></p>
<p>Expresión lógica Aquella en la que se utilizan exclusivamente operadores lógicos y el resultado también será siempre verdadero o falso. Y o &, O o , NO o ~</p>
<p>Expresión alfanumérica Aquella que se utiliza para unir cadenas de texto. Se usa el operador de concatenación y como operandos, cadenas de texto. +</p>
<p>Expresión de asignación Aquella que se utiliza para asignarle un valor a una variable. Se usa el operador de asignación y como operandos el nombre de variable a la izquierda y el valor que se le asigna a la derecha. <-</p>

Imágenes: PSEINT

<https://terepebernal.com/blog/pseudocodigo/algoritmos-pseint-lomas-basico/>



1.1 LENGUAJE DE PROGRAMACIÓN

Por: David Gerónimo

Un lenguaje de programación es un medio formal que permite a un programador escribir instrucciones en forma de algoritmos para controlar el comportamiento de un sistema informático y obtener o ejecutar diversas clases de datos o tareas.

1.1.1 Tipos de algoritmos

Los algoritmos se clasifican en cuantitativos y cualitativos en función de cómo describen los pasos a seguir, ya sea a través de cálculos matemáticos o secuencias lógicas. Además, se divide en computacionales y no computacionales en base a si necesitan o no un ordenador para ser resueltos. Antes de escribir el código en programación, es necesario definir el algoritmo informático, es decir, encontrar la forma de resolver el problema. Luego, el código indica a la máquina qué acciones realizar. Por lo tanto, un programa informático es simplemente una colección de algoritmos ordenados y codificados en un lenguaje de programación que se ejecutan en un ordenador. Sin embargo, los algoritmos no son exclusivos de las áreas de matemáticas, lógica y computación. Los utilizamos en la vida cotidiana, como en manuales de instrucciones o recetas culinarias.

1.1.2 Partes de un algoritmo informático

Un algoritmo se compone de tres partes: Input, Proceso y Output. La Input es la información que se proporciona al algoritmo para que trabaje y produzca la solución esperada. El Proceso es el conjunto de pasos que se sigue para transformar los datos de entrada y obtener la solución. La Output es el resultado final, que es el producto de la transformación de los valores de entrada durante el proceso. Por lo tanto, un algoritmo informático comienza con un estado inicial y valores de entrada, seguido de una serie de pasos, y finaliza con un estado en el que se ha encontrado una solución

1.1.3 Características de los algoritmos

Además, los algoritmos tienen ciertas características en común, tales como ser precisos, objetivos y sin ambigüedad; ser ordenados, con una secuencia clara y precisa para llegar a la solución; ser finitos, tener un número determinado de pasos; ser concretos, proporcionar una solución específica para el problema planteado, y ser definidos, producir el mismo resultado cuando se recibe la misma entrada.

1.1.4 Tipos de algoritmos y ejemplos

Existen varias categorías de algoritmos basadas en diferentes criterios. Una categorización se hace en base a cómo describen los pasos para llegar a la solución y se divide en algoritmos cuantitativos y cualitativos; dependiendo de si utilizan cálculos matemáticos o secuencias lógicas. Otro criterio de clasificación se basa en si requieren o no un ordenador para su resolución y se separan en algoritmos computacionales y no computacionales. Sin embargo, si consideramos su diversión



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: <https://acortar.link/dghpLC>

1.1.5 Algoritmos de búsqueda

En resumen, los algoritmos de búsqueda buscan y encuentran uno o varios elementos específicos dentro de una estructura de datos. Existen dos tipos principales de algoritmos de búsqueda: la búsqueda secuencial y la búsqueda binaria. La búsqueda secuencial implica comparar el elemento buscado con cada elemento del conjunto hasta encontrarlo. La búsqueda binaria implica comparar el elemento buscado con el elemento ubicado en el medio de un conjunto de elementos ordenados y continuar la búsqueda en un intervalo cada vez más pequeño hasta encontrarlo.

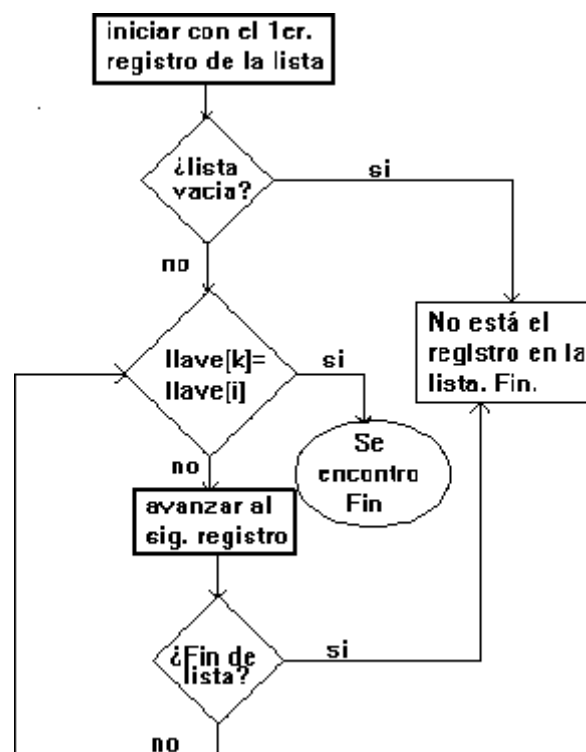


Imagen: Ejemplo del tamaño solicitado (información)

Fuente: <https://acortar.link/bHxpcv>

1.1.6 Algoritmos de ordenamiento

Algoritmos de ordenamiento reorganizan los elementos de una lista de acuerdo a una relación de orden, siendo las formas más comunes de ordenar la numérico y la lexicográfica

Entre los algoritmos de ordenamiento más conocidos se encuentran:

- Ordenamiento de burbuja: compara cada elemento de la lista con el siguiente y cambia su posición si no están en el orden correcto, revisándolo varias veces hasta que no se requieran más cambios.

- Ordenamiento por selección: coloca el elemento más pequeño disponible en cada posición consecutiva de la lista.

1.1.7 Algoritmos voraces

Los algoritmos voraces se basan en una estrategia de búsqueda que selecciona la opción más óptima en cada paso para llegar a una solución global óptima.

Ejemplos de algoritmos voraces incluyen:

- Algoritmo de Dijkstra, que se utiliza para encontrar el camino más corto desde un vértice origen a los demás vértices en un grafo con pesos en cada arista.
- Codificación Huffman, un método de compresión de datos que analiza la frecuencia de aparición de caracteres en un mensaje y les asigna códigos de longitud variable. Los caracteres con mayor frecuencia tienen códigos más cortos.

Sin embargo, estos algoritmos a veces no encuentran la solución global óptima, ya que toman decisiones solo basadas en la información disponible en el momento y no en la futura.

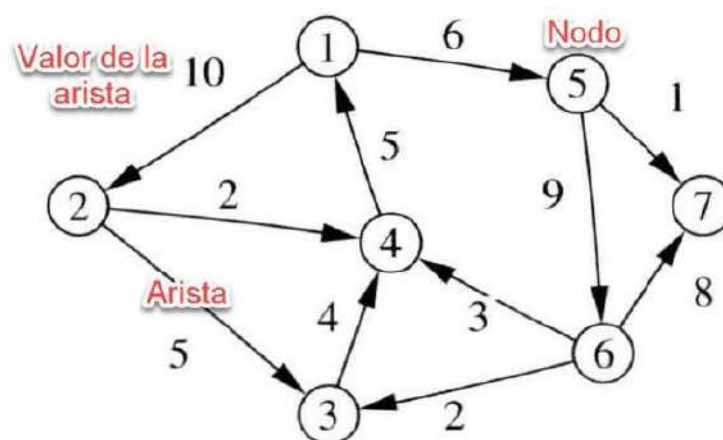


Imagen: Ejemplo del tamaño solicitado (información)
Fuente: <https://acortar.link/jzvhO3>

1.1.8 Programación dinámica

La programación dinámica es un enfoque para resolver problemas que consiste en dividir un problema complejo en subproblemas y calcular y almacenar sus soluciones para evitar tener que calcularlos de nuevo. Esto reduce el tiempo de ejecución de un

algoritmo al optimizar la recursión. Sin embargo, para que la programación dinámica sea efectiva, el problema debe tener subestructuras óptimas y subproblemas superpuestos, es decir, las soluciones óptimas de los subproblemas deben ser utilizables para encontrar la solución óptima del problema en su totalidad, y el problema debe ser divisible en subproblemas que se reutilizan para producir el resultado final.

1.1.9 Algoritmos probabilísticos

La técnica de programación probabilística implica el uso de una fuente de aleatoriedad en la lógica del algoritmo. Por medio de un muestreo aleatorio de la entrada, se puede llegar a una solución que no sea perfecta, pero que es adecuada para el problema.

Esta técnica se utiliza en situaciones en las que hay limitaciones de tiempo o memoria y se puede aceptar una solución aceptable en promedio. Dado que los mismos datos pueden dar lugar a soluciones diferentes y algunas de ellas pueden ser incorrectas, se repite el algoritmo varias veces con diferentes submuestras aleatorias para aumentar la probabilidad de obtener una solución correcta.

Hay dos tipos principales de algoritmos probabilísticos: el algoritmo de Monte Carlo y el algoritmo de Las Vegas. El algoritmo de Monte Carlo tiene una pequeña probabilidad de no acertar o no encontrar una solución, pero se puede reducir aumentando el tiempo de cálculo. Por otro lado, el algoritmo de Las Vegas se ejecuta en un período de tiempo determinado y, si encuentra una solución en ese tiempo, será correcta, pero es posible que se agote el tiempo y no se encuentre ninguna solución.

2.2 SEUDOCÓDIGO

En ciencias de la computación, y análisis numérico, el pseudocódigo es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo. El pseudocódigo es una forma de describir un algoritmo que se parece a un lenguaje de programación real, pero está diseñado para ser fácilmente comprensible para los humanos. Se utiliza para omitir detalles innecesarios para la comprensión humana y puede incluir descripciones detalladas en lenguaje natural o notación matemática. Es común en libros de texto, publicaciones científicas y en la planificación del desarrollo de software. Ayuda a los programadores a describir la estructura de un programa antes de escribir el código efectivo. Sin embargo, el pseudocódigo no es programable en sí mismo, sino que facilita la programación.

2.1 Aplicaciones

El pseudocódigo se utiliza para describir la forma y estructura de un algoritmo antes de escribirlo en un lenguaje de programación. Los programadores pueden iniciar un proyecto describiendo el código en pseudocódigo antes de escribirlo, lo que les ayuda a tener una visión clara y completa del proyecto antes de comenzar a escribir código real. Es una herramienta útil para programadores que aplican algoritmos desconocidos, ya que les permite traducir la descripción en pseudocódigo a su lenguaje de programación

2.2 Sintaxis

En general, el pseudocódigo no sigue las reglas sintácticas de ningún lenguaje de programación específico y no es estandarizado de manera sistemática, aunque los autores a veces utilizan estructuras, sintaxis y estilo de lenguajes de programación convencionales. Si se desea ejecutar, el pseudocódigo debe ser convertido en un lenguaje de programación para evitar errores. Las sintaxis más populares incluyen Pascal, BASIC, C, C++, Java, Lisp y ALGOL. Por lo general, se omiten las declaraciones de variables, y a veces las funciones, bloques de código y código dentro de un loop son reemplazados por una sentencia en lenguaje natural.

2.3 Funciones y operaciones

Cada escritor emplea su propio código ficticio con sus convenciones correspondientes.

2.4 Estructuras de control

- Estructuras secuenciales: Las órdenes se realizan en un orden fijo generalmente determinado por el número de línea, es decir, las órdenes se llevan a cabo de arriba hacia abajo.
- Estructuras selectivas: Las estructuras de selección son órdenes que pueden ser ejecutadas o no, dependiendo de si se cumple o no una determinada condición.

- Selectiva doble (alternativa): ejecuta una de dos posibles órdenes, dependiendo de si se cumple o no una condición específica.
- Selectiva múltiple: También es frecuente el uso de una selección múltiple, que equivale a anidar varios procesos de selección. En este caso, hay una serie de condiciones que deben ser mutuamente excluyentes, es decir, si una de ellas se cumple, las demás necesariamente tienen que ser falsas, y solo hay un caso en el que todas las condiciones son falsas.

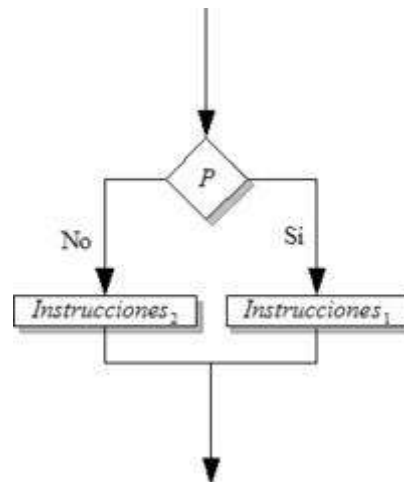


Imagen: Ejemplo del tamaño solicitado (información)
Fuente: <https://acortar.link/SlogdZ>

2.5 Estructuras iterativas

Las estructuras iterativas representan la realización de órdenes más de una vez.

- Bucle mientras: El ciclo se repite mientras la condición sea verdadera, si al llegar por primera vez al ciclo y la condición es falsa, el bloque de código dentro del ciclo nunca se ejecuta.
- Bucle repetir: Existen otras formas derivadas de la anterior. La estructura de control "repetir" se utiliza cuando es necesario que el bloque de código dentro del ciclo se ejecute al menos una vez y continúe hasta que se cumpla la condición.
- Bucle hacer: El ciclo "hacer-mientras" se utiliza para repetir un bloque de instrucciones mientras se cumpla una determinada condición.

- Bucle para: Una estructura de control muy común es el bucle "FOR", que se utiliza cuando se desea repetir un bloque de código un número determinado de veces, usando una variable como índice que aumenta (o disminuye) en cada iteración.
- Bucle para cada: Finalmente, también es frecuente usar la estructura de control "para cada". Esta sentencia se utiliza cuando se tiene una lista o un conjunto y se desea repetir un bloque de código para cada elemento de la lista o conjunto.

2.6 El anidamiento

El pseudocódigo del algoritmo de ordenamiento de burbuja consiste en varias estructuras anidadas, donde cualquier instrucción puede ser reemplazada por una estructura de control. Este algoritmo clasifica los elementos de una lista de menor a mayor. Las estructuras anidadas están identificadas por una sangría y una conexión mediante flechas entre los delimitadores de cada nivel de anidamiento.

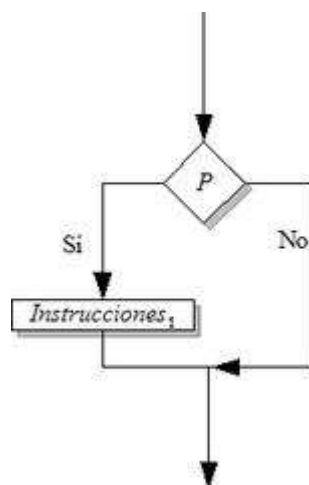


Imagen: Ejemplo del tamaño solicitado (información)
Fuente: <https://acortar.link/RgeCBf>

3.1 Diagramas de flujo

Un Diagrama de Flujo es una representación gráfica de un algoritmo que muestra los pasos o procesos necesarios para resolver un problema. Su construcción es crucial ya que permite traducir el algoritmo a un código en un Lenguaje de Programación.

3.2 símbolos utilizados en los diagramas de flujo

- Un símbolo de inicio o fin en un diagrama de flujo representa el comienzo o el final de un algoritmo. También puede indicar una pausa o una interrupción planificada que deba ser realizada en un programa.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

- Este es utilizado para representar un proceso específico, y suele ser utilizado para representar una instrucción o cualquier tipo de operación que cause un cambio en un valor.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

- Este símbolo se usa para representar una operación de entrada o salida.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

- Este símbolo se utiliza para representar decisiones y ramificaciones basadas en operaciones lógicas o comparaciones de datos.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

- Este símbolo es utilizado para unir dos secciones de un diagrama a través de un conector de salida y un conector de entrada, creando así un vínculo en la misma página del diagrama.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

- Este símbolo se utiliza para conectar dos partes de un diagrama que no están en la misma página, mediante un conector de salida y un conector de entrada.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

- Este símbolo se usa para señalar el orden en que se deben realizar las operaciones en el diagrama de flujo, es decir, para indicar la dirección en que se deben seguir las acciones dentro del algoritmo.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

- Este símbolo se utiliza para indicar la impresión de datos en una impresora.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

- Este símbolo se utiliza para representar la visualización de información en la pantalla o en el monitor.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: uv.mx/personal/aherrera/files/2020/05/DIAGRAMAS-DE-FLUJO.pdf

4.1 Deontología Informática

La Deontología informática es un conjunto de deberes y normas éticas que rigen a los profesionales que trabajan con tecnología y medios masivos de comunicación, como internet, y que se especializan en una determinada rama de la informática.

La deontología informática se enfoca en el comportamiento ético y moral de los profesionales que trabajan con tecnología y medios de comunicación, en especial, con la información que es un recurso valioso para las empresas. Históricamente, los profesionales de la informática no han abordado cuestiones éticas, pero su importancia aumenta a medida que las computadoras influyen en la vida de las personas. La deontología informática se ocupa de regular el uso de los sistemas informáticos que tienen impactos que van más allá de lo imaginado.



Imagen: Ejemplo del tamaño solicitado (información)
Fuente: <https://acortar.link/5FnR1T>

1.2.1 Deontología

Por: Patrick Leonardo

En un sentido más concreto, tiene que ver con el comportamiento moral o ético, es decir con los principios y normas morales que regulan las actividades humanas. La deontología informática, por extensión, trata, por tanto, de la moral o ética profesional en el manejo del activo más importante que tienen las empresas, un bien cada vez más apreciado, que es la información. Históricamente, los profesionales de la informática, como grupo, no han sido involucrados en cuestiones de ética. A menudo, se ve, a las computadoras simplemente como máquinas y algoritmos y no se perciben las serias cuestiones de éticas inherentes a su utilización. De cualquier modo, cuando se considera que esas máquinas influyen, directa e indirectamente, en la calidad de vida de millones de individuos se comprende que es un tema de gran importancia. Las computadoras se utilizan para diseñar, analizar, soportar y controlar las aplicaciones que protegen y guían las vidas de las personas. El uso de los sistemas informáticos puede tener efectos más allá de lo que alcanza la imaginación.

1. Deontologia

Es en su sentido más amplio, la ciencia o tratados de los deberes y normas morales. En un sentido más concreto, tiene que ver con el comportamiento moral o ético, es decir con los principios y normas morales que regulan las actividades humanas.



Imagen: Deontologia informática y virus

Fuente:

https://www.google.com/search?q=deontologia+informatica&tbm=isch&ved=2ahUKewitmL6y34T9AhVBjeAKHS2iDs_kQ2-cCegQIABAA&oq=deontologia+inf&gs_lcp=CgNpbWcQARgBMgUIABCABDIFCAAQgAQyBQgAEIAEOgQllxAnOgQIABBD_OgYIABAIEB46BAgAEB46BwgAEIAEEBhQjwNYkwpglxRoAHAAeACAAeQBiAGmBpIBBTauMy4ymAEA_oAEBggELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=DvfiY63ZOMGaggetxLrIDA&rlz=1C1CHZN_esGT938GT938#imgrc=H5aw0Rk2m-EskM

2. Deontología informática-Etica profesional de la informática

Por extensión, trata, por tanto de la moral o ética profesional en el manejo del activo más importante que tienen las empresas, un bien cada vez más apreciado, que es la información.

Históricamente, los profesionales de la informática, como grupo, no han sido involucrados en cuestiones de ética. A menudo, se ve a las computadoras simplemente como maquinas y algoritmos y no se perciben las serias cuestiones éticas inherentes a su utilización.

De cualquier modo, cuando se considera que esas maquinas influyen, directa e indirectamente, en la calidad de vida de millones de individuos se comprende que es un tema de gran importancia. Las computadoras se utilizan para diseñar, analizar, soportar y controlar aplicaciones que protegen y guían las vidas y finanzas de las personas. El uso de los sistemas informáticos puede tener efecto más allá de lo que alcanza la imaginación. Así pues, se debe reconsiderar la actitud sobre los actos que demuestran una carencia de respeto a los derechos y privacidad de las computadoras y datos de otras personas.



Imagen: Deontología informática-deontologia_danicarito

Fuente:

https://www.google.com/search?q=deontologia+informatica&tbm=isch&ved=2ahUKewitmL6y34T9AhVBjeAKHS2iDs kQ2-cCegQIABAA&oq=deontologia+inf&gs_lcp=CgNpbWcQARgBMgUIABCABDIFCAAQgAQyBQgAEIAEOgQIIXAnOgQIABBD OgYIABAIEB46BAgAEB46BwgAEIAEEBhQjwNYkwpglxRoAHAAeACAAeQBiAGmBpIBBTauMy4ymAEA oAEBagELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=DvfiY63ZOMGaggetxLrIDA&rlz=1C1CHZN esGT938GT938#imgrc=184Rw30i8ma8SM

3. El delito informático

El delito informático se puede definir como la manipulación de los registros informáticos que contienen información referente a los activos de una organización o empresa objeto del delito, para ocultar el robo o desfalco realizado.

- Manipulación de datos de entrada.
- Cambios no autorizados en los programas.
- Transmisión ilegal, interceptación o desviación de la información tele procesada.
- Alteración de los ficheros de datos.



Imagen: Cuales son los delitos informáticos mas comunes?

Fuente:

https://www.google.com/search?q=El+delito+informatico&tbm=isch&ved=2ahUKewjklm334T9AhVUwikDHaqVAdw_Q2-cCegQIABAA&og=El+delito+informatico&gs_lcp=CgNpbWcQA1CACFiACGCRDGgAcAB4AIABAIgBAJBAJgBAKABAaoBC_2d3cy13aXotaW1nwAEB&sclient=img&ei=GPfiY-WkIdSEp8kPqquG4A0&rlz=1C1CHZN_esGT938GT938#imgrc=SLp7SjW2rEGrBM

4. La informatica

Es la ciencia que estudia el fenómeno de la información, los sistemas de información y el procesamiento, transferencia y utilización de la información, principalmente, pero no necesariamente mediante computadores y telecomunicaciones como herramientas para el beneficio de la humanidad.

4.1. Etica en la informatica

Se define como la disciplina que identifica y analiza los impactos de las tecnologías de la información en los valores humanos y sociales.



Imagen: Objetivo de la ética informática

Fuente:

https://www.google.com/search?q=etica+en+la+informatica&tbm=isch&ved=2ahUKewi38PvY5YT9AhXfwikDHUTDB_mUQ2-cCegQIABAA&oq=etica+en+la+in&gs_lcp=CgNpbWcQARgCMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIECAAQzOLCAAQgAQQsQMgQwE6CAgAEIAEELDQggIABCxAXCDAVCLBViRF2CLJmgAcAB4A4ABuAalAewfkgENMC4xLjcuMC4yLjEuMZgBAKABAqoBC2d3cy13aXotaW1_nwAEB&scient=img&ei=qv3iY7zDt-Fp8kPxlabqAY&rlz=1C1CHZN_esGT938GT938#imgrc=oa7kO2GSCyQuIM

1.2.2 Medidas de contingencia y seguridad

Un Plan de Contingencia dentro del plan estratégico de una empresa prevé un conjunto de medidas y decisiones a tomar, por parte de tu empresa, cuando ésta atraviesa por un incidente o situación de crisis, con el objetivo de garantizar su continuidad.

1. Plan de contingencia

Un plan de contingencia es un conjunto de procedimientos alternativos a la operatividad normal de cada institución.

Su finalidad es la de permitir el funcionamiento de esta, aun cuando alguna de sus funciones deje de hacerlo por culpa de algún incidente tanto interno como ajeno a la organización.

Todas las instituciones deberían contar con un plan de contingencia actualizado, valiosa herramienta en general basada en un análisis de riesgo.

Permitirá ejecutar un conjunto de normas, procedimientos y acciones básicas de respuesta que se debería tomar para afrontar de manera oportuna, adecuada y efectiva, ante la eventualidad de incidentes, accidentes y/o estados de emergencias que pudieran ocurrir tanto en las instalaciones como fuera de ella, por ejemplo el secuestro de un funcionario.

Los riesgos los puedes eliminar, transferir, mitigar o aceptar. Ello dependerá de varios factores tales como la probabilidad de ocurrencia o impacto del riesgo.

Un Plan de Contingencia de Seguridad Informática consiste los pasos que se deben seguir, luego de un desastre, para recuperar, aunque sea en parte, la capacidad funcional del sistema, aunque, y por lo general, constan de reemplazos de dichos sistemas.

2. Los objetivos del plan de contingencia

Los objetivos del plan de contingencia son el de planificar y describir la capacidad para respuestas rápidas, requerida para el control de emergencias. Paralelo al plan se debe identificar los distintos tipos de riesgos que potencialmente podrían ocurrir e incorporar una estrategia de respuesta para cada uno, con algunos objetivos específicos:

- Establecer un procedimiento formal y por escrito que indique las acciones a seguir frente a determinados riesgos.
- Optimizar el uso de recursos humanos y materiales
- Un control adecuado para cumplir con las normas y procedimientos establecidos.

Los planes de contingencia son necesarios en todo sistema y no podría dejarse de lado en el tema de seguridad.



Imagen: Plan de contingencia para hacersegurar la continuidad del negocio

Fuente:

https://www.google.com/search?q=plan+de+contingencia&tbm=isch&ved=2ahUKewj7sur25oT9AhXMh-AKHQDDA0IQ2-cCegQIABAA&oq=plan+de+contingencia&gs_lcp=CgNpbWcQAzIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQ6BAgAEEM6BwgAEIAEEBhQtgRYtgRgrAZoAHAAeACAAbkBiAG4ApIBAzAuMpgBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=9f7iY7vkEsyP_ggeAho-QBA&rlz=1C1CHZN_esGT938GT938#imgrc=2jiUUg4mMuRzqM

3. Cuales son las etapas de un plan de contingencia?

Haciendo una síntesis para su elaboración la podríamos dividir en cinco etapas.

Evaluación
Planificación
Pruebas de viabilidad
Ejecución
Recuperación

Las tres primeras etapas hacen referencia al componente preventivo y las ultimas a la ejecución del plan una vez ocurrido el siniestro.

4. Que medidas se pueden usar para la seguridad informática

Hay 5 medidas esenciales las cuales son:

- Hacer copias de seguridad o respaldos.
- Fomentar una cultura de contraseñas robustas.
- Proteger el correo electrónico.
- Utilizar antivirus.
- Controlar el acceso a la información.



Imagen: Plan de contingencia para hacersegurar la continuidad del negocio

Fuente:

https://www.google.com/search?q=medidas+de+contingencia+y+seguridad+informatica&rlz=1C1CHZN_esGT938GT_938&sxsrf=AJOqIzXOj47KTJB8iFWF64nzseL-B-7bBg:1675822275593&source=Inms&tbn=isch&sa=X&ved=2ahUKewjk0MW77IT9AhX0SzABHZLpC5UQ_AUoAXoECA_EQAw&biw=1366&bih=657&dpr=1#imgrc=HY7kvQ-pQbW2LM

1.2.1 Seguridad industrial en informática

Se refiere a la protección y el fortalecimiento de los ordenadores y sistemas basados en Internet contra el acceso no intencionado o no autorizado, modificaciones, robo y la obliteración.

1. Instalar un software antivirus confiable

Un buen programa antivirus confiable es un requisito básico de cualquier sistema de seguridad cibernética. Aparte, el software anti-malware también es esencial. Trabajan como una frontera final para la defensa de ataques no deseados, si pasan a través de la red de seguridad de la empresa. Además, su cometido es la detección y eliminación de virus y malware, adware y spyware. También exploran y filtran descargas y correos electrónicos potencialmente dañinos.



Imagen: Software seguro

Fuente:

https://www.google.com/search?q=instalar+un+software+antivirus+confiable&tbm=isch&ved=2ahUKewiCgMme64T9AhUSxSkDHaluCO8Q2-cCegQIABAA&oq=instalar+un+software+antivirus+confiable&gs_lcp=CgNpbWcQAzIFCAAQgAQ6BAgAEENQ6QhY6QhgIQpoAHAAeACAAXyIAe8BkgEDMC4ymAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=egPiY4LbF5KKp8kPot_2g-A4&rlz=1C1CHZN_esGT938GT938#imgrc=fUeyEjVa6CsTJM

2. Utilizar contraseñas complejas

Casi todas las aplicaciones basadas en un ordenador y en la Web requieren una clave para acceder a ellas. Bien se trate de las respuestas a las preguntas de seguridad o de las contraseñas, hay que asegurarse de crear contraseñas complejas para que sea difícil para los hackers vulnerarlas.

Para obtener respuestas a preguntas de seguridad, se puede considerar traducirlas a otro idioma utilizando herramientas de traducción online gratuitas. Esto puede hacer que sean impredecibles y difíciles de descifrar, y menos susceptibles a la ingeniería social.

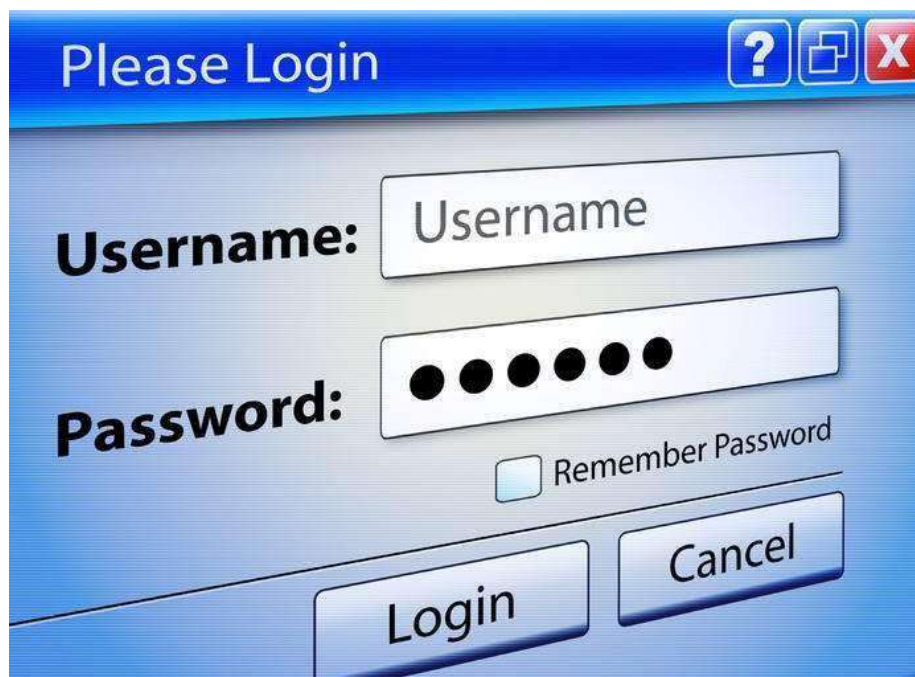


Imagen: Contraseñas difíciles de descubrir

Fuente:

https://www.google.com/search?q=Utilizar+contrase%C3%B1as+compleja&rlz=1C1CHZN_esGT938GT938&sxsrf=AJOqlzWMRwlkr_OTFz1OjuCVhTDIbLMzeQ:1675823621853&source=lnms&tbm=isch&sa=X&ved=2ahUKEwJn8L698YT9A_hXrmYQIHb4cBAMQ_AUoAXoECAEQAw#imgrc=dcQPZh-gdlgQiM

3. Protegerte con firewall

El Firewall es necesario, ya que ayuda a proteger el tráfico de red – entrante y saliente. Puede impedir que los hackers ataquen tu red bloqueando ciertos sitios web. También puede programarse para que el envío de datos de propiedad y correos electrónicos confidenciales de la red de tu empresa esté restringido.



Imagen: El mejor firewall

Fuente:

https://www.google.com/search?q=Protegerte+con+Firewall&tbm=isch&ved=2ahUKEwiewJq-8YT9AhVCHd8KHeZgBNYQ2-cCegQIABAA&oq=Protegerte+con+Firewall&gs_lcp=CgNpbWcQAzIECCMQJ1DaBVjaBWCjB2gAcAB4AIABtgKIAbYCKgEDMy0xmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=BwrjY96gFcK6_AbmwZGwDQ&rlz=1C1CHZN_esGT9_38GT938#imgrc=wJyrNfTYOSanVM

4. Instalar software de cifrado

Si gestionas datos relativos a tarjetas de crédito, cuentas bancarias y números de la seguridad social en una base de datos, tiene sentido tener un programa de cifrado para mantener los datos seguros cambiando la información del ordenador a códigos ilegibles.

De esta forma, incluso si se roban los datos, será inútil para el hacker, ya que no tendrá las claves para descifrar los datos y descifrar la información.



Imagen: software

Fuente:

https://www.google.com/search?q=instalar+software+de+cifrado&rlz=1C1CHZN_esGT938GT938&xsrf=AJOqzXazc6TdkoWhlyNvYS2lrHI7lhANg:1675823745294&source=lnms&tbn=isch&sa=X&ved=2ahUKEwigi6348YT9AhWqfTABHW_DXDmkQ_AUoAnoECAMQBA#imgrc=gOZZ2JiUg2_DxM

5. Ignorar los correos electrónicos sospechosos

Hay que habituarse a no abrir nunca ni responder correos electrónicos de aspecto sospechoso, incluso si parecen ser de un remitente conocido. Si abres el correo electrónico, no hagas clic en enlaces sospechosos ni descargues archivos adjuntos. Si lo haces, puedes convertirte en una víctima de robo de dinero y de identidad online, incluyendo las «estafas de phishing».

Los correos electrónicos de phishing son los que parece que proceden de remitentes fidedignos, como un banco o alguien con quien puedes haber hecho negocios. A través de ellos, los hackers intentan adquirir tus datos privados y financieros, como datos de cuentas bancarias y números de tarjetas de crédito.



Imagen: Correos extraños

Fuente:

https://www.google.com/search?q=ignorar+los+correos+electr%C3%B3nicos+sospechosos&tbm=isch&ved=2ahUK_EwjE18oT9AhUNneAKHZV7AusQ2-cCegQIABAA#imgrc=O67-KAZUFSPX8M

6. Limitar el acceso a datos críticos

En las empresas, el número de personas con acceso a datos críticos debe ser mínimo, el Director de la empresa y ciertas personas de confianza.



Imagen: Tener límites al momento de accionar

Fuente:

[hcCegQIABAA&oq=Limitar+el+acceso+a+datos+cr%C3%ADticos&gs_lcp=CgNpbWcQAzoECCMQJ1C8CVi8CWDJC2gAcAB4AIABflgB7QGSAQMwLjKYAQCgAQGgAQtnD3Mtd2l6LWltZ8ABAQ&sclient=img&ei=JAvjY6CDFIuEp8kP2Y6zqAQ#imgrc=klyw9jzRTehSkM](https://www.google.com/search?q=Limitar+el+acceso+a+datos+cr%C3%ADticos&gs_lcp=CgNpbWcQAzoECCMQJ1C8CVi8CWDJC2gAcAB4AIABflgB7QGSAQMwLjKYAQCgAQGgAQtnD3Mtd2l6LWltZ8ABAQ&sclient=img&ei=JAvjY6CDFIuEp8kP2Y6zqAQ#imgrc=klyw9jzRTehSkM)

7. Hacer una copia de seguridad regularmente

Cada semana, hay que realizar una copia de seguridad de los datos en un disco duro externo o en la propia nube, o programar copias de seguridad automáticas para garantizar que la información se almacene de forma segura.



Imagen: seguridad estable

Fuente:

https://www.google.com/search?q=Hacer+una+copia+de+seguridad+regularmente&tbm=isch&ved=2ahUKEwjmrXp_8oT9AhXZyCkDHcGkAbUQ2-cCegQIABAA&oq=Hacer+una+copia+de+seguridad+regularmente&gs_lcp=CgNpbWcQAzoECCMQJ1DVAIjVAmD1BWgAcAB4AIABaYgBzwGSAQMwLjKYAQcGAAQgAQtdn3Mtd2l6LWltZ8ABAQ&sclient=img&ei=bgvjY-KrGNmRp8kPwcmGqAs#imgrc=U6bW983KvQv0kM

8. Protección de la red Wi-fi

Para proteger la red Wi-Fi de infracciones por parte de piratas informáticos, hay que cambiar el nombre de su punto de acceso inalámbrico o enrutador, también llamado Identificador de conjunto de servicios (SSI)



Imagen: seguridad estable

Fuente:

https://www.google.com/search?q=Protecci%C3%B3n+de+la+red+Wi-Fi&tbm=isch&ved=2ahUKEwjY58KL84T9AhVlc98KHbcOB14Q2-cCegQIABAA&oq=Protecci%C3%B3n+de+la+red+Wi-Fi&gs_lcp=CgNpbWcQAzoECCMQJ1CcB1icB2C5CGgAcAB4AIAB2AGIACmCkgEFMC4xLjGYAQCgAAQgAQtdn3Mtd2l6LWltZ8ABAQ&sclient=img&ei=tQvjY5i5OMjm_Qa3nZzwBQ#imgrc=HdV7Ue8OjF8E1M

9. Portátiles y smartphones seguros

Los ordenadores portátiles y los smartphones contienen una gran cantidad de datos valiosos, y esa es también la razón por la que están en un mayor riesgo de perder datos o de que se roben. La protección de ambos dispositivos implica encriptación, protección por contraseña y habilitación de la opción de «limpieza remota».



Imagen: portátil seguro

Fuente:

https://www.google.com/search?q=Port%C3%A1tiles+y+Smartphones+seguros&rlz=1C1CHZN_esGT938GT938&sxsrf=AJOqlzXLWjtcUuJ3CIKPeBasP4ZdWjdX6w:1675824164487&source=lnms&tbn=isch&sa=X&ved=2ahUKEwji1Z7A84T9AhUUZTABHYUcAbIQ_AUoAXoECAEQAw#imgcr=xNCIPEb6itysyM

10. Comunicar políticas de seguridad

Tener una política escrita de seguridad informática puede no ser suficiente, tienes que asegurarte de que sus detalles sean comunicados y entendidos por tus empleados, para que puedan ponerlos en práctica. Esa es la única manera de hacer efectivas estas políticas. También hay que realizar la modificación periódica de estas políticas en función de la relevancia de los contenidos.

1.2.1 Introducción a la seguridad industrial

Uno de los activos más valiosos para cualquier empresa es la información que maneja. La información es el conjunto de datos que da sentido a una empresa, datos que la definen, datos con los que trabaja y datos que, en manos inadecuadas, pueden llevar a la misma a la ruina. Extendiendo este concepto de seguridad al mundo de las telecomunicaciones y la informática, puede entenderse desde dos puntos de vista: seguridad de la información y seguridad informática.

La seguridad de la información es el conjunto de medidas y procedimientos, tanto humanos como técnicos, que permiten proteger la integridad, confidencialidad y disponibilidad de la información:

- Integridad: certificando que tanto la información como sus métodos de proceso son exactos y completos.
- Confidencialidad: asegurando que únicamente pueden acceder a la información y modificarla los usuarios autorizados.
- Disponibilidad: permitiendo que la información esté disponible cuando los usuarios la necesiten.

Este término, por tanto, es un concepto amplio que engloba medidas de seguridad que afectan a la información independientemente del tipo de esta, soporte en el que se almacene, forma en que se transmita, etc. La seguridad informática, por su parte, es una rama de la seguridad de la información que trata de proteger la información que utiliza una infraestructura informática y de telecomunicaciones para ser almacenada o transmitida. Podemos distinguir los siguientes tipos:

– En función de lo que se quiere proteger:

- Seguridad física: se asocia a la protección física del sistema ante amenazas como inundaciones, incendios, robos, etc.
- Seguridad lógica: mecanismos que protegen la parte lógica de un sistema informático (datos, aplicaciones y sistemas operativos). Uno de los medios más utilizados es la criptografía.

-En función del momento en que tiene lugar la protección:

- Seguridad activa: se encarga de prevenir, detectar y evitar cualquier incidente en los sistemas informáticos antes de que se produzca (medidas preventivas). Por ejemplo, utilización de contraseñas.
- Seguridad pasiva: comprende todas aquellas técnicas o procedimientos necesarios para minimizar las consecuencias de un incidente de seguridad (medidas correctoras). Por ejemplo, las copias de seguridad.



Imagen: Seguridad informática

Fuente:

https://www.google.com/search?q=.+Introducci%C3%B3n+a+la+seguridad+industrial+informatic&tbm=isch&ved=2_ahUKEwik8ofs9IT9AhWPmeAKHQGHAF0Q2-cCegQIABAA#imgrc=zDmxq8GwZpTbtM

1.3 Seguridad Industrial en Informática

Por: Joshua Gonzalez

1.3.2 Señalizaciones

Hay 5 tipos principales de señales de seguridad

Colores:

- Azul para las acciones obligatorias.
- Rojo como color de prohibición.
- Amarillo como color de prudencia.
- Verde para las acciones positivas.

Formas:

- Discos o Círculos se usan para las prohibiciones o instrucciones.
- Los triángulos se usan para las advertencias.
- Los Cuadrados y Rectángulos se usan para la señalización de emergencia y de información.

Ahora si, veamos cada tipo por separado con ejemplos concretos.



Imagen: Cartel Publicado por el Ministerio de Trabajo
 Fuente: <https://www.areatecnologia.com/images/se%C3%B1ales-seguridad.jpg>

1.3.3 Mapa de Riesgos

Es un instrumento que, mediante relevamiento y representación de riesgos y agentes contaminantes, permite **localizar los factores** en un espacio de trabajo determinado. Entre los objetivos más importantes de la elaboración de un MAPA DE RIESGOS de una empresa o sector de la misma, se puede enumerar:

1. **Implementar planes y programas de prevención**, en función de las prioridades observadas.
2. **Permitir una identificación, análisis y seguimiento** periódico de los riesgos mediante la implementación de **sistemas de control de gestión de prevención participativos**.
3. Evaluar la eficacia de las **intervenciones preventivas** que se adoptan desde la gestión empresarial
4. Mejorar las condiciones de trabajo a través de la participación de los trabajadores y sus representantes.



Imagen: Eslabones de Prevención y Seguridad

Fuente: <https://seguridadsaludocupacional.weebly.com/uploads/8/8/5/4/88542438/sin-t-tulo.jpg?549>

1.3.4 Seguridad Informática

La seguridad informática es una disciplina que se encarga de proteger sistemas y dispositivos electrónicos, así como la información almacenada en ellos, contra ataques, intrusiones y otros riesgos. Esto incluye medidas de protección contra malware, phishing, hacking, y la protección de datos personales. Es importante para garantizar la confidencialidad, integridad y disponibilidad de la información en la era digital.

La seguridad de la información, como concepto, se basa en cuatro pilares: la disponibilidad, la integridad, la confidencialidad y la autenticación.

Ingeniero en seguridad informática



- Funciones**
 - Formular las políticas de seguridad informática
 - Documentar guías de trabajo y procedimientos de seguridad informática
 - Implementar medidas preventivas y correctivas de incidentes
- Formación**
 - Ingeniería Informática, Ciencias de la Computación, o afines
 - Posgrado en Seguridad Informática y/o Certificaciones reconocidas como CISA, SSCP, CISSP, entre otras.
- Habilidades**
 - Pensamiento crítico y analítico
 - Sentido de urgencia
 - Conocimientos en gestión de riesgos informáticos y de seguridad de la información
 - Protocolos de manejo de incidencias de seguridad informática
- Salario**
 - Junior: 54.000 \$
 - Medio: 74.000 \$
 - Senior: 111.500 \$

Imagen: Cuadro de Información

Fuente: <https://www.freelancermap.com/blog/wp-content/uploads/2021/04/ingeniero-seguridad-informatica.jpg>

1.3.5 Métodos de seguridad informática

Por: Brandon Grijalva

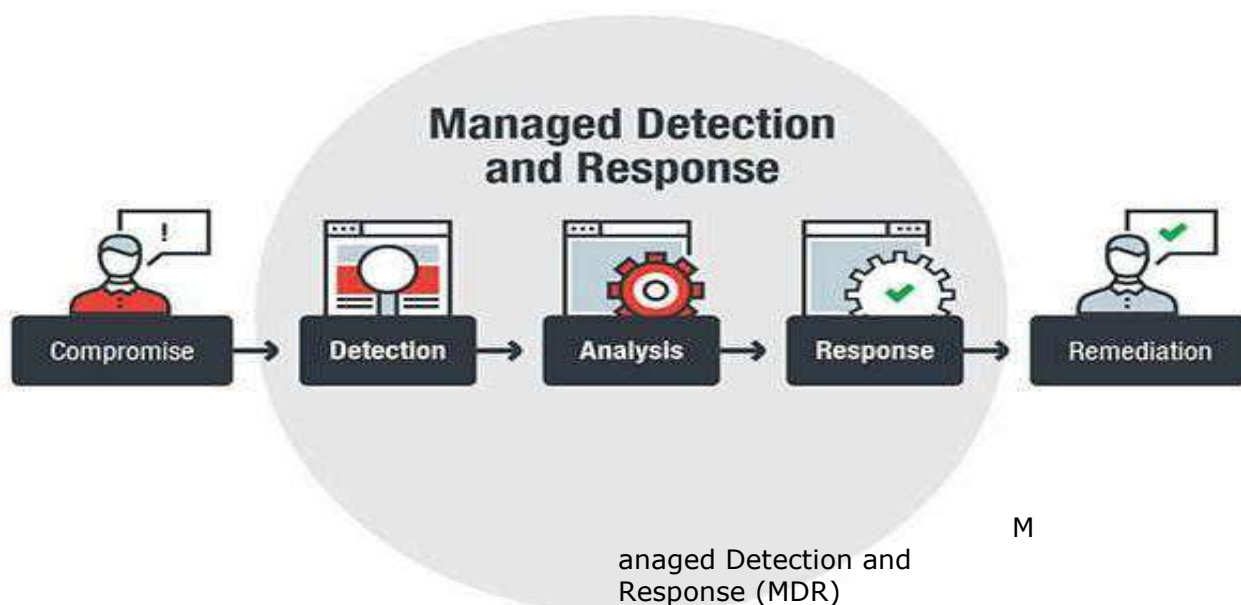
Los ataques informáticos son un gran problema para las empresas, ya que ponen en riesgo su información confidencial y las obligan a gastar millones de dólares para recuperar el control de sus sistemas y datos.

Para que puedas prevenirlos, a continuación, te recomendaré 7 métodos de seguridad para reducir un riesgo de ataque.

1. Inteligencia artificial

La inteligencia artificial tiene la cualidad de adelantarse y predecir futuros ataques en los sistemas más vulnerables. Ante esta ventaja, la empresa puede anticiparse a reforzar estas áreas y evitar daños mayores.

Un ejemplo de lo anterior es la creación del Managed Detection and Response (MDR), un sistema de seguridad avanzado que es capaz de buscar, supervisar, analizar y actuar ante una amenaza de ataque cibernético.



Fuente:

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.trendmicro.com%2Finfo%2Fus%2Fsecurity%2Fdefinition%2Fmanaged-detection-and-response&psig=A0vVaw2sYT4pS6LTVLuBqjDYugaV&ust=1675824195952000&source=images&cd=vfe&ved=0CBAQjRxaFwoTCPIwwbu6gv0CFQAAAAAdAAAAABAD>

2. Software antivirus

Un software antivirus es un recurso imprescindible en cualquier equipo informático, ya sea personal o corporativo.

Esta herramienta protege de distintos ataques, como troyanos, worms, adware, ransomware, entre otros programas dañinos. El antivirus previene, protege y elimina cientos de softwares maliciosos que constantemente bombardean tu información.

Pagar por un buen antivirus es una fuerte inversión para la empresa, pero la seguridad de la información empresarial es más valiosa y es muy importante.



Antivirus

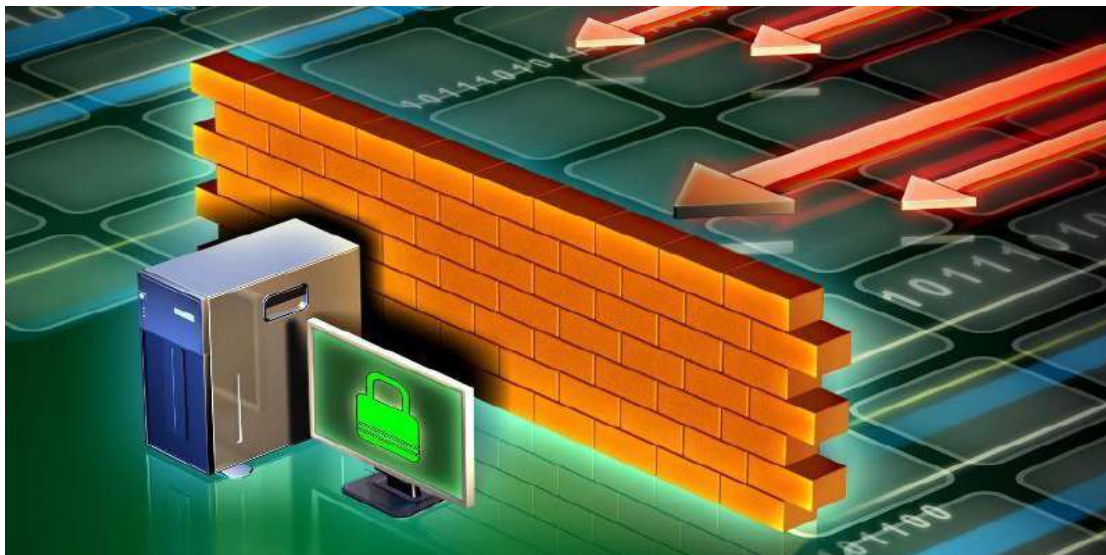
Fuente:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fes.safetydetectives.com%2F&psig=AOvVaw0v70wB_oMMRbvhe5VcpDVw&ust=1675824598013000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCMCK9qu6_g_v0CFQAAAAAdAAAAABAD

3. Firewall o cortafuegos

El firewall es el encargado de inspeccionar la intranet para identificar el tráfico de internet, reconocer a los usuarios y restringir el acceso a aquellos no autorizados. Si bien es una tecnología bastante antigua, las nuevas versiones de estos servicios funcionan perfectamente.

La información de una empresa es valiosa y puede ser utilizada de muchas maneras, por eso suele ser uno de los blancos para atacar. Aunque un profesional pueda engañar a esta defensa, el cortafuegos sigue siendo importante para cualquier usuario.



Firewall DNS

Fuente: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fdinahosting.com%2Fblog%2Fque-es-y-para-que-sirve-firewall-dns%2F&psig=AOvVaw0OJzdKyxz8seV98TTuWRHt&ust=1675825019622000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCICP1KG6gv0CFQAAAAAdAAAAABAD>

4. Plan de seguridad informática

Generar un plan de seguridad informática es imprescindible para cualquier organización, ya que les permitirá a las empresas detectar vulnerabilidades en sus sistemas y establecer medidas para prevenir ataques. El objetivo es que ayude a proteger los datos y sistemas críticos del negocio.



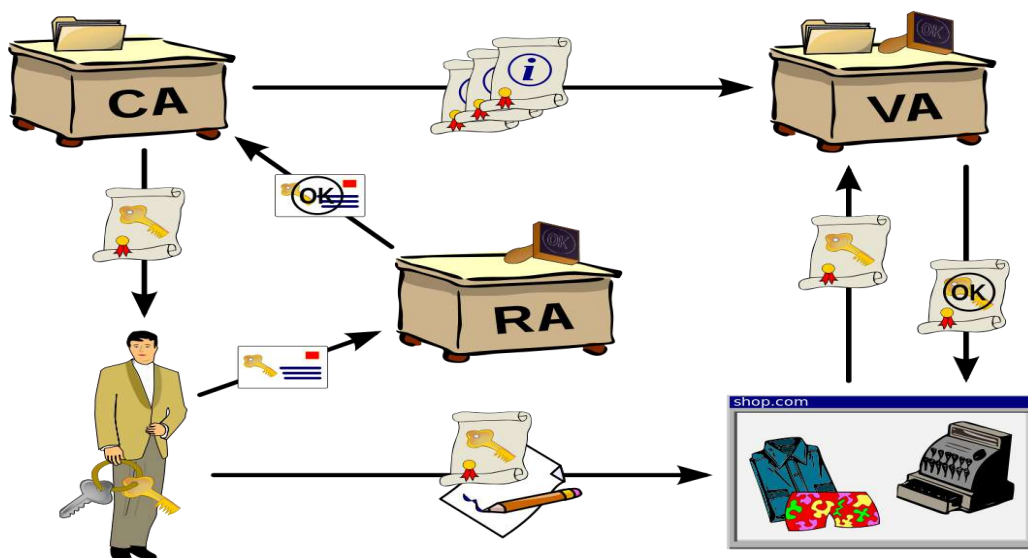
Firewall Plan de seguridad informática para tu empresa

Fuente: https://www.google.com/url?sa=i&url=https%3A%2F%2Fconexure.es%2Fplan-de-seguridad-informatica%2F&psig=AOvVaw1mQUpW-q_R6ZU62arvkEng&ust=1675825251738000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCMjrwP-5gv0CFQAAAAAdAAAAABAD

5. Infraestructura de clave pública o PKI

La PKI, o infraestructura de clave pública, es la herramienta que permite el intercambio de información entre usuarios de una manera segura. Gracias a esta tecnología, se pueden distribuir e identificar las claves de cifrado público.

Esto te permitirá saber con certeza quién es el emisor del mensaje, con todo lo que eso representa para la seguridad de datos de una empresa.



Infraestructura de clave pública

Fuente: <https://upload.wikimedia.org/wikipedia/commons/thumb/3/34/Public-Key-Infrastructure.svg/1200px-Public-Key-Infrastructure.svg.png>

6. Pentesting

El pentesting, o testeo de penetración, consiste en realizar ataques a los sistemas de defensa que posees. El proceso consiste en formar dos grupos y probar las defensas de la empresa en cada uno. Básicamente, es una auditoría de seguridad informática que pone a prueba tu protección.



Pentesting

Fuente:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FInfraestructura_de_clave_p%25C3%25BAblica&psig=AOvVaw2YS6uJN9ojOS1Kom8WkuY&ust=1675825473006000&source=images&cd=vfe&ved=0CBAQjRxaFwoTCMidsry4gv0CFQAAAAAdAAAAABAI

7. Personal capacitado

Por último, pero no menos importante, en la empresa debe existir personal capacitado para poner en marcha estos procesos, así como para mantenerlos funcionando. De nada sirve un sistema de seguridad informático si nadie lo opera.



Personal Capacitada

Fuente:

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.gob.mx%2Fprofedet%2Fes%2Farticulos%2Fla-importancia-de-la-capacitacion-para-las-y-los-trabajadores%3Fidiom%3Des&psig=AOvVaw0T-Ehdaq4B-le8KldQLhUw&ust=1675826158091000&source=images&cd=vfe&ved=0CBAQjRxaFwoTCIDOk4m5qv0CFQAAAAAdAAAAABAE>

Beneficios de implementar y mantener prácticas de seguridad informática

Existen múltiples aspectos que demuestran que la ciberseguridad y las buenas prácticas de seguridad informática son fundamentales en cualquier industria. Entre sus principales beneficios están los siguientes:

- Adaptarse a las necesidades y paradigmas del mercado.
- Protegerse de software maliciosos.
- Mejorar la toma de decisiones y facilitar la obtención de resultados comerciales y estructurales.

- Alcanzar los más altos estándares de calidad relacionados con la protección de datos personales y la seguridad de bases de datos.
- Satisfacer las necesidades de los colaboradores al proporcionar ambientes laborales seguros.
- Proteger la productividad del negocio.
- Generar mayor confianza dentro del mercado y con los clientes, facilitando el relacionamiento y la autoridad de la marca en su giro.

1.3.6 Procesos de seguridad Industrial

Como es evidente, la mejor manera para aplicar una Seguridad Industrial buena, es elaborando un plan de gestión de riesgos que sea eficaz, de manera que se eviten situaciones de peligro. Por eso es importante, prevenir cualquier riesgo y para eso sirve la Seguridad Industrial.

Por estos motivos, A continuación algunos aspectos para asegurar la mejora de la Seguridad Industrial en una organización.

Propósitos y Objetivos de un Sistema de Seguridad Física

- Asegurar la capacidad de supervivencia de la organización ante eventos que pongan en peligro su existencia.
- Proteger y conservar los activos de la organización, de riesgos, de desastres naturales o actos mal intencionados.
- Reducir la probabilidad de las pérdidas, a un mínimo nivel aceptable, a un costo razonable y asegurar la adecuada recuperación.
- Asegurar que existan controles adecuados para las condiciones ambientales que reduzcan el riesgo por fallas o mal funcionamiento del equipo, del software, de los datos y de los medios de almacenamiento.

- Controlar el acceso, de agentes de riesgo, a la organización para minimizar la vulnerabilidad potencial.

Factores que afectan la seguridad física

Los riesgos ambientales a los que está expuesta la organización son tan diferentes según sean las personas, las situaciones y los entornos. El tipo de medidas de seguridad que se pueden tomar contra factores ambientales dependerá de las medidas de tecnología considerada y de dónde serán utilizadas.

En este punto solo se mencionan los factores que afectan a la seguridad física de una organización.

Factores ambientales

Incendios: Los incendios son causados por el uso inadecuado de combustibles, fallas de instalaciones inalámbricas defectuosas y el inadecuado almacenamiento y traslado de sustancias peligrosas.



Incendio en taller

Fuente:

https://www.google.com/url?sa=i&url=http%3A%2F%2Fperiodicocontexto.blogspot.com%2F2016%2F01%2Finendio-ocasiona-perdida-total-de-2.html&psig=AOvVaw03BzkPzAJXZLuLNOXNIUI5&ust=1675828414454000&source=images&cd=vfe&ved=0CB_AQjRxqFwoTCJj5mL3Bgv0CFQAAAAAdAAAAABAE

Inundaciones. Es la invasión de agua por acumulación en terrenos planos, ocasionada por falta de drenaje ya sea natural o artificial. Esta es una de las causas de mayores desastres en centros de cómputo.

Sismos. Estos fenómenos sísmicos pueden ser tan poco intensos que solamente instrumentos muy sensibles los detectan, o tan intensos que causan la destrucción de edificios y hasta la pérdida de vidas humanas.

Humedad. Se debe proveer de un sistema de calefacción, ventilación y aire acondicionado separado, que se dedique al cuarto de computadoras y al área de máquinas.

Factores humanos

Robos. Las computadoras son posesiones valiosas de las empresas y pueden ser expuestas a algún robo.

Actos vandálicos. En las empresas existen empleados descontentos que pueden tomar represalias contra los equipos y las instalaciones. Actos vandálicos contra el sistema de red. Muchos de estos actos van relacionados con el sabotaje.

Fraude. Cada año millones de dólares son sustraídos de empresas y, en muchas ocasiones las computadoras han sido utilizadas para dichos fines.

Sabotaje. Es el peligro más temido en los centros de cómputo. Empresas que han intentado implementar sistemas de seguridad de alto nivel, han encontrado que la protección contra el sabotaje es uno de los ítems más difíciles, el sabotaje puede ser un empleado o un sujeto ajeno a la empresa.

Terroismo. Por la situación que enfrenta el mundo, las empresas deben incrementar sus medidas de seguridad, porque las empresas grandes en el mundo son un blanco para los terroristas.



Prevenir robo de equipos

Fuente:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.antena3.com%2Fnoticias%2Fsociedad%2Fcinc-o-menores-roban-ordenadores-portatiles-colegio-palma-venden-euros_2016111958304f600cf244336f0b0cf1.html&psig=AOvVaw2fKD0vdMI115r1b03Wu8dx&ust=1675829051674000&source=images&cd=vfe&ved=0CBAQjRxaFwoTCJi9h-3Dgv0CFQAAAAAdAAAAABAJ

PLAN DE CONTINGENCIA

Un plan de contingencia trata de un conjunto de procedimientos de recuperación para casos de desastre; es un plan formal que describe pasos apropiados que se deben seguir

en caso de un desastre o emergencia.



El Plan de Contingencia contempla tres tipos de acciones las cuales son:

- **Prevención.** Conjunto de acciones a realizar para prevenir cualquier contingencia que afecte la continuidad operativa
- **Detección.** Deben contener el daño en el momento, así como limitarlo tanto como sea posible, contemplando todos los desastres naturales y eventos no considerados.
- **Recuperación.** Abarcan el mantenimiento de partes críticas entre la pérdida del servicio y los recursos, así como su recuperación o restauración.

CONTROLES AMBIENTALES

Control de Perímetro

El primer paso consiste en establecer en términos generales si se trata de una instalación de riesgo alto, medio o bajo.

Clasificación de las instalaciones

- **Instalaciones de alto riesgo:** Datos o programas que contienen información confidencial de interés nacional o que poseen un valor competitivo alto en el mercado.
 - Pérdida financiera potencial considerable para la comunidad a causa de un desastre o de un gran impacto sobre los miembros del público.
 - Pérdida potencial considerable para la institución.
- **Instalación de riesgo medio:** Son aquellas con aplicaciones cuya interrupción causas grandes inconvenientes y posiblemente el incremento de los costos; sin embargo, se obtiene poca pérdida material.
- **Instalación de bajo riesgo:** Son aquellas con aplicaciones cuyo procesamiento tiene poco impacto material en la institución en términos de costo o de reposición del servicio afectado.



Instalación de alto riesgo

Fuente: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.ceupe.com%2Fblog%2Ftodo-lo-que-se-debe-saber-virus-informaticos.html&psig=AOvVaw0jlbmtmeXDnlJqTaxbeMa&ust=1675829652916000&source=images&cd=vfe&ved=0CBAQjRxaFwoTCIDb5lvGqv0CFQAAAAAdAAAAABAE>

ANÁLISIS DE SISTEMAS BIMESTRE II



2. Ciclo de Vida del Software

Por: Rodrigo Ramírez.

SDLC (Systems Development Life Cycle), es la estructura de procesos, actividades y tareas relacionadas con el desarrollo y mantenimiento de un producto software, abarcando la vida completa del sistema.

2.1 Fases de SDLC

Comunicación, planificación y análisis, estudio de viabilidad, análisis del sistema, diseño, codificación, integración, pruebas, implementación, formación, mantenimiento y funcionamiento.

2.1.1 Comunicación

El cliente solicita un producto de software, plasma sus necesidades y presenta la solicitud para el desarrollo del software.

2.1.2 Planificación y análisis

Aquí se fijan en los requisitos que piden los clientes, estudiar y analizar si no son concretos o son contradictorios.

2.1.3 Estudio de viabilidad

Luego de fijar los requisitos, se idea un plan para procesar el software. Se investiga la viabilidad financiera y tecnológica. Para ellos se utilizan algoritmos para saber si el proyecto de software es factible.

2.1.4 Análisis del sistema



Se asigna recursos y planifica el tiempo de duración del proyecto. Se buscan limitaciones del producto y se identifican los impactos del proyecto sobre toda la organización en su conjunto.



2.1.5 Diseño

Se hace un diseño lógico y otro físico. Para esto se crean metadatos, diagramas o pseudocódigos.

2.1.6 Codificación

Se denomina fase de "programación/desarrollo" es en la que se elige el lenguaje de programación más conveniente, se desarrollan programas ejecutables y sin errores, al final de esta fase se puede obtener PMV(Producto mínimo viable).

2.1.7 Integración

Se integran bibliotecas, bases de datos u otros programas que necesite.

2.1.8 Pruebas

Se prueba repetidamente, tantas veces sea necesario para ver que tenga una funcionalidad de 100%.





Imagen: procesos del ciclo de vida del software

Fuente: <https://www.sergestec.com/que-es-el-ciclo-de-vida-de-desarrollo-de-sistemas-software/>

2.1.9 Implementación

Se instala el software, se evalúa la integración, la adaptabilidad, la portabilidad y se instalan las configuración posteriores necesarias.

2.1.10 Formación

Es la adaptación del usuario, por medio de una capacitación inicial para cada usuario. Comprobar el nivel de uso, la experiencia del usuario y resolver cualquier tipo de dudas.

2.1.11 Mantenimiento y Funcionamiento

En esta última fase se minimizan pequeños errores, se confirma el buen funcionamiento del software, su eficiencia y estabilidad. Si es necesario se dan nuevas formaciones, se adaptan nuevos entornos y se actualiza el código.

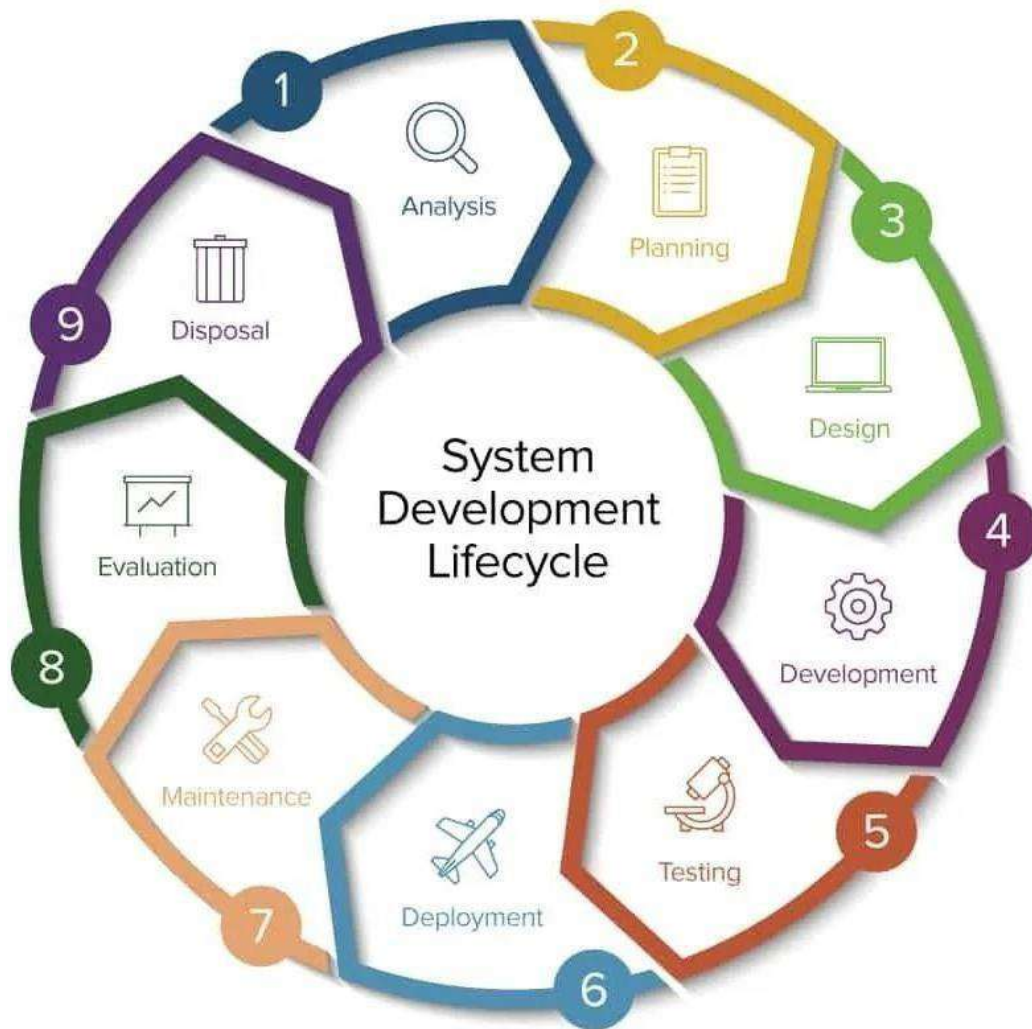


Imagen: sistema de desarrollo del ciclo de vida del software

Fuente: <https://www.sergestec.com/que-es-el-ciclo-de-vida-de-desarrollo-de-sistemas-software/>

2.2 UML

Por: Jose Gabriel Ramos Perez

Lenguaje unificado de modelado, creado para tener arquitectura, Su propósito es proporcionar una forma estándar de visualizar el diseño del sistema. Es una forma generalde describir cómo funciona la estructura de un sistema, cómo funciona, cómo se conectan los objetos y las relaciones entre los objetos. El Lenguaje de modelado unificado se utiliza en software desarrollo.

2.2.1.1Tipos de modelos

Tenemos 2 tipos de modelos en uml, diagrama de estructura, diagrama de comportamiento, Estas variantes existen para representar diferentes tipos de escenas y diagramas utilizados por diferentes tipos de usuarios. Cada rol utilizará un gráfico específico para satisfacer sus necesidades. El objetivo es que el UML exprese visualmente diagramas que sean fáciles deentender para todos.

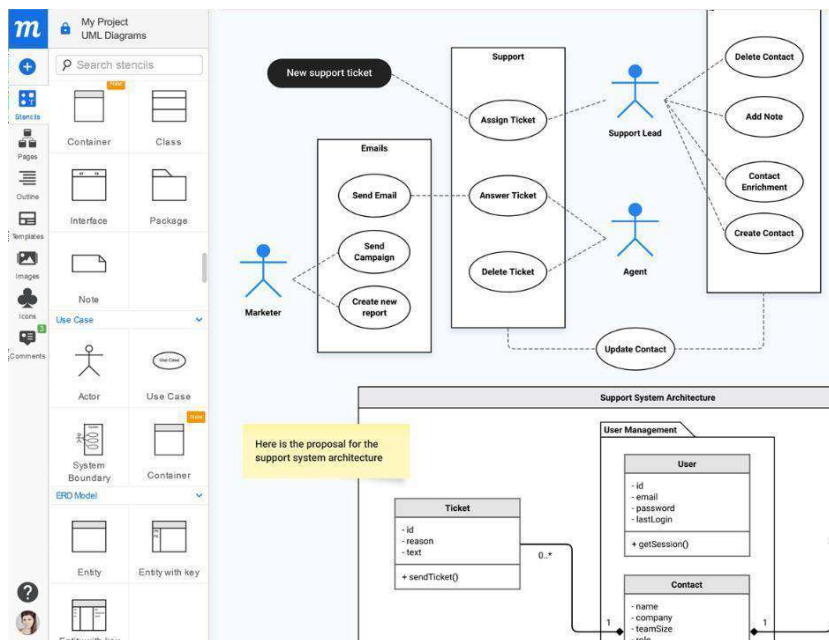
2.2.1.2Diagrama estructural

Un diagrama de estructura representa la estructura estática de un software o sistema y también muestra diferentes niveles de abstracción e implementación. Se utilizan para ayudar a visualizar las diversas estructuras que componen su sistema, como bases de datos o aplicaciones. Muestran cómo se conectan e interactúan entre sí los diferentes elementos.

2.2.1.3Diagrama de comportamiento

Estos diagramas UML visualizan cómo funciona un sistema y cómo interactúa dentro de sí mismo y con los usuarios, otros sistemas y otras entidades. se emplean para visualizar, especificar, construir y documentar los aspectos dinámicos de un sistema. Representa la secuencia de estados por los que pasa un objeto durante su vida útil en respuesta a eventos. Estos diagramas muestran los diferentes estados del proceso.





<https://moqups.com/img/mqlps/images/uml-diagram-tool/hero-image.png>

2.2.1 Caso de Uso

Cuando crea modelos de base de datos o diagramas UML con unas herramientas de software, el proceso se simplifica y mejora. UML también es popular como guía de modelado de bases de datos. Estas maquetas son excelentes herramientas visuales para la lluvia de ideas, la creación de diagramas de formalibre y la colaboración



creativa.

Imagen: UML

<https://www.microsoft.com/es-ww/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling#:~:text=Existen%20dos%20tipos%20principales%20de,los%20diferentes%20tipos%20de%20personas.>



2.2.2 Diagrama de proceso

Los diagramas de proceso UML ayudan a visualizar el caso de uso con más detalle. Los diagramas de proceso UML también se pueden usar para representar el flujo de eventos en un proceso comercial. Es básicamente un diagrama de flujo que muestralas operaciones realizadas por el sistema. Documentar el proceso para mejorar la comprensión.

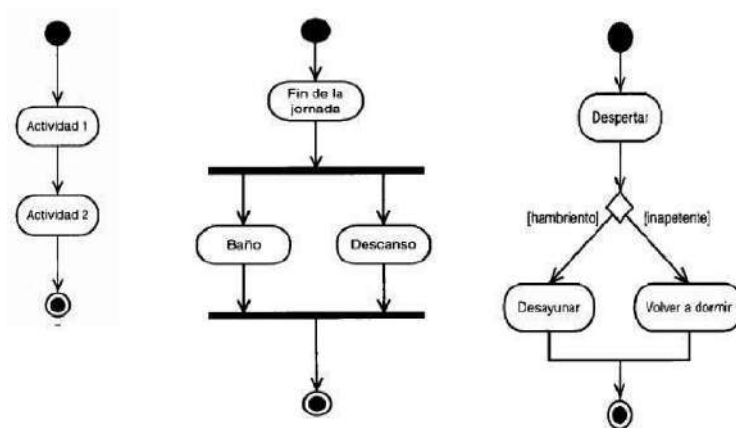


Imagen: Diagrama de proceso UML

Fuente :

<https://lh3.googleusercontent.com/HnPMRnBgYKgLiITKVzdlIt3MgUzMY9kcu3p09RTx98IfVJxstO26pEdld1KeZaPUWvOo7DrNH23DzjJqnM5YZQ1RLjCOfuUsPYy35WNZQ>

2.2.3 Diagrama de Colaboracion

Por: Fernando Jose Raxon Pirir

Los diagramas de colaboración también se conocen como diagramas de comunicación. Pueden demostrar cómo se comunican los objetos para ejecutar las acciones específicas o un aspecto de un caso de uso. Los diseñadores pueden usar diagramas de colaboración para explicar e identificar los roles de los objetos que realizan un flujo específico de eventos en un caso de uso. Son la principal fuente de información utilizada para establecer roles e interfaces de clase.

Al interactuar juntos, los objetos pueden generar funcionalidades (utilizables) de alto nivel. Los objetos colaboran y operan juntos comunicándose (pasando mensajes) entre sí. El diagrama de colaboración representa las relaciones entre los objetos. El diagrama de colaboración ilustra cómo se intercambian mensajes entre clases y objetos (instancias). Se genera para cada proceso del sistema y se relaciona con la fase de desarrollo existente.

2.2.4 Diagrama de Clases

El diagrama de clases es uno de los diagramas incluidos en UML 2.5 clasificado dentro de los diagramas de estructura y, como tal, se utiliza para representar los elementos que componen un sistema de información desde un punto de vista estático.

Es importante destacar que, por esta misma razón, este diagrama no incluye la forma en la que se comportan a lo largo de la ejecución los distintos elementos, esa función puede ser representada a través de un diagrama de comportamiento, como por ejemplo un diagrama de secuencia o un diagrama de casos de uso.

2.2.5 Diagrama de componentes

El diagrama de componentes es uno de los principales diagramas UML. Está clasificado como diagrama de estructura y, como tal, representa de forma estática el sistema de información. Habitualmente se utiliza después de haber creado el diagrama de clases, pues necesita información de este diagrama como pueden ser las propias clases.

Este diagrama proporciona una vista de alto nivel de los componentes dentro de un sistema. Los componentes pueden ser un componente de *software*, como una base de datos o una interfaz de usuario; o un componente de *hardware* como un circuito, microchip o dispositivo; o una unidad de negocio como un proveedor, nómina o envío.

Algunos usos de este tipo de diagrama es el siguiente:

- 2.2.5.1 Se utilizan en desarrollo basado en componentes para describir sistemas con arquitectura orientada a servicios.
- 2.2.5.2 Mostrar la estructura del propio código.
- 2.2.5.3 Se puede utilizar para centrarse en la relación entre los componentes mientras se ocultan los detalles de las especificaciones.
- 2.2.5.4 Ayudar a comunicar y explicar las funciones del sistema que se está construyendo a los interesados o stakeholders.

DIAGRAMA DE COMPONENTES
FACTURACIÓN Y PAGOS
 Benítez Rodríguez Jessica

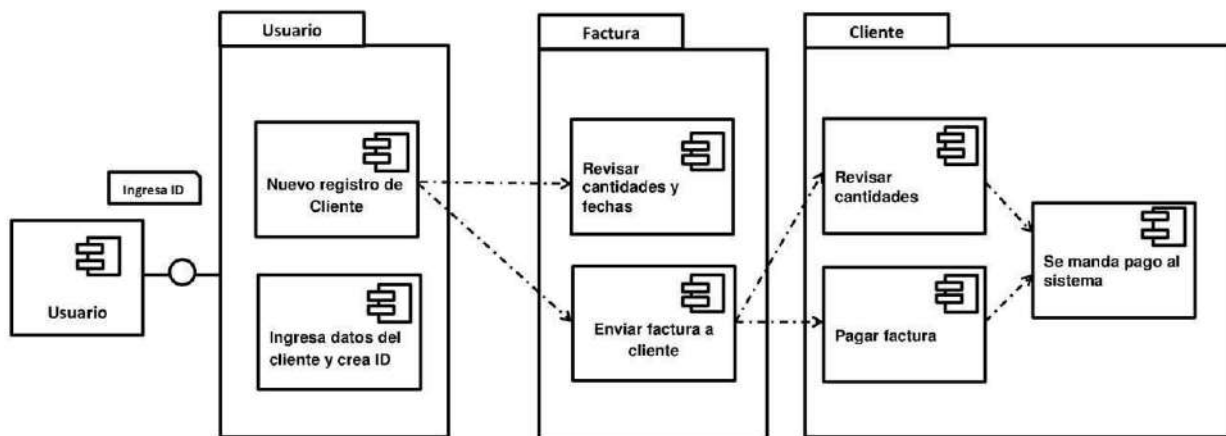










Imagen: Diagrama de componentes

Fuente: [diagrama de componentes - Bing images](#)

2.2.6 Vistas

Un diagrama de componentes proporciona una visión general del sistema y documenta la organización de los componentes del sistema y sus relaciones y dependencias mutuas. Los diagramas de componentes proporcionan una visión orientada a la ejecución, es decir, dan al desarrollador información sobre si un sistema funciona de forma coherente y cumple sus tareas y objetivos.

Los objetivos y propósitos más importantes de este tipo de diagrama son el modelado de sistemas de software basados en componentes, la especificación de arquitecturas de software y la división de sistemas en subsistemas (por ejemplo, interfaz gráfica de usuario/IGU, ámbito empresarial y capa de persistencia con base de datos relacional). Asimismo, se asignan tareas y funciones concretas a las subáreas y sus interfaces dentro del sistema.

Elemento	Símbolo/notación	Explicación
Componente (<i>component</i>)		Símbolo para representar los módulos de un sistema (la interacción y la comunicación tienen lugar a través de interfaces).
Paquete		Un paquete combina varios elementos del sistema (por ejemplo, clases, componentes o interfaces) en un grupo.
Artefacto		Los artefactos son unidades físicas de información (por ejemplo, código fuente, archivos .exe, scripts o documentos) que se generan en el proceso de desarrollo o el tiempo de ejecución de un sistema o son necesarios para estos.
Interfaz ofrecida		Símbolo para una o más interfaces claramente definidas que proporcionan funciones, servicios o datos al mundo exterior (el semicírculo abierto también se denomina enchufe o <i>socket</i>).
Interfaz requerida		Símbolo de una interfaz necesaria para recibir funciones, servicios o datos del exterior (la notación del círculo con palo también se denomina <i>lollipop</i> o <i>piruleta</i>).
Puerto		Este símbolo indica un punto de interacción independiente entre un componente y su entorno.
Relación		Las líneas actúan como conectores e indican las relaciones entre los componentes.
Relación de dependencia		Conector especial para expresar una relación de dependencia entre los componentes del sistema (no siempre se indica).

Fuente: [Diagrama de componentes UML: explicación, creación y ejemplo - IONOS](#)

2.2.5 Diagrama De Componentes

Por: **Gabriel Rivas**

Se utilizan para modelar los componentes que ayudan a hacer esas funcionalidades, representando la forma en la que estos se organizan y sus dependencias. También se trata de los principales diagramas UML y se califica como un diagrama de estructura. Nos hace ver los componentes del sistema, como por ejemplo un diagrama basado en software para una base de datos o una interfaz. También para hardware para un dispositivo, un circuito o un microchip.

2.2.6 Objetos

Componente: Este representa el sistema de módulos sobre la interpretación de la situación del sistema sobre el que trabajaremos.

Paquete: Combinación de los elementos del sistema como la interfaz o componentes del grupo.

Artefacto: Es la unidad física de la información como archivos, código fuente, scripts, documentos o .exe. Es el que genera proceso del desarrollo sobre el tema de la ejecución del sistema.

Interfaz Ofrecida: Este símbolo representa la definición de la función, servicio o datos del exterior.

Interfaz Requerida: Este símbolo recibe funciones, servicios o datos del exterior.

Puerto: Este símbolo indica el punto de relación independiente entre un componente y el ambiente de su entorno.

Relación: Son unas líneas que acuan como una conexión entre las relaciones y los componentes.

Relación de dependencia: Es un símbolo especial que indican expresar una relación de dependencia entre los componentes y del sistema.



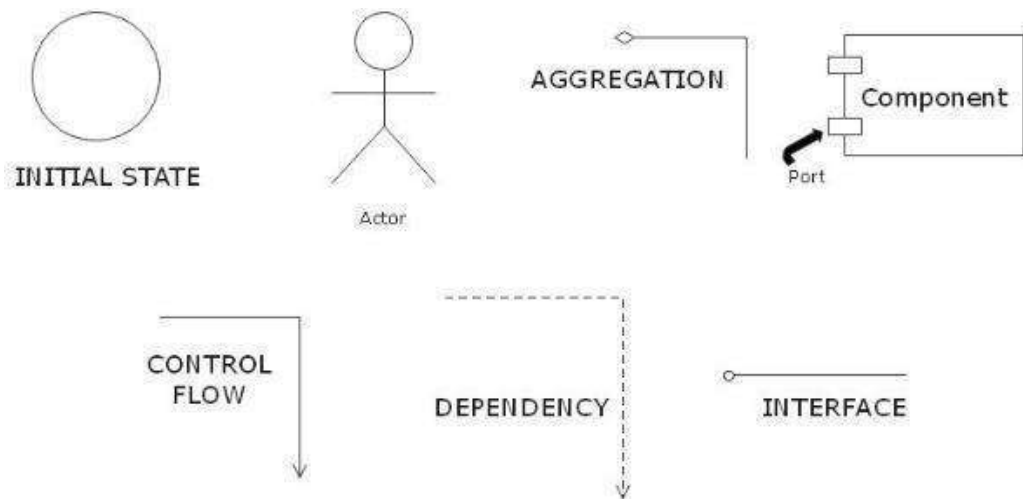


imagen: objetos del diagrama de componentes

2.2.7 Metodos

Diagrama de clase Uml: Nos ofrece una vista desde arriba del sistema en software en una formade desarrollo de software en nivel alto. El diagrama de componentes UML ofrece comporennder elcomportamiento exacto del servicio que proporciona cada pieza de su software le convierta en un mejor desarrollo. El diagrama nos describe el sistema de software implementados en cualquier lenguaje.

2.2.7 Atributos

- Se utilizan en el desarrollo basado en componentes para describir sistemas conarquitectura orientada a servicios
- Mostrar la estructura del propio código
- Se puede utilizar para centrarse en la relación entre los componentes mientrasse ocultan los detalles de las especificaciones
- Ayudar a comunicar y explicar a los interesados las funciones del sistema que seestá construyendo

2.3 Normalizar

Es el proceso de simplificación de los datos ingresados en una base de datos, eso nosayuda sobre el almacenar mayor cantidad de datos con menor espacio, Eliminar redundancia y ordenar los datos más relevantes

MATRICULA	NOMBRE	APELLIDO	No. CARRERA
1	DANIELA	BOHORQUEZ	1234
2	JUAN	HERNANDEZ	1234

No. CARRERA	CARRERA
1234	TINF
6789	CIVIL

MATRICULA	NRG
1	123
1	234
1	237
2	123

ASIGNATURA	NRC
BASE DATOS	123
PROG WEB	234
CALCULO	237

Imagen: Base de datos mostrando normalización

Fuente: <https://www.youtube.com/watch?v=MdYvCMbQji4>

2.3 Casos de Normalización

Por: Ariel Rodriguez

La normalización es un proceso que se aplica a las bases de datos para organizar las tablas y evitar la redundancia o complejidad en los datos. En este sencillo y muy explicativo artículo resolvemos tus dudas sobre la normalización y te explicamos para qué sirve, pues la verdad es más sencilla de lo que parece.

La normalización de una base de datos es la aplicación de una serie de reglas para evitar en el futuro realizar consultas innecesariamente complejas. En otras palabras, están enfocadas en eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas que creamos para organizar las bases de datos.

1.1.1 Características

Evitar la redundancia de datos Proteger

la integridad de los datos

Evitar problemas de actualización de los datos en las tablas

2.3.1 1NF (Primera forma normal)

Por definición, una entidad que no tiene columnas repetidas o grupos de datos puede denominarse Primera forma normal. En la primera forma normal, cada columna es única.

A continuación, se muestra cómo se vería nuestra tabla Empleados y Departamento si estuviera en la primera forma normal (1NF):

empNum	apellido	nombre de pila	deptName	deptCity	deptCountry
1001	Andrews	Jacobo	Cuentas	Nueva York	Estados Unidos
1002	Schwartz	Miguel	Tecnología	Nueva York	Estados Unidos
1009	taza	Harry	HORA	Berlina	Alemania
1007	Harvey	Parker	Administración	Londres	Reino Unido
1007	Harvey	Parker	HORA	Londres	Reino Unido

Imagen: Ejemplo del caso 1NF

Fuente: <https://spa.myservername.com/database-normalization-tutorial>



Aquí, todas las columnas de las tablas de Empleados y Departamentos se han agrupado en una y no hay necesidad de conectar columnas, como deptNum, ya que todos los datos están disponibles en un solo lugar.

Pero una tabla como esta con todas las columnas requeridas en ella, no solo sería difícil de administrar sino también difícil de realizar operaciones y también ineficiente desde el punto de vista del almacenamiento.

2.3.2 2NF (Segunda forma normal)

Por definición, una entidad que es 1NF y uno de sus atributos se define como la clave principal y los atributos restantes dependen de la clave principal.

A continuación, se muestra un ejemplo de cómo se vería la tabla de empleados y departamentos:

Tabla de empleados

empNum	apellido	nombre de pila
1001	Andrews	Jacobo
1002	Schwatz	Miguel
1009	taza	Harry
1007	Harvey	Parker
1007	Harvey	Parker

Imagen: Ejemplo caso 2NF

Fuente: <https://spa.myservername.com/database-normalization-tutorial>

Mesa de Departamentos:

deptNum	deptName	deptCity	deptCountry
1	Cuentas	Nueva York	Estados Unidos
2	Tecnología	Nueva York	Estados Unidos
3	HORA	Berlina	Alemania
4	Administración	Londres	Reino Unido

Imagen: Ejemplo caso 2NF

Fuente: <https://spa.myservername.com/database-normalization-tutorial>



Tabla EmpDept:

empDeptID	empNum	deptNum
1	1001	1
2	1002	2
3	1009	3
4	1007	4
5	1007	3

Imagen: Ejemplo

Fuente: <https://spa.myservername.com/database-normalization-tutorial>

Aquí, podemos observar que hemos dividido la tabla en forma 1NF en tres tablas diferentes. La tabla Empleados es una entidad sobre todos los empleados de una empresa y sus atributos describen las propiedades de cada empleado. La clave principal de esta tabla es empNum.

De manera similar, la tabla Departamentos es una entidad sobre todos los departamentos de una empresa y sus atributos describen las propiedades de cada departamento. La clave principal de esta tabla es deptNum.

En la tercera tabla, hemos combinado las claves primarias de ambas tablas. Las claves principales de las tablas Empleados y Departamentos se denominan claves externas en esta tercera tabla.

Si el usuario quiere una salida similar a la que teníamos en 1NF, entonces el usuario tiene que unir las tres tablas, usando las claves primarias.

Una consulta de muestra se vería como se muestra a continuación:

|_+_|

2.3.3 3NF (tercera forma normal)

Por definición, una tabla se considera en tercera normalidad si la tabla / entidad ya está en la segunda forma normal y las columnas de la tabla / entidad no dependen de forma transitiva de la clave primaria.

Comprendamos la dependencia no transitiva con la ayuda del siguiente ejemplo.

Digamos que una tabla llamada Cliente tiene las siguientes columnas: Identificación del cliente - Clave principal que identifica a un cliente único ClienteZIP - Código postal de la localidad en la que reside el cliente

CustomerCity - Ciudad en la que reside el cliente

En el caso anterior, la columna CustomerCity depende de la columna CustomerZIP y la columna CustomerZIP depende de CustomerID.

El escenario anterior se denomina dependencia transitiva de la columna CustomerCity en CustomerID, es decir, la clave principal. Después de comprender la dependencia transitiva, analicemos ahora el problema de esta dependencia.

Podría haber un escenario posible en el que se realice una actualización no deseada

en la tabla para actualizar CustomerZIP a un código postal de una ciudad diferente sin actualizar CustomerCity, dejando así la base de datos en un estado inconsistente.



Para solucionar este problema, necesitamos eliminar la dependencia transitiva que podría hacerse creando otra tabla, por ejemplo, la tabla CustZIP que contiene dos columnas, es decir, CustomerZIP (como clave principal) y CustomerCity.

La columna CustomerZIP en la tabla Customer es una clave externa para CustomerZIP en la tabla CustZIP. Esta relación garantiza que no haya anomalías en las actualizaciones en las que se actualiza un CustomerZIP sin realizar cambios en CustomerCity.



ANALISIS BIMESTRE 3

3. Administrador de base de datos.

Por: Braulio José Alexander Guarca Lima

Antes de anticiparnos a conocer el concepto a profundidad, es importante que comprendamos primero qué es un administrador en todo el sentido de la palabra. El administrador es aquella persona que tiene la capacidad de poner en práctica sus conocimientos adquiridos en el área de planeación, organización, dirección y control empresarial.

3.1 Definiciones y conceptos de base de datos

Una base de datos es una herramienta para recopilar y organizar información. Las bases de datos pueden almacenar información sobre personas, productos, pedidos u otras cosas.

Las bases de datos son el producto de la necesidad humana de almacenar la información, es decir, de preservarla contra el tiempo y el deterioro, para poder acudir a ella posteriormente. En ese sentido, la aparición de la electrónica y la computación brindó el elemento digital indispensable para almacenar enormes cantidades de datos en espacios físicos limitados.

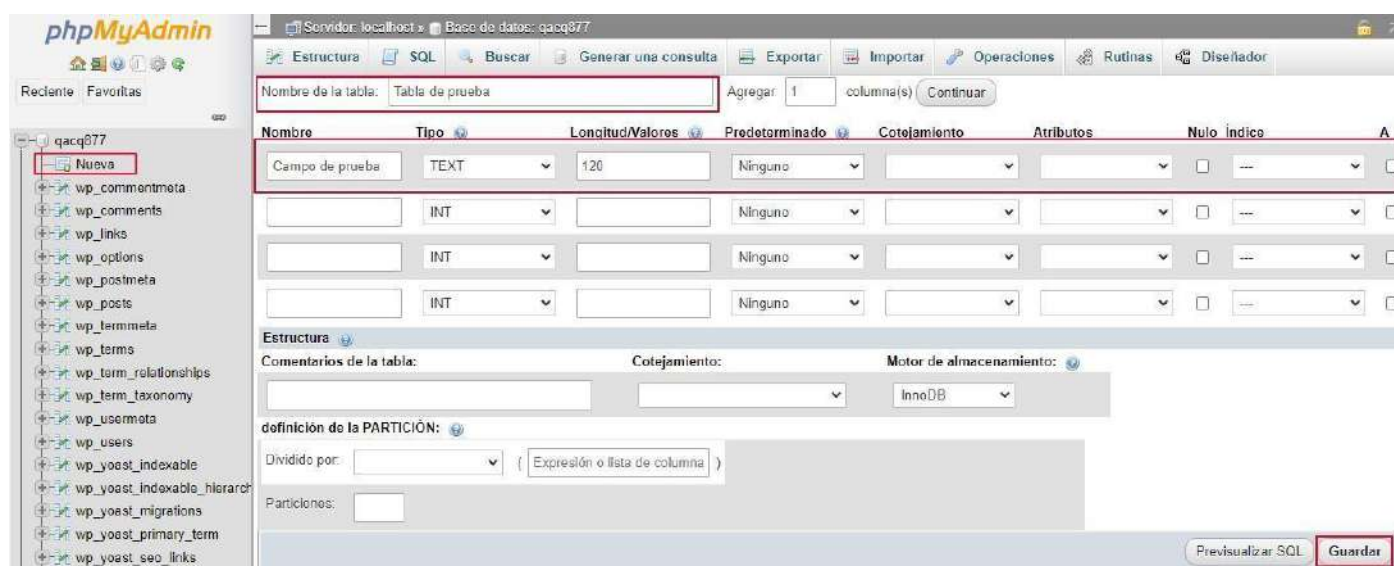


Imagen: Interfaz de phpmyadmin para la creación de una base de datos

Fuente: <https://www.arsys.es/blog/phpmyadmin-basededatos-wordpress>

3.2 Arquitectura de bases de datos: relacionales y no relacionales

Las **bases de datos relacionales** son la arquitectura más utilizada hoy en día. Esta consiste en un conjunto de tablas formadas por filas (registros) y columnas (campos), tal como una hoja de cálculo.

Una de las principales características de la base de datos relacional es que evitar la duplicidad de registros y a su vez garantizar la integridad referencial, es decir, que si se elimina uno de los registros, la integridad de los registros restantes no será afectada. Además, gracias a las claves se puede acceder de forma sencilla a la información y recuperarla en cualquier momento.

Empleados							
ID_e	1º Apellido	2º Apellido	Nombre	Nº SS	Calle	CP	Municipio
1							
2							
3							
4							

Imagen: Ejemplo de base de datos relacional

Fuente: <https://ayudaleyprotecciondatos.es/bases-de-datos/relacional/>

Una **base de datos no relacional** es aquella que no usa el esquema tabular de filas y columnas que se encuentra en la mayoría de los sistemas de base de datos más tradicionales. En su lugar, las bases de datos no relacionales usan un modelo de almacenamiento que está optimizado para los requisitos específicos del tipo de datos que se almacena.

Las principales características de una base de datos no relacional son: la información no se almacena en tablas sino a través de documentos, son bases de datos muy útiles para organizar y gestionar información no estructurada, o cuando no se tiene una noción clara de los datos a almacenar, son bases de datos con alto grado de escalabilidad y están diseñadas para soportar grandes volúmenes de datos, no utilizan el lenguaje SQL para consultas, aunque sí lo pueden usar como herramienta de apoyo.



Imagen: Ejemplo de base de datos no relacional

Fuente: <https://jesgargardon.com/blog/las-bases-de-datos-no-relacionales/>

3.3 Función del administrador de base de datos

Un Administrador de Bases de Datos, se encarga de gestionar y dar mantenimiento a las bases de datos de diferentes sistemas informáticos. Su trabajo consiste en verificar que sean seguras y estén actualizadas. Además de reorganizarlas para hacerlas más rápidas y fáciles de usar.

Los administradores de bases de datos son responsables de las bases de datos informatizadas de las empresas. Las empresas utilizan bases de datos para mantener y actualizar grandes cantidades de información de forma estructurada para poder usarla más eficazmente.

Los administradores de bases de datos se encargan de que la base de datos sea segura y se utilice con eficacia. El uso y la seguridad de la información contenida en bases de datos están regulados por una estricta ley de protección de datos.

3.4 Tipos de datos estructurados

Por: Eddy Guerrero

Los tipos de datos estructurados en programación son estructuras de datos que almacenan y organizan datos de una manera específica, lo que permite un acceso y modificación eficiente de los datos. Algunos ejemplos comunes de tipos de datos estructurados incluyen:

Int: Este tipo de dato se utiliza para almacenar valores numéricos enteros y puede tener valores en un rango específico, dependiendo del sistema y la implementación.

Float: Este tipo de dato se utiliza para almacenar valores numéricos con decimales y es capaz de representar números con una mayor precisión que los enteros. La precisión depende de la implementación y el sistema, pero en general, los números de punto flotante pueden tener decimales de 7 u 8 dígitos. Sin embargo, debido a la forma en que los números de punto flotante se almacenan en la memoria, pueden haber errores de redondeo y pérdida de precisión en operaciones matemáticas.

Char: Este tipo de dato se utiliza para almacenar un solo carácter, que puede ser una letra, un número o un símbolo. En la mayoría de los sistemas, un carácter se representa como un número en el código ASCII o Unicode, y se almacena en un solo byte de memoria. Esto significa que un char puede tener un valor entre 0 y 255, dependiendo de la codificación utilizada. Los caracteres se utilizan a menudo para representar texto y para manejar entradas y salidas de texto en un programa.

Boolean: Este tipo de dato solo puede tener dos valores posibles: "verdadero" o "falso". Los valores booleanos se utilizan a menudo en expresiones lógicas y en estructuras de control de flujo para determinar el curso de acción en un programa.

Arrays: una colección de elementos del mismo tipo de datos almacenados en ubicaciones de memoria cercanas.

Un array es una estructura de datos que almacena una colección de elementos, todos del mismo tipo de datos, en ubicaciones de memoria cercana.

Los elementos están indexados y se puede acceder a ellos y modificarlos por su número de índice.

Los array tienen un tamaño fijo y el número de elementos debe especificarse en el momento de la creación.

Los array se utilizan en muchas aplicaciones para almacenar y manipular grandes cantidades de datos de manera eficiente.

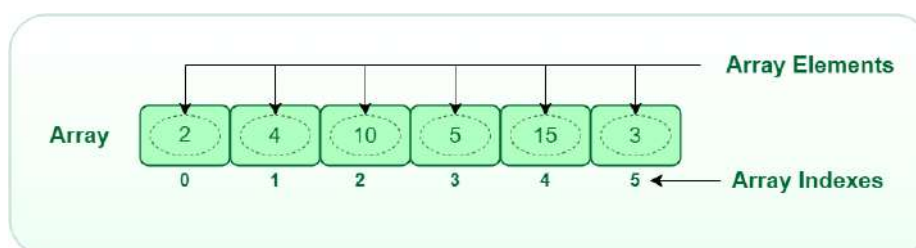


Imagen: Ejemplo de estructura del array

Fuente: <https://media.geeksforgeeks.org/wp-content/uploads/20220721080308/array.png>

Lists (Listas): una colección de elementos del mismo o diferente tipo de datos almacenados en ubicaciones de memoria no cercanas.

Las listas permiten la asignación de memoria dinámica, lo que significa que el tamaño de una lista puede cambiar en tiempo de ejecución y los elementos se pueden agregar o eliminar de la lista según sea necesario.

Las listas también permiten inserciones y eliminaciones de tiempo constante al principio o al final de la lista. Sin embargo, acceder a elementos en un índice específico puede llevar un tiempo lineal, ya que requiere recorrer la lista desde el principio.

Las listas se usan ampliamente en muchas aplicaciones, como el almacenamiento de colecciones de elementos, la implementación de colas o pilas o la representación de gráficos.

Stack (Pila): una colección de elementos con un orden de último en entrar, primero en salir (LIFO).

Un stack se utiliza para almacenar elementos y realizar operaciones de inserción y eliminación en la parte superior del stack (también conocido como "tope" del stack). Esta estructura de datos es útil en una variedad de aplicaciones, incluyendo la implementación de funciones recursivas, la evaluación de expresiones matemáticas y el manejo de la memoria en sistemas operativos.

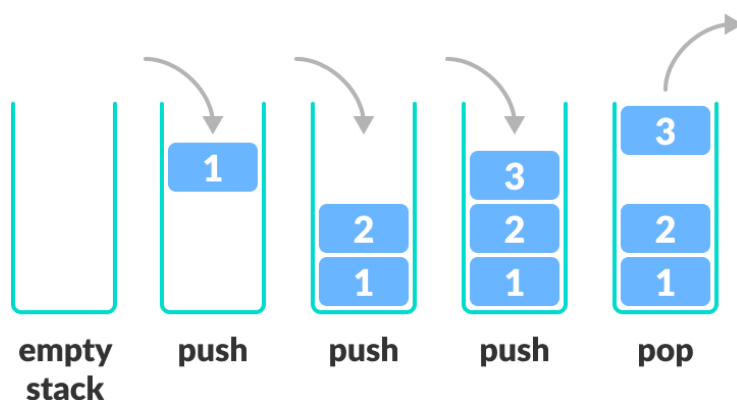


Imagen: Ejemplo de estructura del array

Fuente: <https://cdn.programiz.com/sites/tutorial2program/files/stack.png>

En términos simples, un stack puede ser pensado como una pila de objetos, donde solo se pueden agregar o retirar objetos de la parte superior de la pila.

Queue (Cola): una colección de elementos con un orden de primero en entrar, primero en salir (FIFO).

es una estructura de datos de tipo "first-in, first-out" (FIFO), donde los elementos son insertados al final y se retiran desde el principio. Es decir, el primer elemento en entrar es el primer elemento en salir, similares a las colas en la vida real.

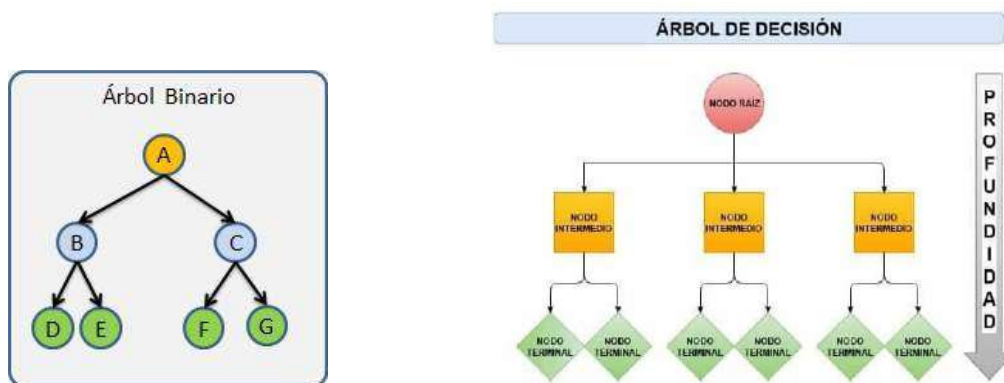


Queue Data Structure

Imagen: Ejemplo de estructura del Queue

Fuente: <https://media.geeksforgeeks.org/wp-content/cdn-uploads/20221213113312/Queue-Data-Structures.png>

Trees (Árboles): Un árbol es una estructura de datos en forma de gráficos recurrentes que se utiliza para almacenar información y realizar búsquedas eficientes. Cada elemento en el árbol se conoce como nodo y puede tener uno o más nodos hijos. El nodo raíz es el nodo principal del árbol y los nodos hijos se conectan a partir de él. Los árboles se utilizan en muchos algoritmos de computación, incluyendo árboles de búsqueda binaria y árboles de decisión.



Imágenes representativas acerca de arboles

Fuente: <http://www.oscarblancarteblog.com/wp-content/uploads/2014/08/arbolorbinario.png>

Fuente: <https://www.maximaformacion.es/wp-content/uploads/2021/07/Estructura-de-un-arbol-de-decision.jpg>

Graphs (Gráficos): una colección de nodos y aristas que representan las relaciones entre los nodos. los gráficos son un tipo de estructura de datos no lineal que no sigue un orden estricto, como una lista o una matriz, sino que permite representar relaciones entre elementos de manera más flexible y adaptable. Esto los hace especialmente útiles para modelar problemas reales que involucran relaciones complejas entre objetos y entidades.

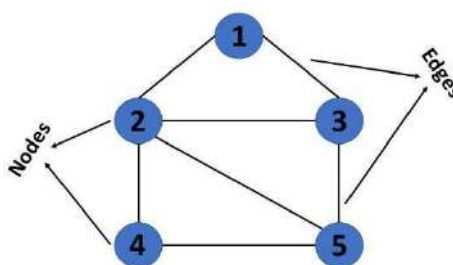


Imagen: representación acerca de los gráficos

Fuente:

https://www.simplilearn.com/ice9/free_resources_article_thumb/Graph%20Data%20Structure%20-%20Soni/what-is-graphs-in-data-structure.png

Hash tables (Tablas hash): una estructura de datos que utiliza una función hash para asignar claves a índices en una matriz.

Las tablas hash son una estructura de datos que se utiliza para implementar un mapa que asocia claves con valores. Son eficientes en términos de tiempo para las operaciones de búsqueda, inserción y eliminación de elementos

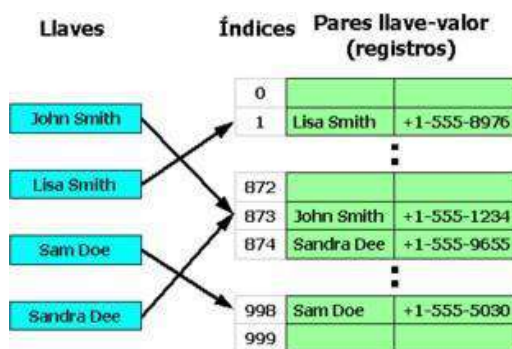


Imagen: representación de tablas hash

Fuente: https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/Tabla_hash_3.png/350px-Tabla_hash_3.png

3.5 Lenguajes estructurados de consultas

Los lenguajes estructurados de consultas (SQL, por sus siglas en inglés) son lenguajes de programación diseñados para gestionar y manipular bases de datos relacionales. Permiten a los usuarios realizar consultas complejas a las bases de datos, seleccionar, agrupar y filtrar información, así como también insertar, actualizar y eliminar datos.

SQL es ampliamente utilizado en aplicaciones empresariales y es un estándar en el desarrollo de bases de datos relacionales. Es compatible con una amplia variedad de sistemas de bases de datos, incluyendo MySQL, PostgreSQL, Microsoft SQL Server, Oracle y muchos otros.

SQL es un lenguaje de alto nivel y fácil de aprender, lo que lo hace accesible tanto para desarrolladores como para usuarios no técnicos que desean realizar consultas a bases de datos complejas. Además, permite la creación de vistas, procedimientos almacenados y disparadores para automatizar tareas repetitivas en las bases de datos.



Imagen: SQL: Lenguaje de Consulta Estructurada

Fuente: <https://ojt.com/wp-content/uploads/2021/08/sql.png>

3.6 Almacenamiento de datos

El almacenamiento de datos es el proceso de guardar y conservar información en un dispositivo de almacenamiento de manera permanente o temporal. Estos dispositivos incluyen discos duros, unidades flash USB, discos externos, discos sólidos (SSD), entre otros. La información se puede almacenar en diferentes formatos, como texto, imágenes, audio, video, etc. El almacenamiento de datos es fundamental para el funcionamiento de cualquier sistema informático, ya que permite guardar información y recuperarla cuando sea necesario. Además, el almacenamiento de datos se utiliza en una amplia variedad de aplicaciones, desde la gestión de bases de datos hasta la nube y la gestión de datos empresariales.



Imagen: almacenamiento de tipos de datos

Fuente: <https://www.tecnologias-informacion.com/datalake.jpg>

3.7 Creaciones de tablas Data

La creación de tablas de datos depende del lenguaje de programación o herramienta que se utilice.

Las tablas de datos son una estructura de almacenamiento común en la informática que permite organizar y representar información de manera clara y eficiente. Las tablas de datos consisten en filas y columnas, donde cada fila representa un registro o una unidad de información, y cada columna representa una categoría o una característica de ese registro.

Las tablas de datos se utilizan ampliamente en diversas aplicaciones, incluyendo bases de datos, hojas de cálculo, análisis de datos y mucho más. Pueden ser creadas y manipuladas con diversas herramientas y lenguajes de programación, incluyendo SQL, Python, R, y muchos otros.

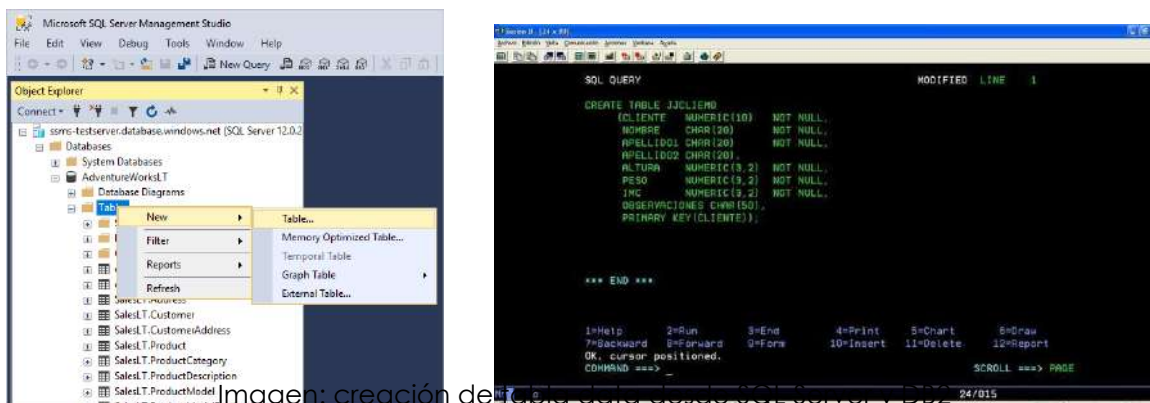


Imagen: creación de...

Fuente: <http://3.bp.blogspot.com/-ldM9SRa3V3M/Ue-KEJxl1ev/AAAAAAAAALo/WtI6pRMhQBQ/s1600/Acceso+QMF.JPG>

Fuente: <https://learn.microsoft.com/es-es/sql/ssms/media/design-tables/new-table.png?view=sql-server-ver16>

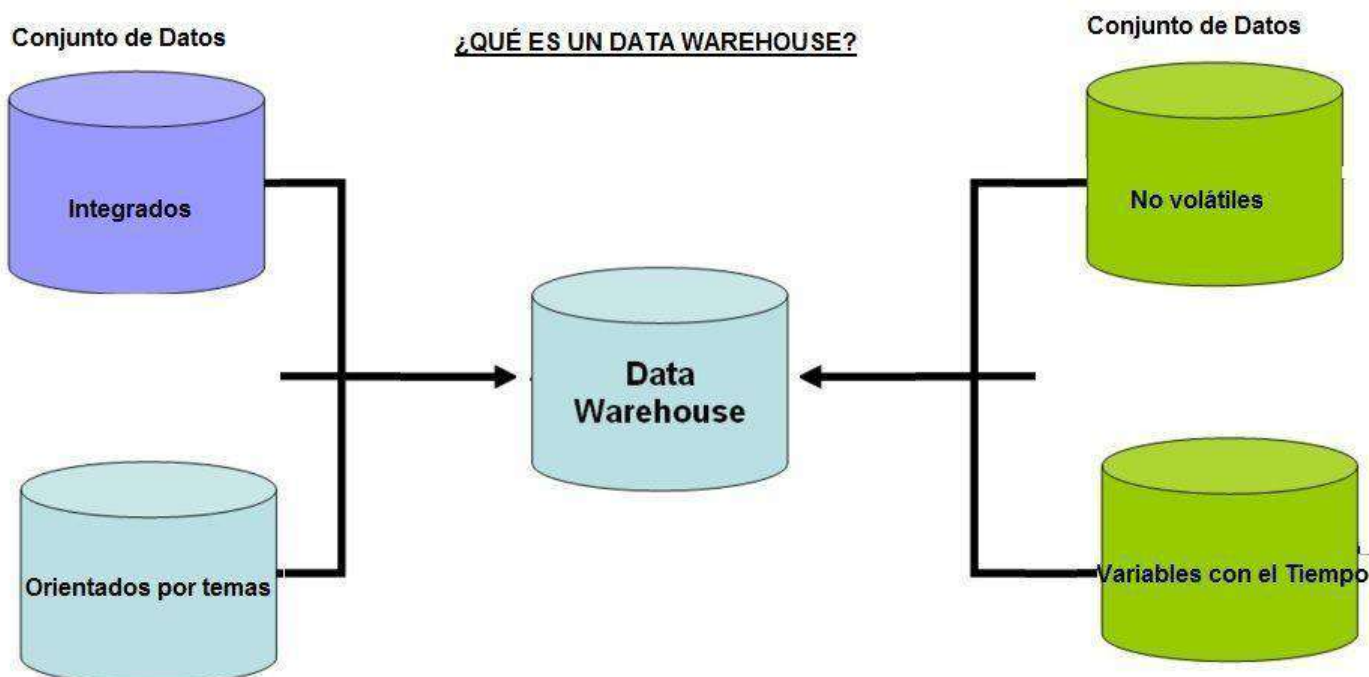
3.8 Warehouse

Por: Ricardo T.

En informática, un almacén de datos (en inglés, "warehouse") es un sistema centralizado y organizado que almacena grandes cantidades de datos, los cuales pueden ser utilizados para análisis, reportes y toma de decisiones. Estos almacenes de datos se crean para mejorar la eficiencia en la recolección, integración y accesibilidad de los datos, lo que permite a las empresas y organizaciones tomar decisiones más informadas.

Un almacén de datos se compone de una gran cantidad de información que se extrae de diversas fuentes, incluyendo sistemas transaccionales, archivos, sistemas de seguimiento, entre otros, y se integra y organiza de manera estructurada en un solo lugar. Además, se utiliza tecnología de bases de datos para optimizar la velocidad y la eficiencia en el acceso y análisis de los datos.

En resumen, un almacén de datos es un sistema de gestión de información que se utiliza para almacenar, integrar y analizar grandes cantidades de datos para mejorar la toma de decisiones y la eficiencia en la gestión de información.



Fuente: <https://www.areatecnologia.com/informatica/imagenes/que-es-un-data-warehouse.jpg>

Características:

algunas características adicionales que se pueden mencionar sobre los almacenes de datos:

1. Integración de datos: Los almacenes de datos combinan datos de diversas fuentes para crear una vista única y completa de la información.
2. Escalabilidad: Los almacenes de datos deben ser escalables para poder manejar grandes cantidades de datos y proporcionar un rendimiento óptimo en el análisis.

3. Accesibilidad: Los almacenes de datos deben ser fácilmente accesibles para los usuarios y aplicaciones que los necesitan, y deben proporcionar una forma sencilla de realizar consultas y análisis

4. Confiabilidad y seguridad: La información almacenada en un almacén de datos debe ser confiable y segura, y deben implementarse medidas de seguridad para proteger la información y garantizar su privacidad.

5. Flexibilidad: Los almacenes de datos deben ser flexibles y permitir la adición de nuevos datos y la realización de actualizaciones sin interrupciones en el servicio.

En conclusión, los almacenes de datos son un componente importante en la gestión de información y proporcionan una gran cantidad de valor a las empresas y organizaciones que los utilizan.

3.9 Introducción a DB2

Antes de las Bases de Datos se utilizaban archivos secuenciales para almacenar datos. Estos daban un acceso muy rápido pero sólo de forma secuencial en donde para acceder a una posición, se debía recorrer el archivo entero. Más tarde aparecieron los archivos indexados, donde el acceso ya podía ser aleatorio (acceder de una vez a la posición deseada del mismo).

El sistema de archivos era el sistema más común de almacenamiento de datos. Para compartirlos entre varias máquinas surgió el NFS2 (Network file system), y más tarde para evitar fallos en los sistemas de archivo aparecieron los sistemas RAID3 (Redundant Array Of Independent/Inexpensive Disks).

Pero los programas y datos cada vez eran más complejos y grandes. Se requería de un almacenamiento que garantizara un cierto número de condiciones y que permitiera operaciones complejas sin que se violaran estas restricciones. Además cada usuario que accediera a los datos debía tener su trabajo protegido de las operaciones que hicieran el resto de usuarios.

Respondiendo a estas necesidades, surgieron las Bases de Datos Jerárquicas, donde los datos se situaban siguiendo una jerarquía.

Las Bases de Datos Jerárquicas tenían el problema que los accesos a los datos eran unidireccionales, y era más complicado hacer el camino inverso (pero posible, aunque el tiempo de cálculo era mayor).

Por ejemplo: Era fácil saber qué cuentas tenía un cliente, pero no tanto saber de qué cliente era una cierta cuenta.

Para dar absoluta libertad a las relaciones entre tablas surgieron las Bases de Datos Relacionales (Relational Data Base Management System):



Imagen: Lanzamiento de DB2 en 1995 por IBM

Fuente: <https://www.next-decision.fr/images/editeurs/stockage/ibm-db2.jpg>

3.10 ¿Qué es DB2?

Db2 es una familia de productos de sistemas de gestión de bases de datos (DBMS) de IBM que sirven a varias plataformas de sistemas operativos (SO) diferentes.

DB2 es un sistema de gestión de bases de datos relacionales (RDBMS) introducido originalmente por IBM en 1983 para ejecutarse en su plataforma mainframe MVS (almacenamiento virtual múltiple).

¿Qué es una base de datos Db2 y cómo funciona?

Una base de datos Db2 es un grupo de datos tratados agrupadamente como una unidad. Una base de datos es un gran conjunto estructurado de datos duraderos y su propósito es almacenar, recuperar y manipular información relacionada.

Para crear y utilizar bases de datos Db2. El DBMS de Db2 funciona como servidor para gestionar datos en bases de datos en un entorno multiusuario, lo que permite que muchos usuarios concurrentes accedan a los mismos datos simultáneamente.

Las bases de datos DB2 tienen estructuras lógicas y estructuras físicas que el DBMS gestiona por separado. El almacenamiento físico de datos se puede administrar sin afectar el acceso a las estructuras de almacenamiento lógicas. Las bases de datos Db2 se crean utilizando mandatos del lenguaje de definición de datos, o DDL, y se componen de espacios de tabla, tablas con filas y columnas, vistas, índices, procedimientos almacenados y otras estructuras de soporte.



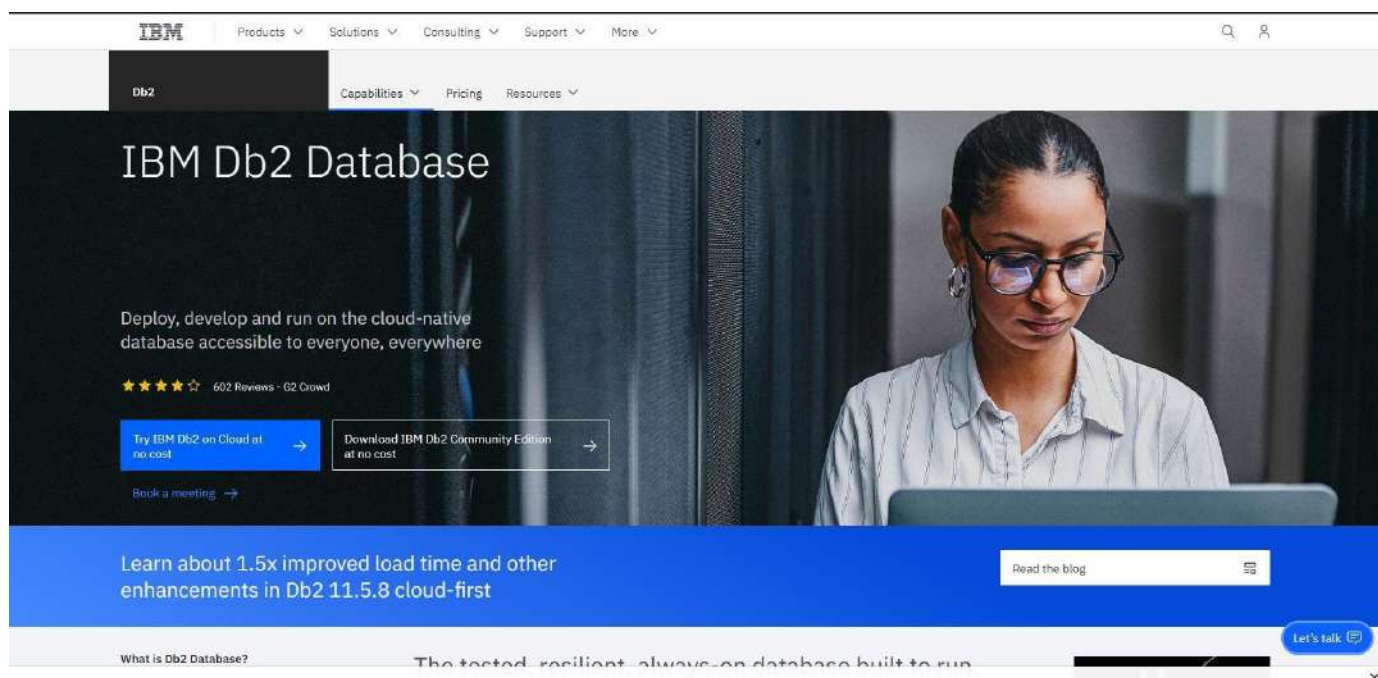
Imagen: IBM Courses-BlueMark Academy

Fuente: <https://bluemarkacademy.com/wp-content/uploads/2020/12/db2-for-zos-logo.jpg>

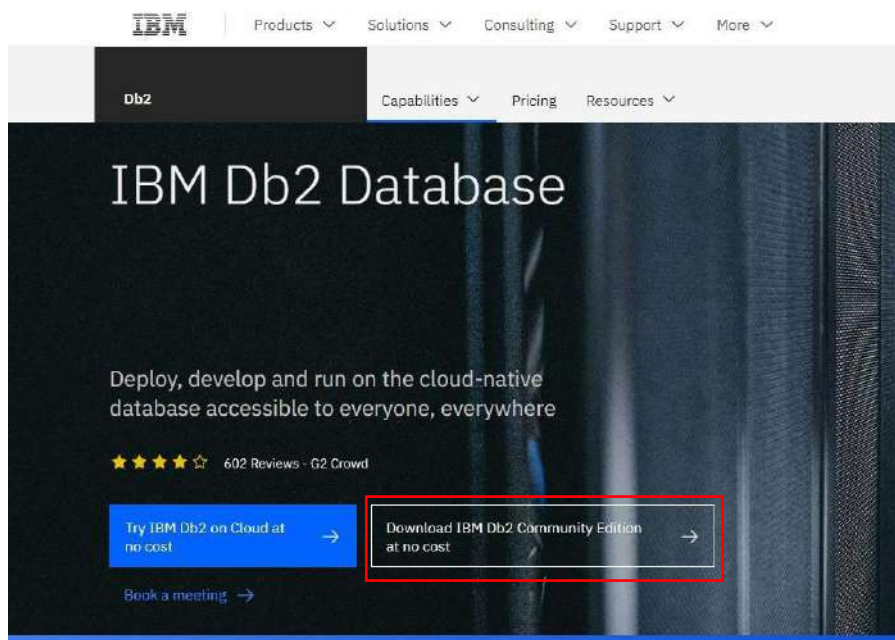
3.11 Instalación del servidor de base de datos

Para instalar DB2 tendremos que ingresar al siguiente link:

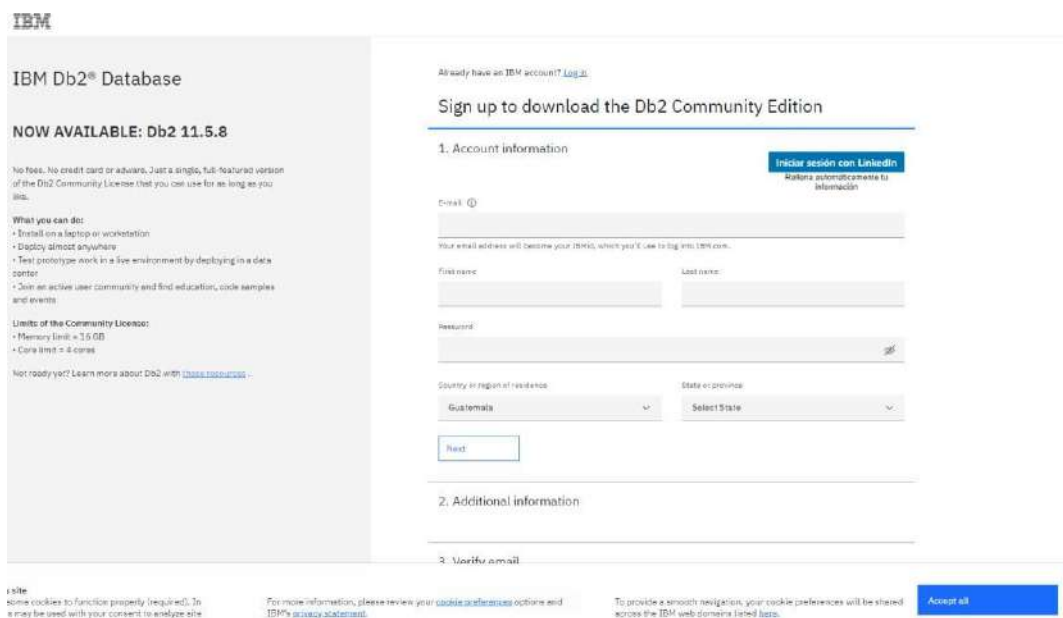
<https://www.ibm.com/products/db2/database> , la cual nos llevará a la página de descarga



1) Clic en Download IBM Db2 Community Edition at no cost



2) Debemos de registrarnos para poder acceder a las descargas



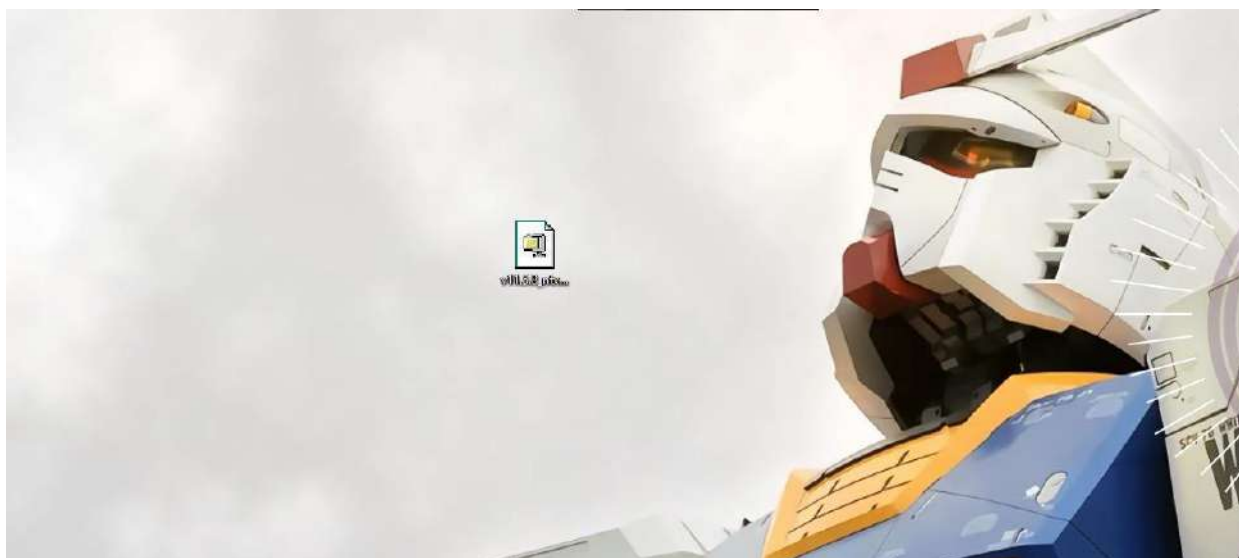
3) Descargaremos la versión de Windows



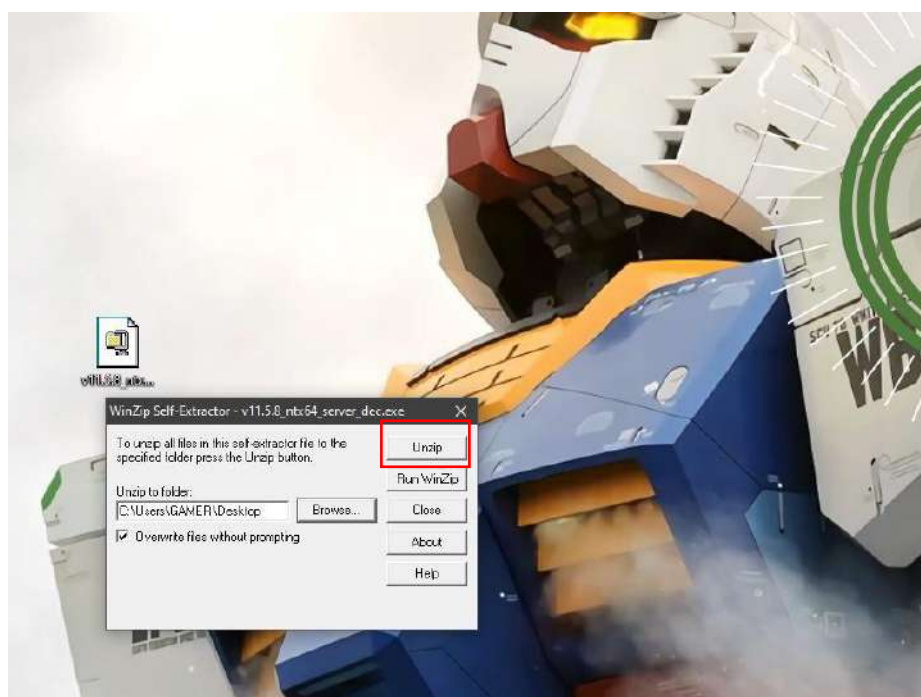
IBM Db2 v11.5.8 current

Microsoft Windows (x64)	1.4GB	Download
Linux (x64)	1.9GB	Download
Linux on IBM Z	1.4GB	Download
Linux (Power PC)	1.7GB	Download
IBM AIX	1.8GB	Download

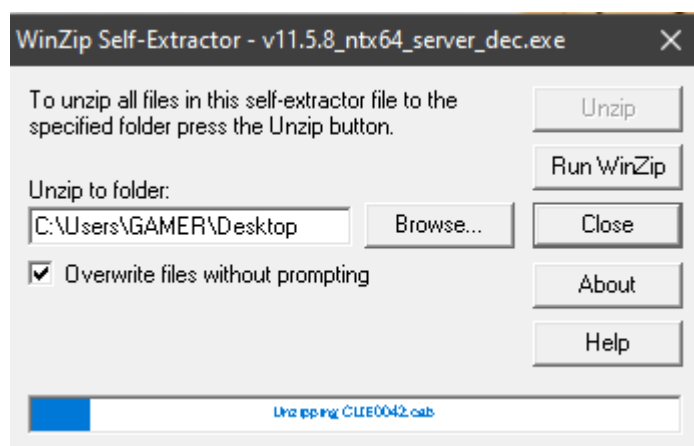
- 4) Una vez descargada, nos dejara un archivo ejecutable para descomprimir los archivos, lo ejecutamos



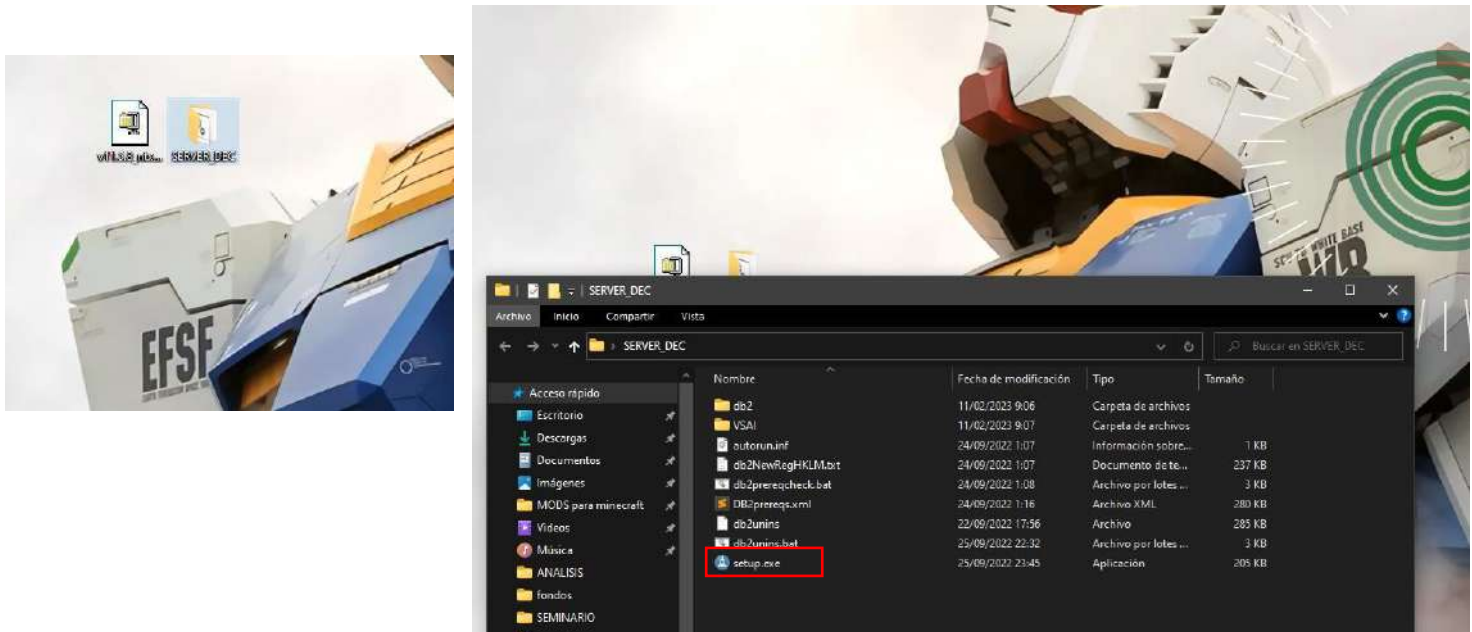
- 5) Se nos abrirá una ventana donde escogeremos la ruta donde queremos que se descomprima la carpeta y le daremos en unzip



Esperamos a que se descomprima



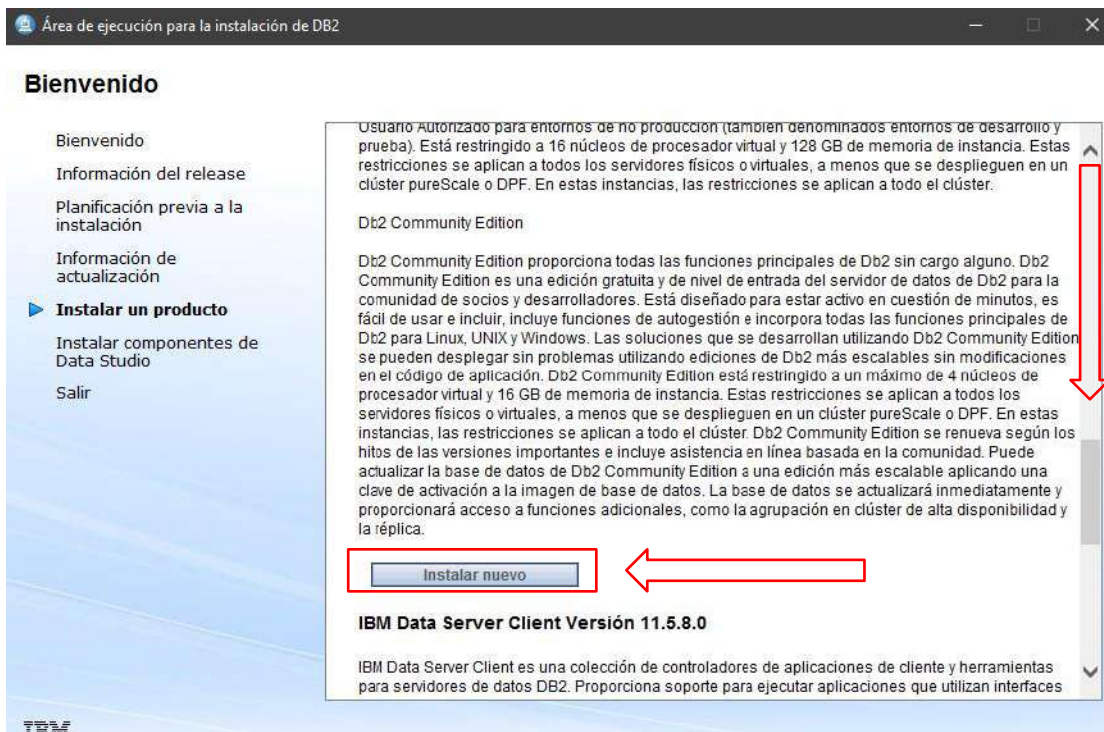
- 6) Una vez descomprimida nos dejara una carpeta llamada SERVER_DEC, Para instalar DB2, hacemos doble clic en el archivo setup.exe, se iniciará el programa de instalación de DB2:



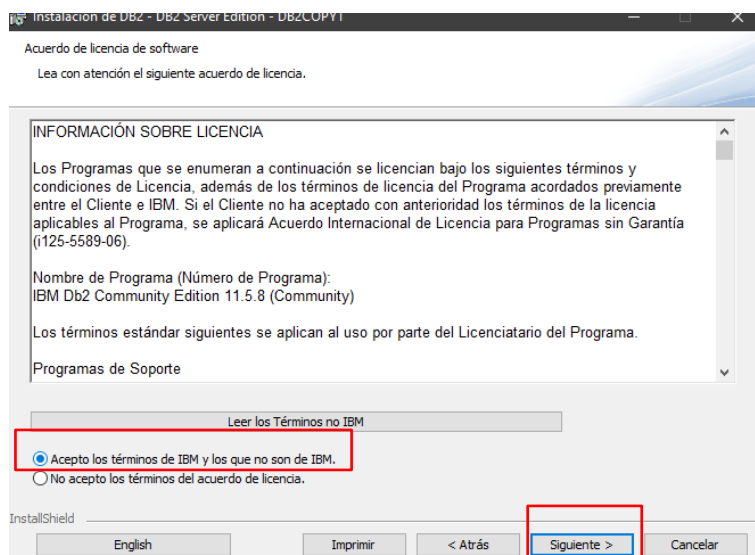
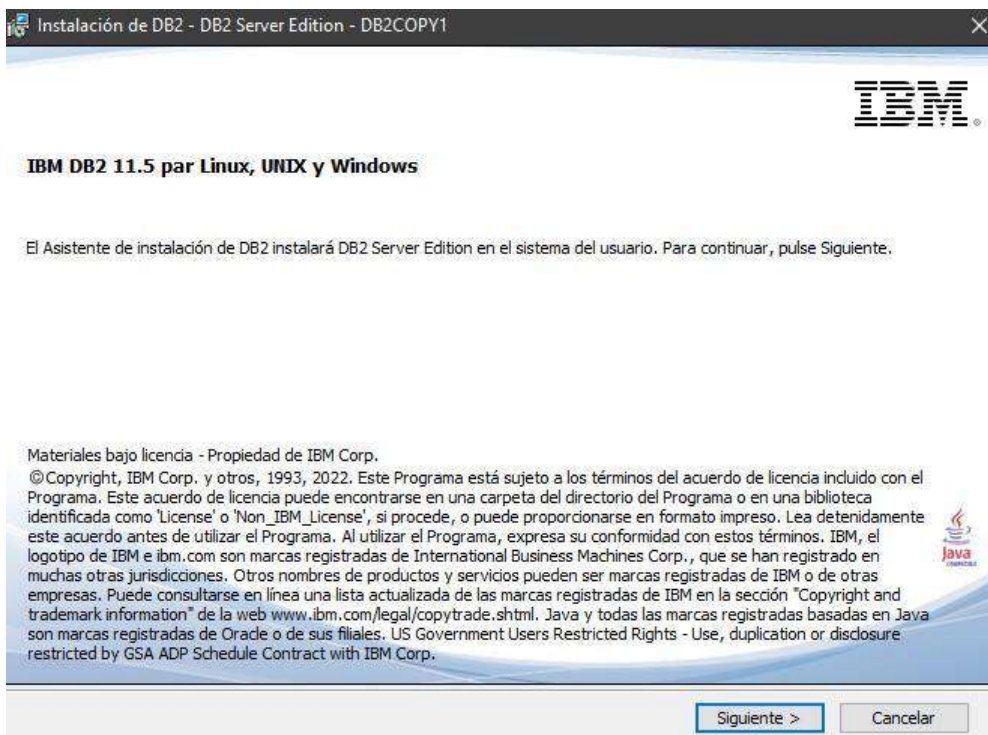
- 7) Se nos abrirá una ventana donde nos darán la bienvenida y obtendremos información de actualizaciones, nos ubicaremos en Instalar un producto



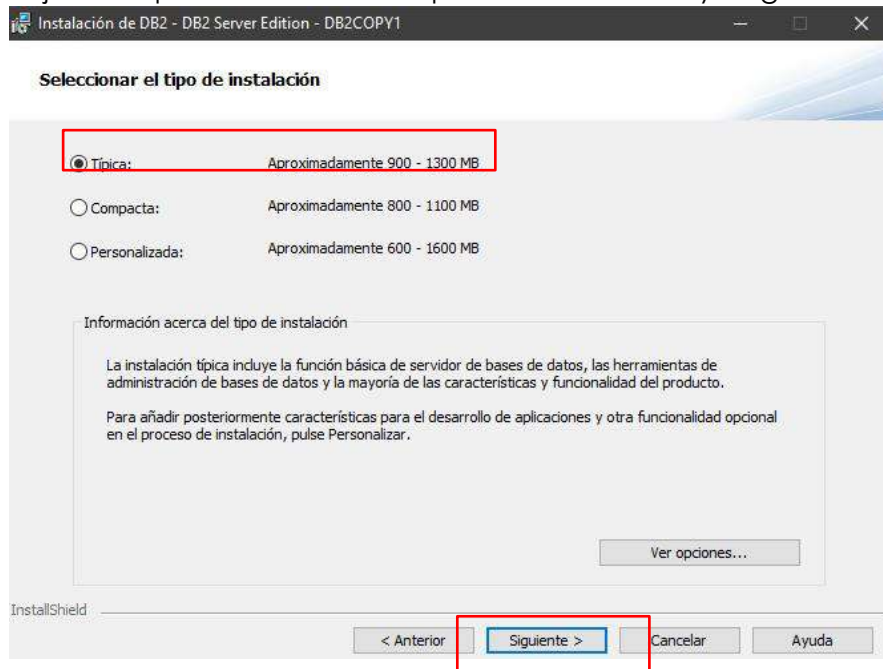
- 8) Desplazamos la barra de desplazamiento hacia abajo y haga clic en el botón Instalar **nuevo**



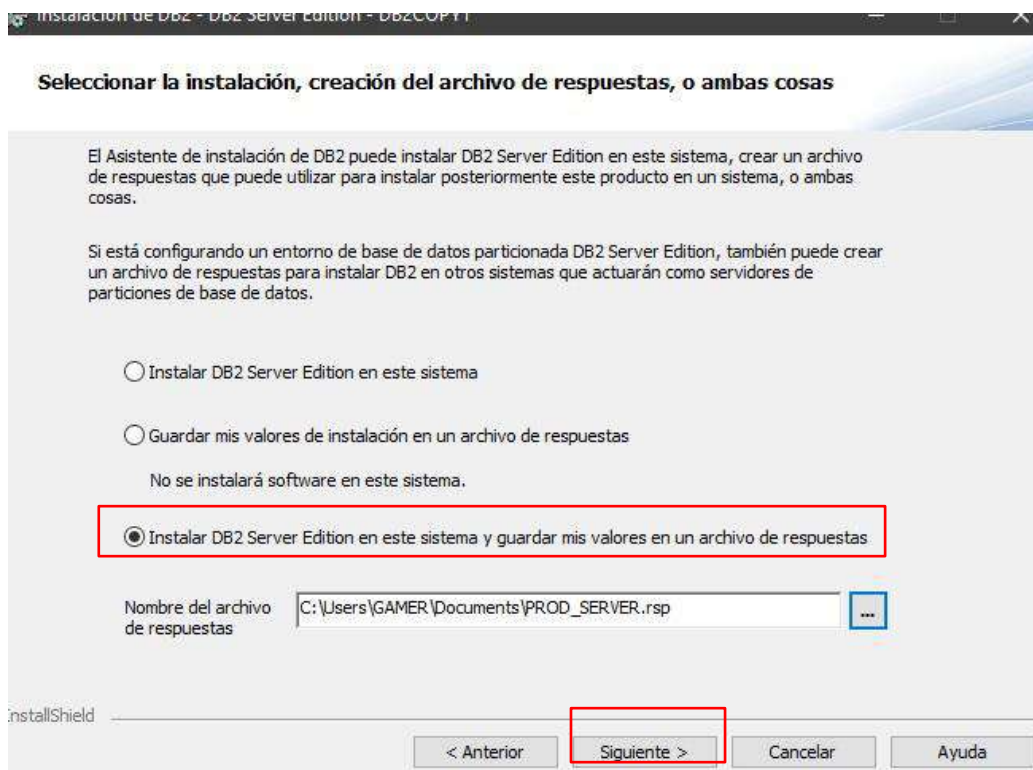
9) Nos enviará a otra ventana donde nos dirán sus términos y condiciones, le daremos siguiente



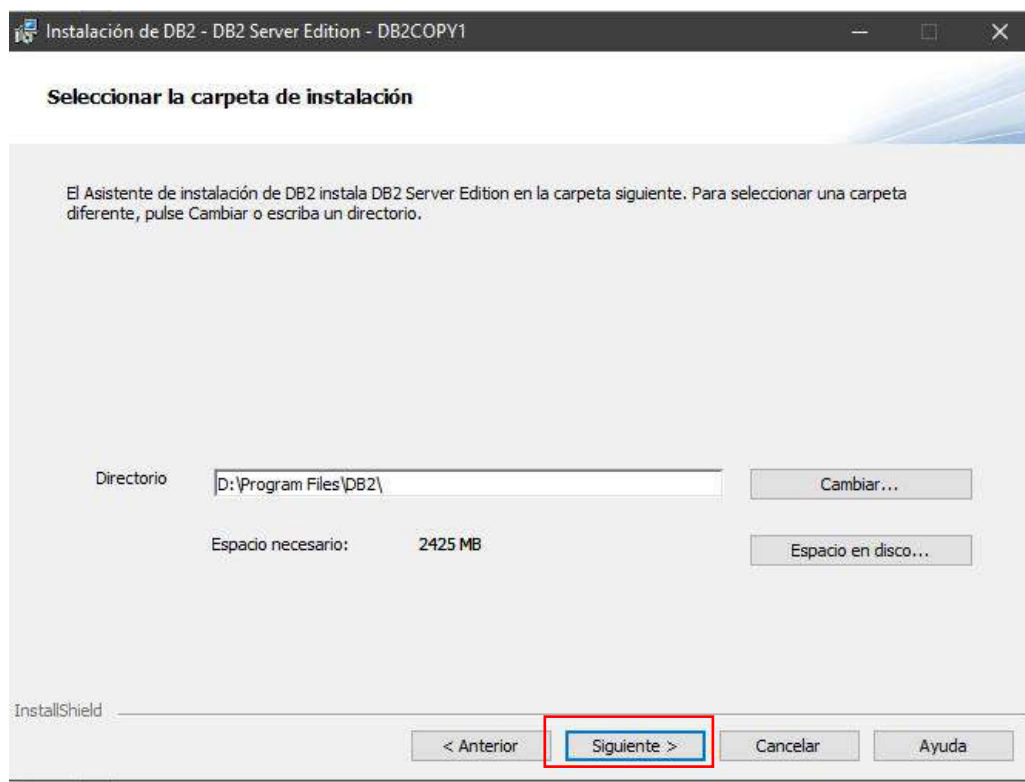
10) Elijamos Típica en la lista de tipos de instalación y haga clic en el botón Siguiente:



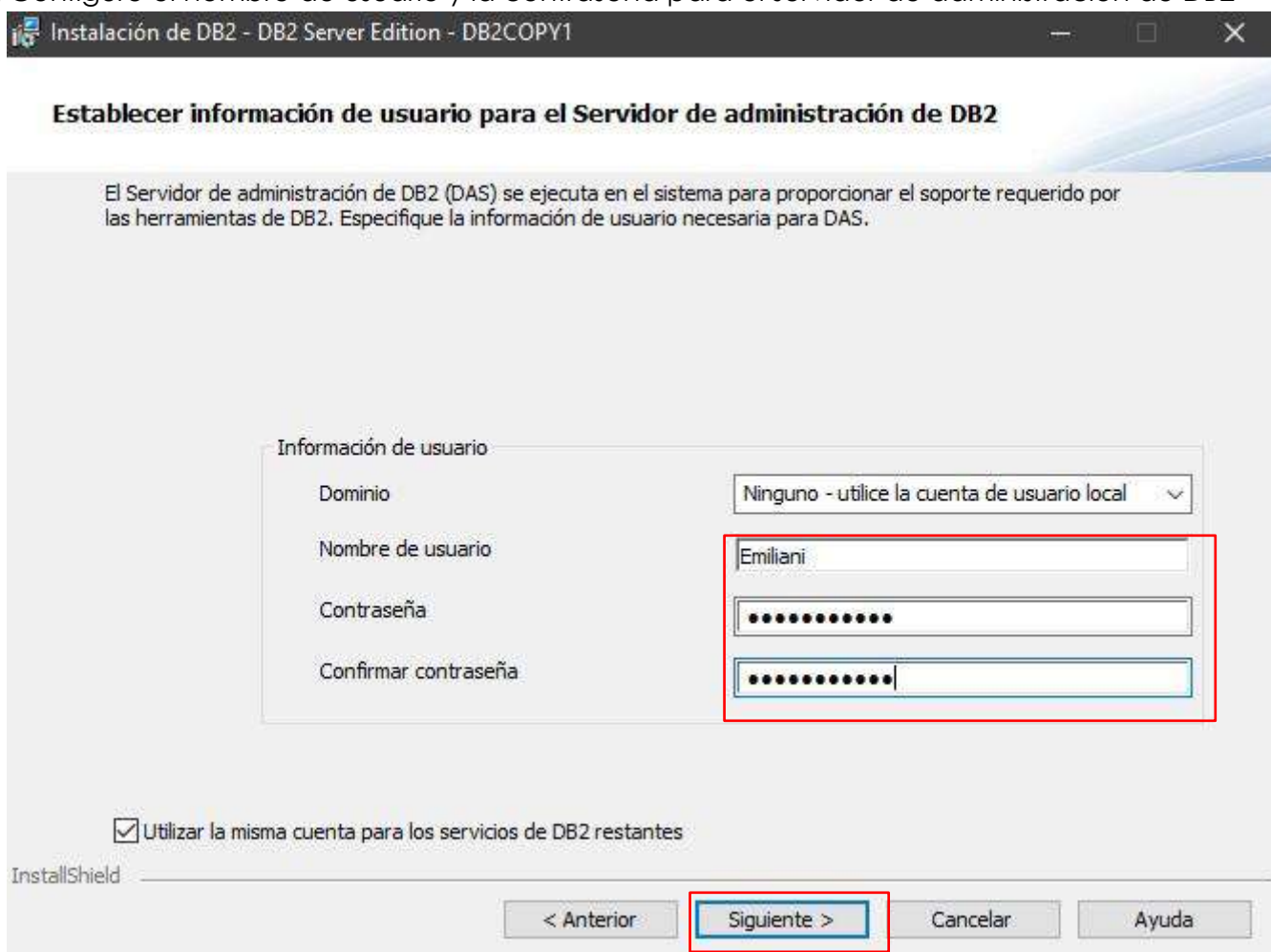
11) Seleccione la tercera opción para crear un archivo de respuesta y haga clic en el botón Siguiente



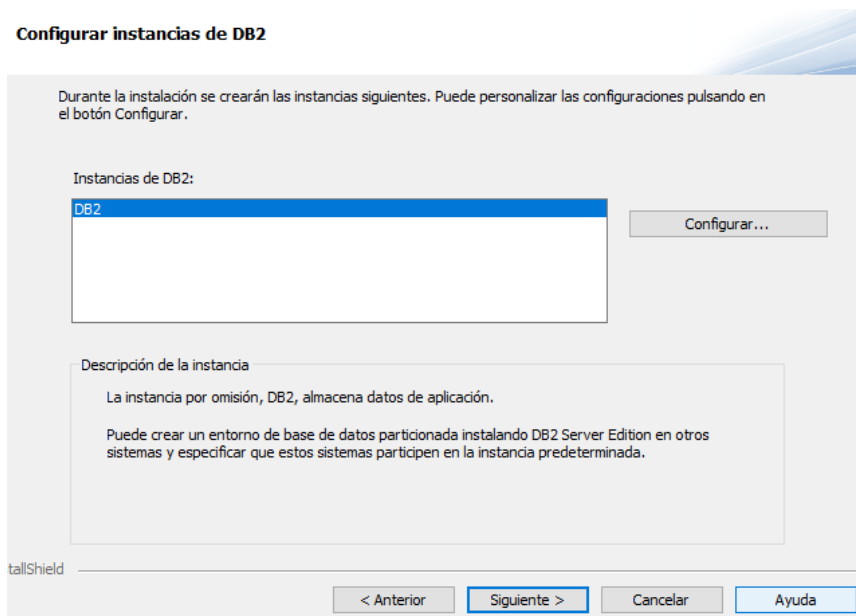
12) Seleccione la carpeta de instalación, déjela como está y haga clic en el botón Siguiente



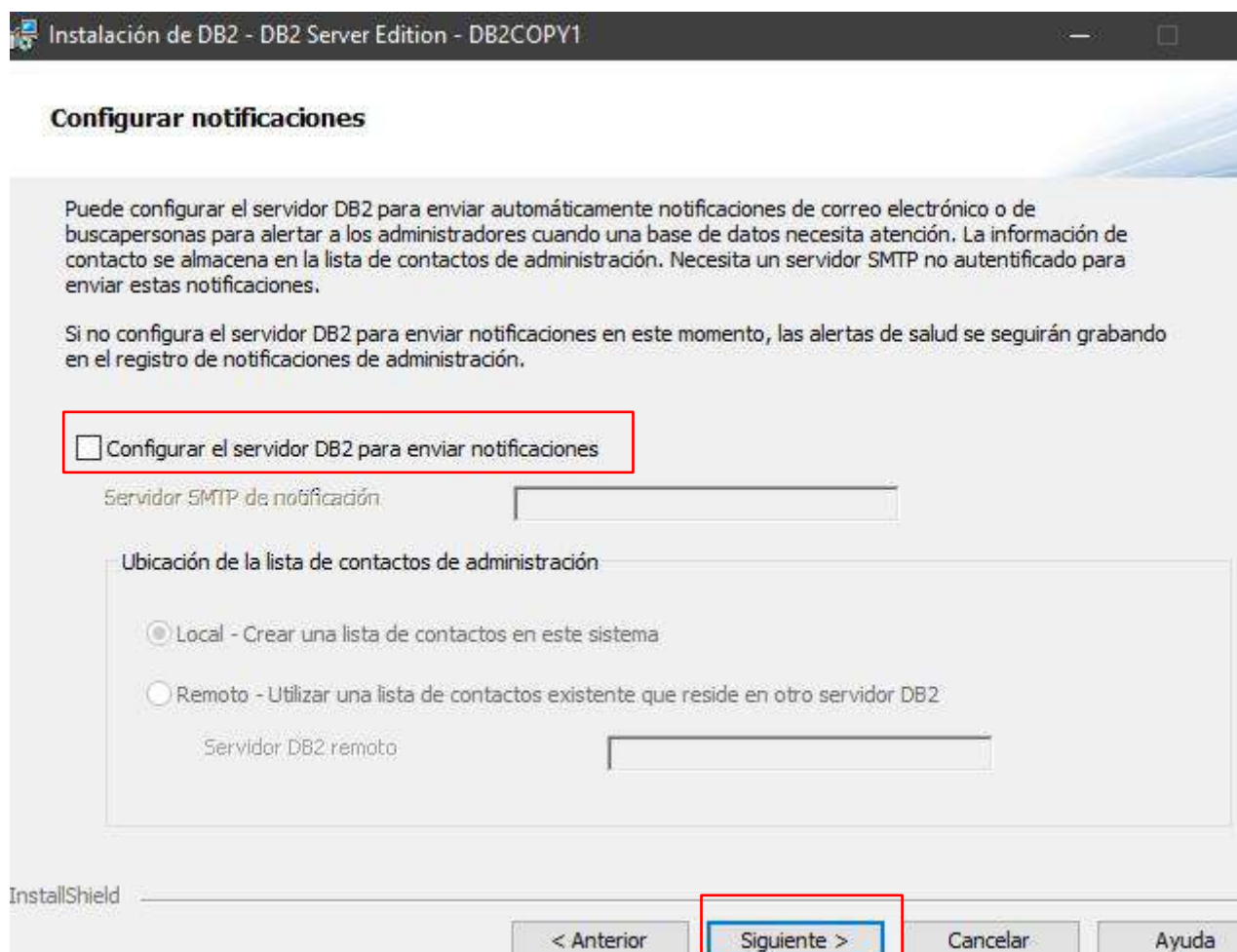
13) Configure el nombre de usuario y la contraseña para el servidor de administración de DB2



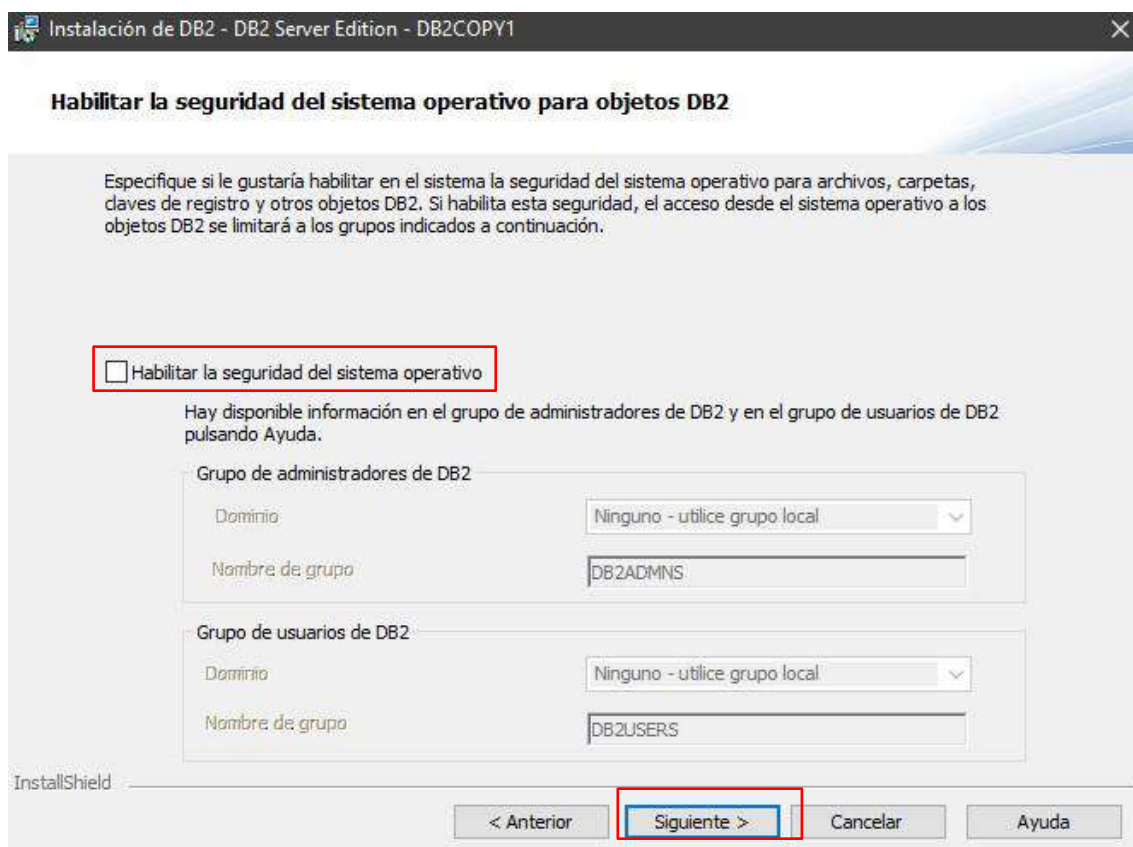
14) Configure las instancias de DB2, déjelo como DB2 y haga clic en el botón Siguiente



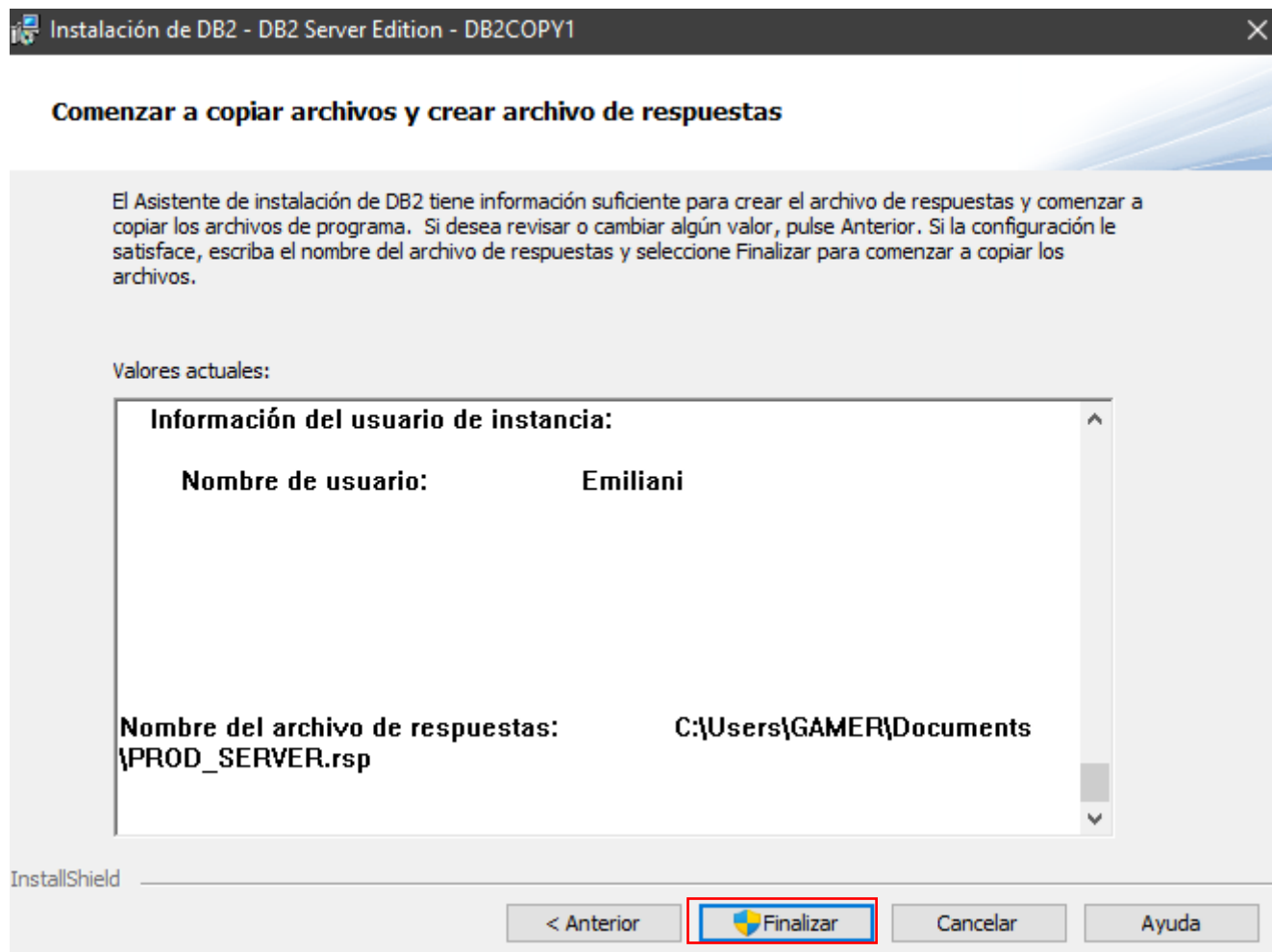
15) Configure las notificaciones, desmarque la casilla de verificación y haga clic en el botón Siguiente



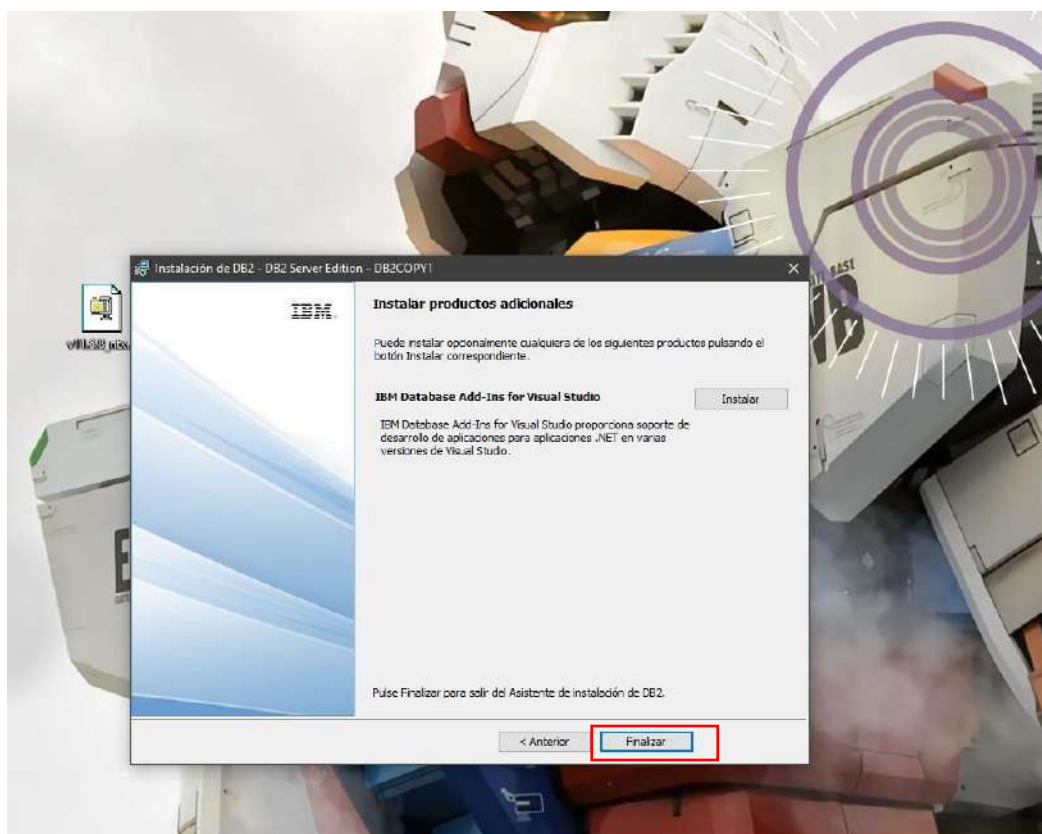
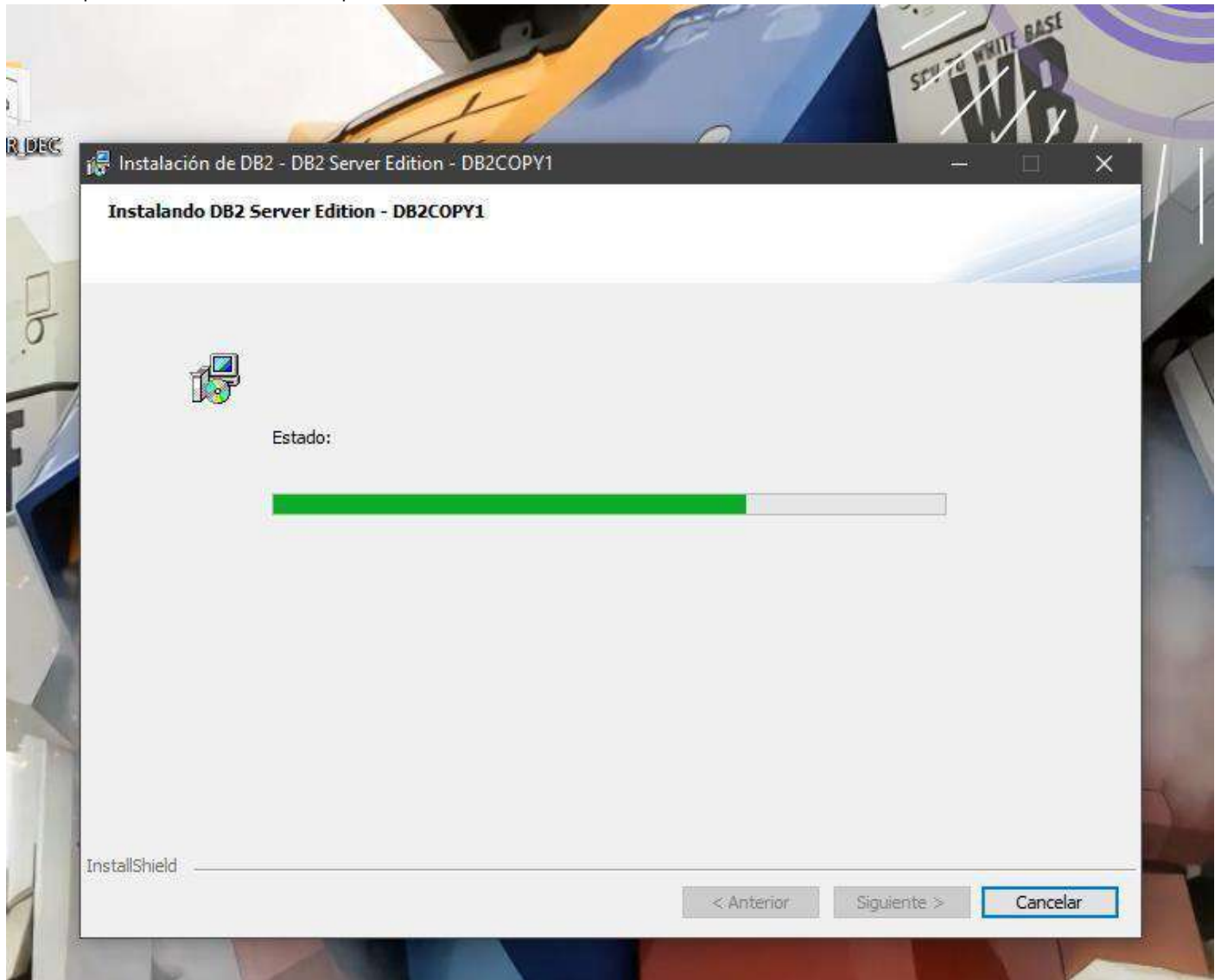
16) Habilite la seguridad del sistema operativo para los objetos de DB2, desactive la casilla de verificación **Habilitar la seguridad del sistema operativo** y haga clic en el botón Siguiente:

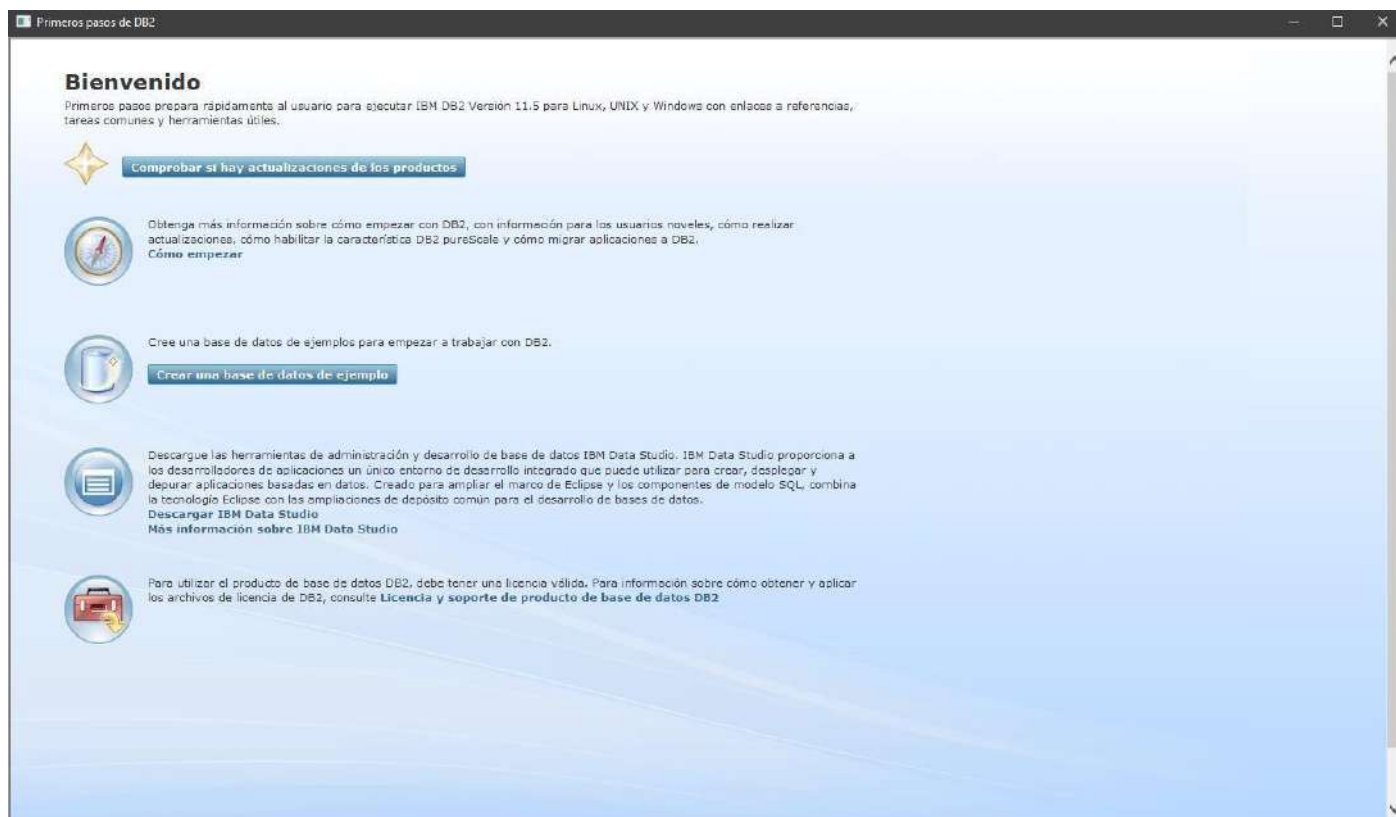


17) Revisa la información haz clic en el botón Finalizar



Se empezara a instalar la aplicación, al terminar le daremos en finalizar





Fuente: PROPIA

3.12 Configuración DB2

Por: José Herrera

Desde los primeros pasos de la instalación podemos configurar DB2, la mayoría de configuraciones se deben dejar por defecto. La primera de ellas es el tipo de instalación, lo mejor es elegir la opción "típica".

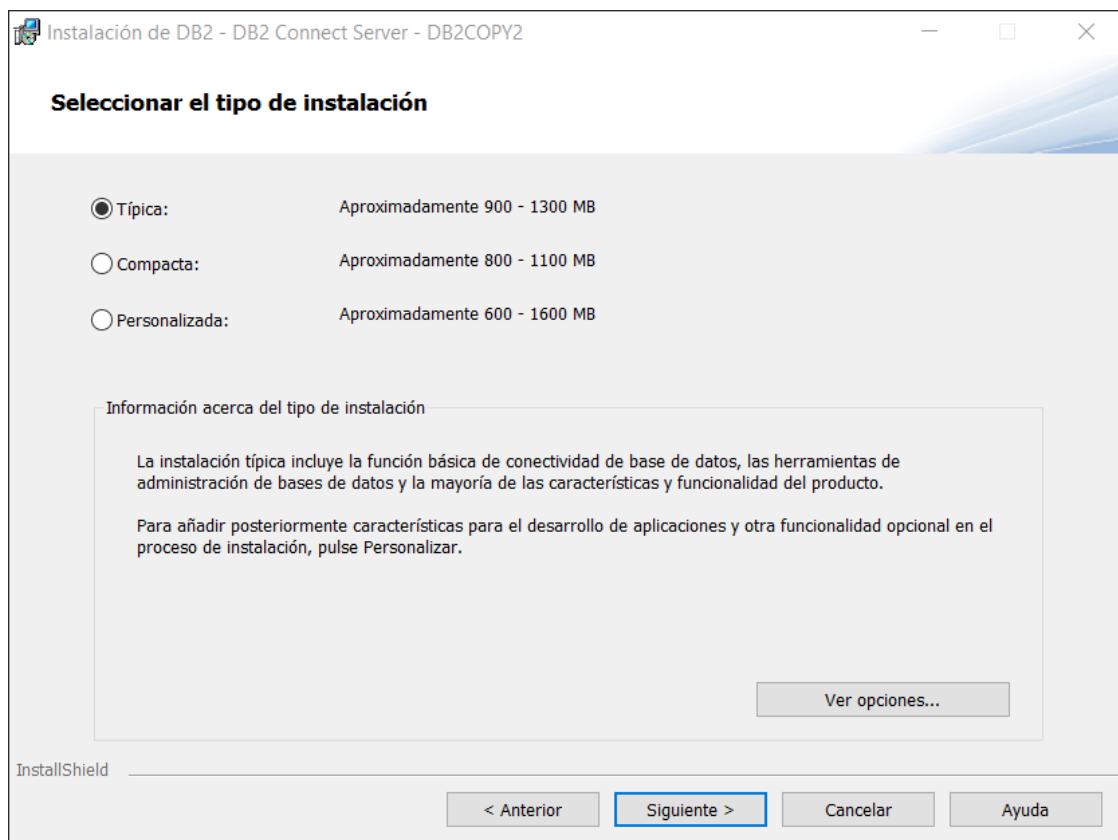


Imagen: Instalación típica de DB2
Fuente: José Santiago Herrera Furlán

A continuación, se pide elegir si se quiere crear un "archivo de respuestas", al igual que antes, lo mejor es dejarlo en la opción por defecto.

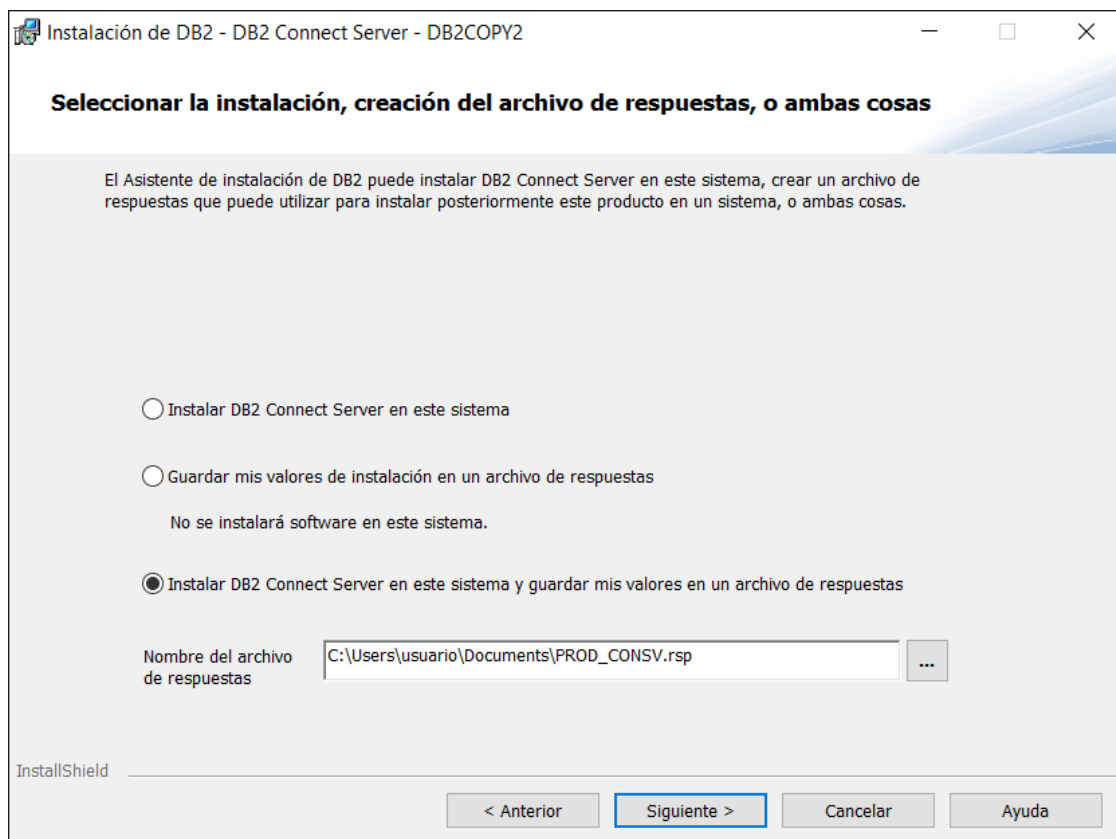


Imagen: Configuración del archivo de respuestas de DB2
Fuente: José Santiago Herrera Furlán

Después, se pide elegir el directorio de instalación. Para evitar errores posteriores intentando conectar la base de datos, es recomendable dejarlo por defecto.

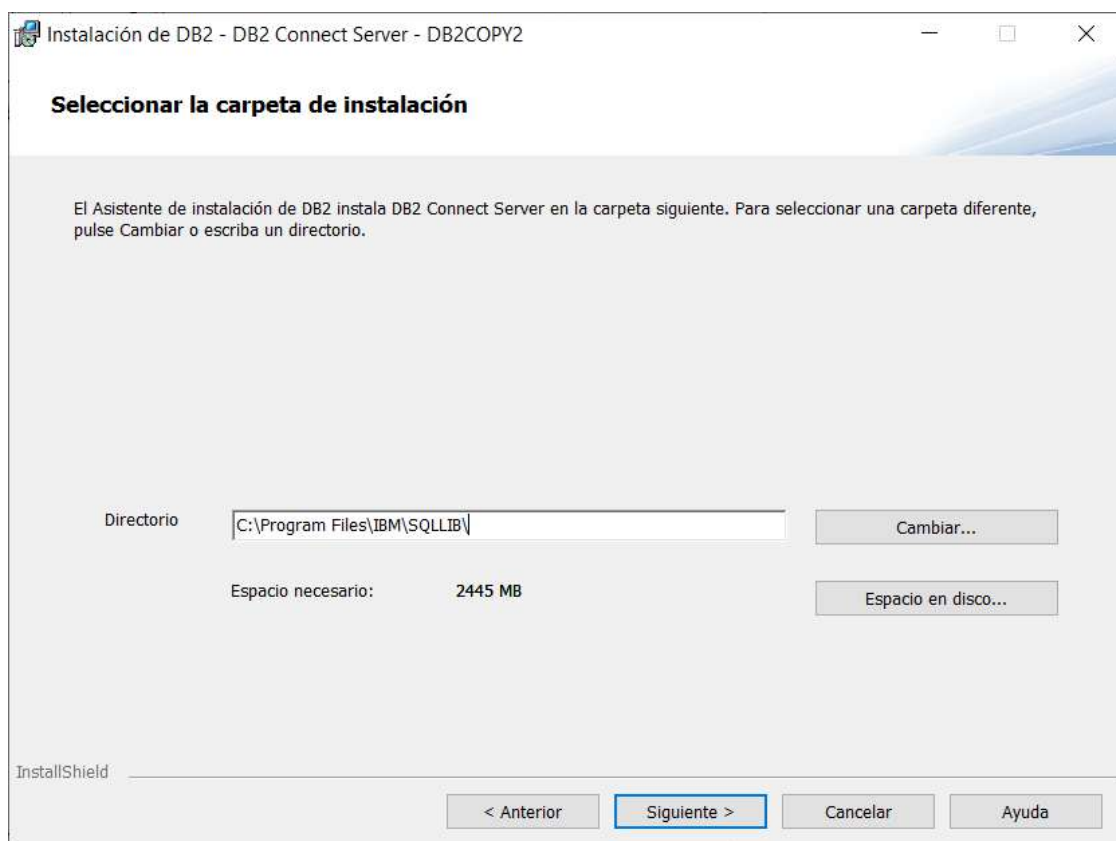


Imagen: Elección del directorio de instalación de DB2
Fuente: José Santiago Herrera Furlán

Después, el instalador pide ingresar la información del usuario que será responsable de gestionar el sistema. Existen 3 campos importantes:

- **Dominio:** La dejaremos en su valor por defecto.
- **Nombre de usuario:** En este campo va el nombre del usuario maestro que gestionará todo el sistema, es recomendable dejarle el nombre por defecto "db2admin".
- **Contraseña:** Aquí pondremos una contraseña segura para acceder a DB2, es muy importante recordar esta contraseña para realizar futuras conexiones a las bases de datos.

Imagen: Configuración completa de la información de usuario de DB2
Fuente: José Santiago Herrera Furlán

En el siguiente apartado se permite configurar las instancias de DB2, que son los entornos en donde se almacenan las bases de datos. En este caso se puede dejar la configuración por defecto.

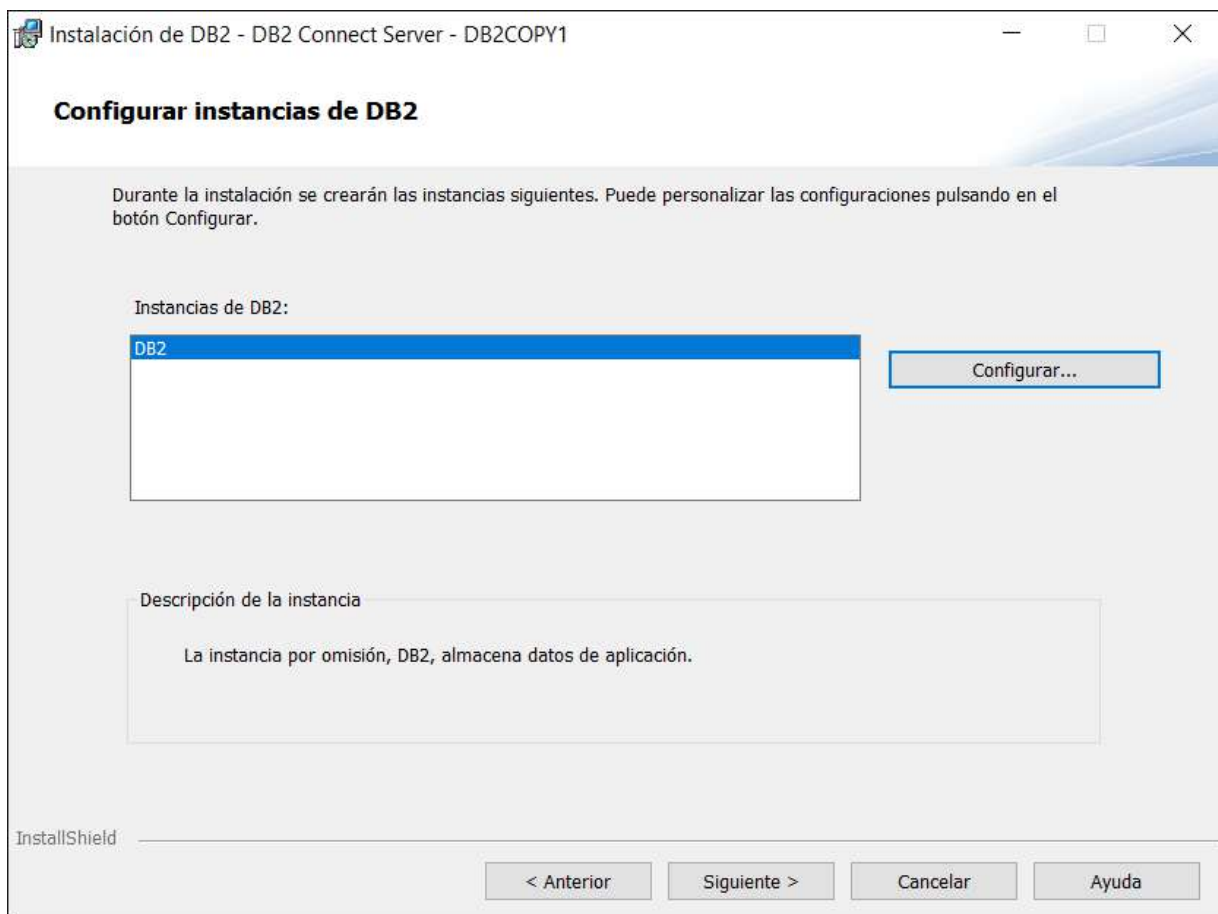


Imagen: Elección de las instancias de DB2
 Fuente: José Santiago Herrera Furlán

Después se permite configurar las notificaciones automáticas. Estas notificaciones serán enviadas al servidor SMTP indicado, sin embargo, si se desea, se pueden desactivar totalmente las notificaciones.

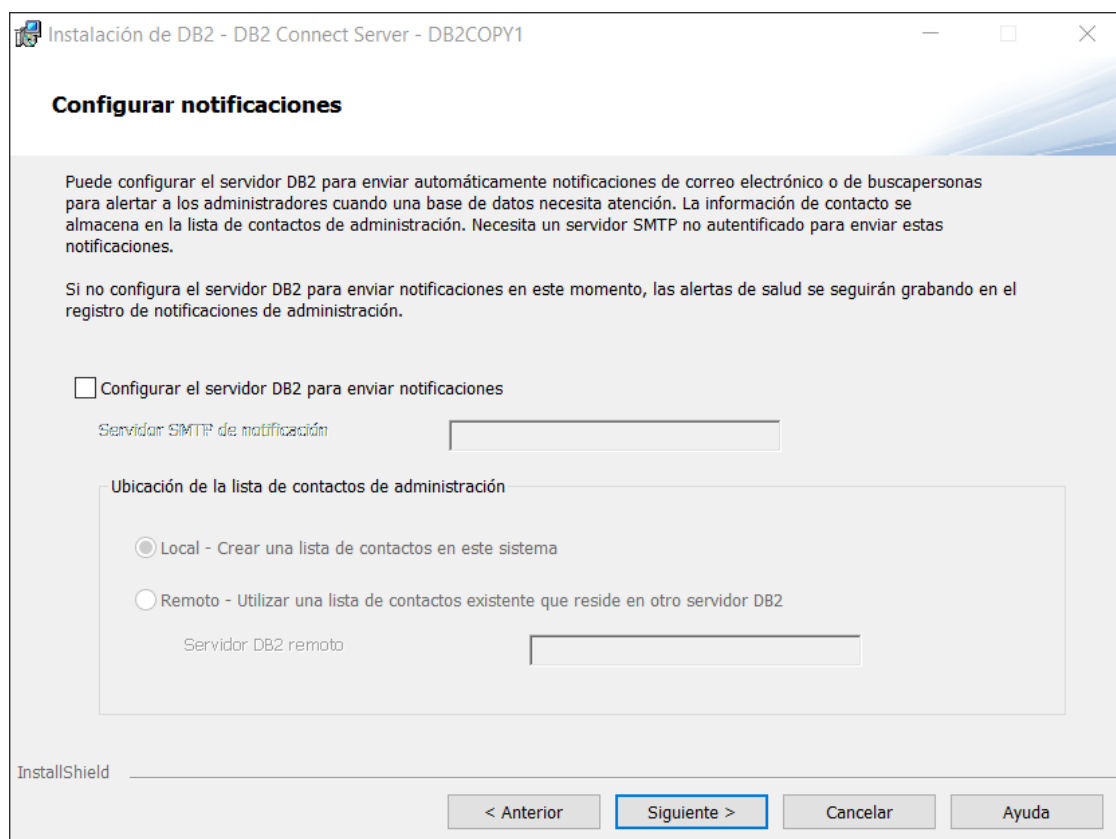


Imagen: Configuración para tener notificaciones desactivadas en DB2

Por último, se pide configurar la seguridad de DB2. La seguridad del gestor de base de datos limita el acceso en una red a dos grupos: uno para administradores y uno para los usuarios. Al igual que con las notificaciones es posible desactivar esta característica que es inútil si sólo un usuario va a utilizar DB2.

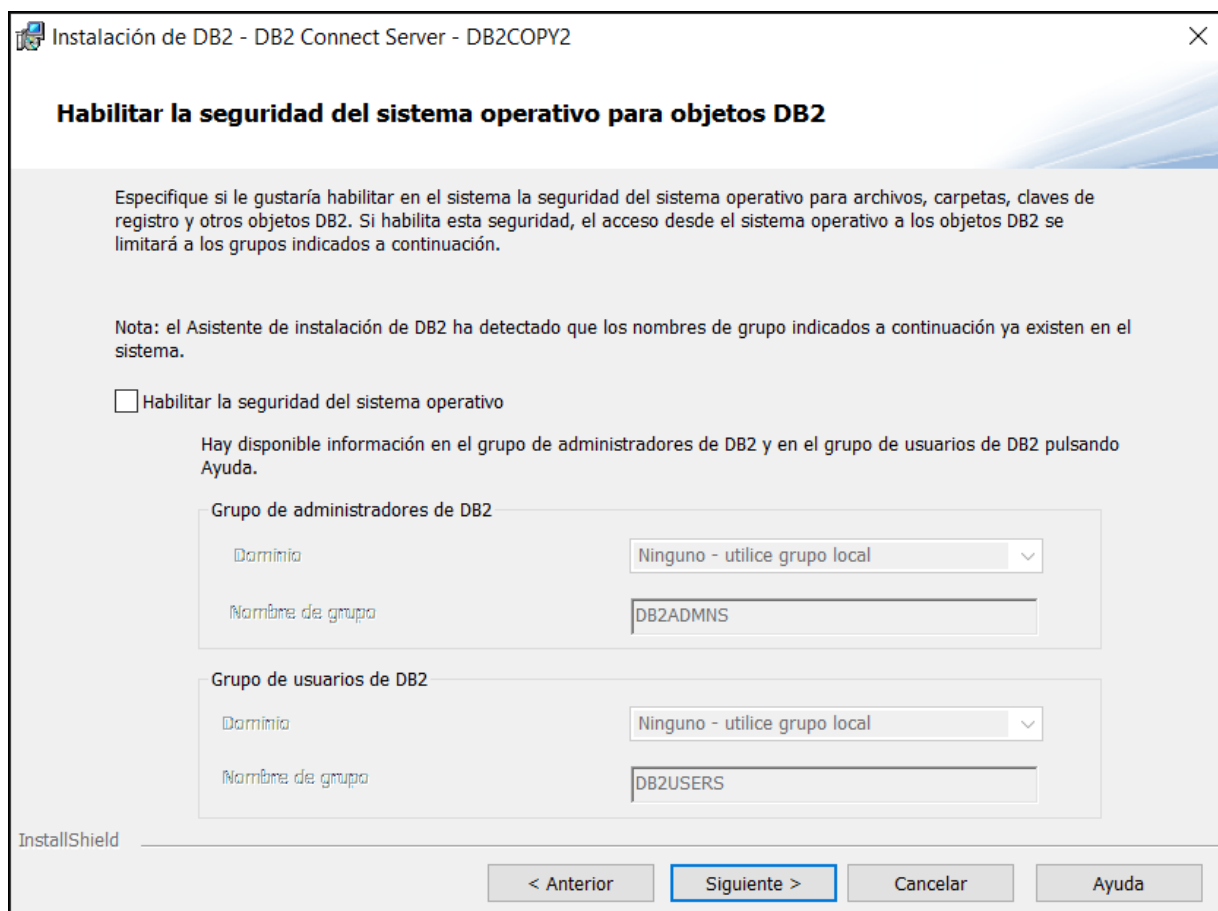


Imagen: Configuración para tener la seguridad desactivada en DB2
 Fuente: José Santiago Herrera Furlán

3.13 Base de datos de muestra DB2

Por: José Herrera

Para crear la primera base de datos en DB2, primero se debe ejecutar la aplicación del procesador de comandos, desde esta aplicación podremos inyectar instrucciones SQL a nuestro servidor y así crear nuestra primera base de datos.

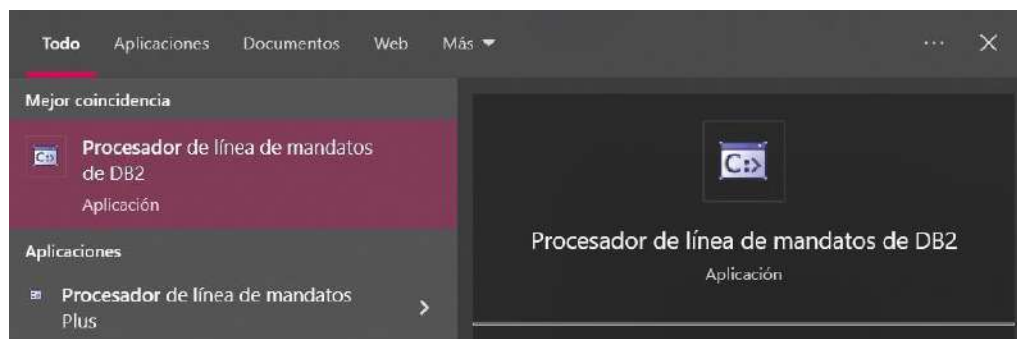


Imagen: Procesador de línea de comandos de DB2 en el buscador de Windows

Utilizando instrucciones SQL crearemos nuestra primera base de datos. Este proceso puede tardar unos momentos. La instrucción SQL es la siguiente:

CREATE DATABASE ejemplo

```
(c) Copyright IBM Corporation 1993,2007
Procesador de línea de mandatos para DB2 Client 11.5.8.0

Puede emitir mandatos del gestor de bases de datos y sentencias de SQL desde
el indicador de mandatos. Por ejemplo:
    db2 => connect to sample
    db2 => bind sample.bnd

Para la ayuda general, escriba: ?.
Para la ayuda sobre los mandatos, escriba: ? mandato, donde mandato puede ser
las primeras palabras clave de un mandato del gestor de bases de datos. Por
ejemplo:
? CATALOG DATABASE para ayuda sobre el mandato CATALOG DATABASE
? CATALOG           para ayuda sobre todos los mandatos CATALOG.

Para salir de la modalidad db2 interactiva, escriba QUIT en el indicador de
mandatos. Fuera de la modalidad interactiva, todos los mandatos deben tener el
prefijo 'db2'.
Para listar los valores actuales de opciones de mandato, escriba LIST COMMAND
OPTIONS.

Para obtener ayuda más detallada, vea el Manual de consulta en línea.

db2 => CREATE DATABASE ejemplo
DB20000I  El mandato CREATE DATABASE ha finalizado satisfactoriamente.
db2 => _
```

Imagen: Ventana con el comando de creación de una base de datos
Fuente: José Santiago Herrera Furlán

3.14 Creación de una base de datos DB2

Por: José Herrera

Con una base de datos creada podemos agregarle una tabla y registros utilizando instrucciones SQL. Para crear la primera tabla, primero debemos conectarnos a nuestra base de datos con la siguiente instrucción:

CONNECT TO ejemplo

```
db2 => CONNECT TO ejemplo

Información de la conexión con la base de datos

Servidor bases datos   = DB2/NT64 11.5.8.0
ID autorizaci3n SQL    = USER
Alias base datos local = EJEMPLO

db2 =>
```

Imagen: Conexión con una base de datos en DB2
Fuente: José Santiago Herrera Furlán

A continuación, crearemos la tabla, para ello utilizamos una instrucción SQL que se crea con la siguiente sintaxis:

```
CREATE TABLE NombreDeTabla (
    NombreDeCampo1 TipoDeDato,
    NombreDeCampo2 TipoDeDato,
    NombreDeCampo3 TipoDeDato)
```

La instrucción real quedaría tal que así:

```
CREATE TABLE persona (
    id int primary key not null,
    nombre varchar(25) not null,
    grado varchar(25) not null)
```

Esta instrucción crea una tabla llamada persona con 3 campos: id (llave primaria), que es un entero; nombre, que es una cadena de 25 caracteres; y grado, que es otra cadena de 25 caracteres. Nótese que en DB2 debemos escribir la instrucción en una sola línea, ya que no podemos dar saltos de línea porque la tecla "enter" envía la instrucción. Podemos ver la tabla usando la siguiente instrucción:

```
SELECT * FROM persona
```

```
db2 => CREATE TABLE persona (id int primary key not null, nombre varchar(25) not
null, grado varchar(25) not null)
DB20000I El mandato SQL ha finalizado satisfactoriamente.
db2 => SELECT * FROM persona

ID              NOMBRE              GRADO
-----
0 registro(s) seleccionado(s).

db2 =>
```

Imagen: Creación y visualización de una tabla en DB2
Fuente: José Santiago Herrera Furlán

Por último, ingresaremos registros a la tabla, esto se hace utilizando la siguiente sintaxis:

```
INSERT INTO NombreDeTabla VALUES (ValorDelCampo1, ValorDelCampo2,
ValorDelCampo3)
```

La instrucción real quedaría así:

```
INSERT INTO personas VALUES (1, 'Santiago', '6to Computacion')
```

Con esta instrucción ingresamos un registro de id igual a 1, nombre igual a Santiago, y grado igual a 6to Computacion. Nótese cómo los valores que son cadenas de caracteres se escriben entre comillas, mientras que el valor numérico no las lleva. Al terminar el ingreso del registro podemos revisar nuestra tabla de nuevo utilizando la instrucción:

SELECT * FROM persona

```
db2 => INSERT INTO persona VALUES (1, 'Santiago', '6to Computacion')
DB20000I El mandato SQL ha finalizado satisfactoriamente.
db2 => SELECT * FROM persona

ID          NOMBRE          GRADO
-----
          1 Santiago          6to Computacion

1 registro(s) seleccionado(s).

db2 =>
```

Imagen: Ingreso y visualización de un registro en DB2
Fuente: José Santiago Herrera Furlán

3.15 Tipos de dato DB2

Por: José Herrera

En DB2 hay varios tipos de dato que principalmente se pueden clasificar en: datos numéricos, de caracteres, binarios, de fecha y hora, y booleanos. A su vez, dentro de estos tipos de dato pueden existir aquellos que almacenan datos muy extensos, estos son llamados objetos grandes que se caracterizan por terminar con las siglas "LOB".

Los datos **numéricos** almacenan cualquier número y se clasifican en:

- **SMALLINT**: Números enteros de 2 bytes (del -32,768 a 32,767).
- **INTEGER**: Números enteros de 4 bytes (del -2,147,483,648 a 2,147,483,647).
- **BIGINT**: Números enteros de 8 bytes (del aprox. -9 billones a 9 billones).
- **DECIMAL o NUMERIC**: Números decimales con precisión de hasta 31 dígitos.
- **REAL**: Números decimales con precisión de hasta 8 dígitos.
- **DOUBLE o FLOAT**: Números decimales con precisión de hasta 16 dígitos.
- **DECFLOAT**: Números decimales de hasta 34 dígitos. Permite almacenar los valores abstractos del infinito y NaN (valores indefinidos, como un número dividido entre 0).

Los datos **de caracteres** almacenan texto y se clasifican en:

- **CHAR**: Cadena de caracteres de longitud fija, de hasta 255 caracteres.
- **VARCHAR**: Cadena de caracteres de longitud variable, de hasta 32,672 caracteres.
- **CLOB (Objeto grande)**: Cadena de caracteres de hasta 2 mil millones de caracteres.

Los datos **binarios** se utilizan para guardar secuencias de bytes, con el objetivo de guardar imágenes, audio, etc. Se clasifican en:

- **BINARY**: Almacena hasta 255 bytes.
- **VARBINARY**: Variable, de hasta 32,672 bytes de longitud
- **BLOB (Objeto grande)**: Puede contener hasta 2 mil millones de bytes.

Los datos de **fecha y hora** almacenan ya sea una hora, una fecha o ambos. Se clasifican en:

- **DATE**: Contiene una fecha, es decir, año, mes y día.
- **TIME**: Contiene una hora, sus minutos y sus segundos.
- **TIMESTAMP**: Contiene el año, mes, día, hora, minutos, segundos y fracciones de segundo.

Los valores **booleanos** son valores lógicos que sólo pueden tener dos valores: verdadero (*TRUE*) o falso (*FALSE*). Sólo existe un tipo:

- **BOOLEAN**: Almacena los valores booleanos.

3.16 Conexión a una base de datos DB2

Por: Jose Carlos Jordan Cardona

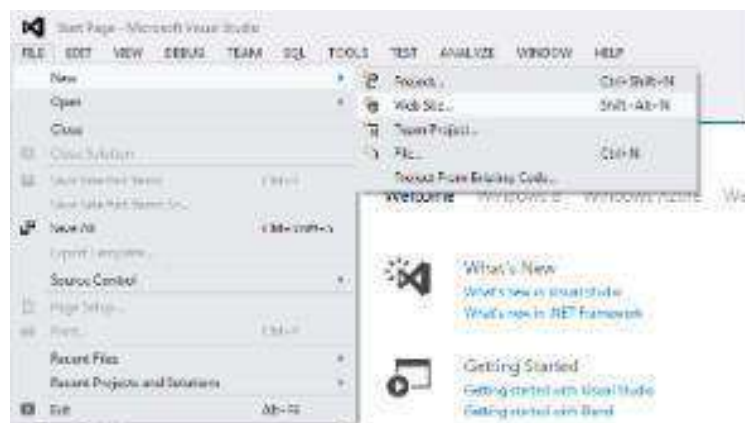
Introducción

En la actualidad los lenguajes de programación no traen incorporada una base de datos, por tal motivo es necesario conectar el lenguaje de programación con una base de datos, un requisito importante para crear esta conexión es usar un string de conexión, en esta guía desarrollaremos un progreso para conectar un lenguaje de programación (ASP.NET con Visual Basic) y una base de datos relacional (SQL Server).

Elaboración de un ejemplo.

-Crear la base de datos ejemplo, la cual puedes descargar del siguiente enlace(Fuente)

-Abrimos el "Visual Studio 2012" y vamos a crear un sitio web, para elaborar nuestro ejemplo, asi como lo muestra la siguiente figura:

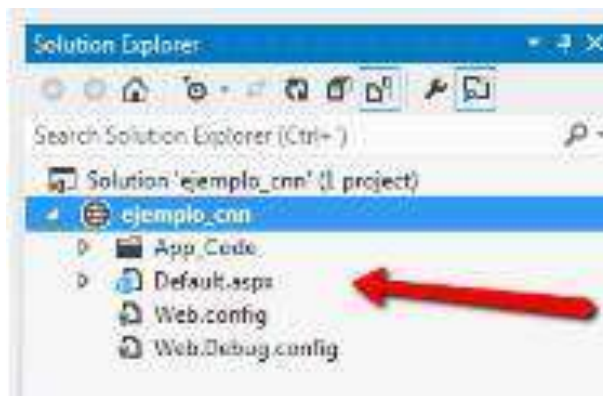


Fuente: <https://silo.tips/download/11-conexion-a-base-de-datos>



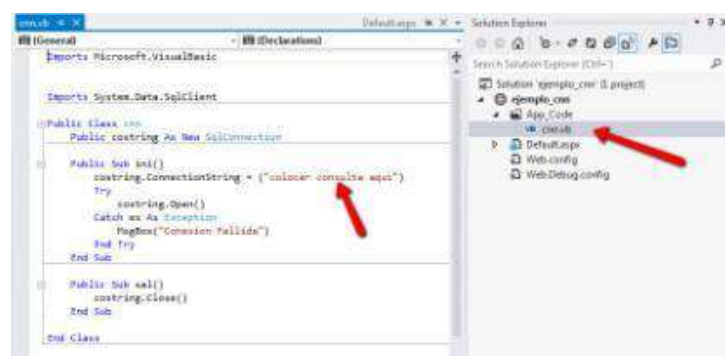
Fuente: <https://silو.tips/download/11-conexion-a-base-de-datos>

Vamos a crear un sitio web llamado “ejemplo_cnn” vamos a usar “Visual Basic” y el “.NET Framework 4”, presionamos ok. Ahora colocaremos el contenido de la carpeta “ejemplo_cnn” que está ubicada en la siguiente dirección (Fuente) agregaremos los archivos al sitio quedando de la siguiente manera:



Fuente: <https://silو.tips/download/11-conexion-a-base-de-datos>

Ahora expandiremos la carpeta “App_Code” y abriremos el archivo “cnn.vb” el cual tiene la siguiente estructura:



3.17 Manipulación de datos

Por: Jose Carlos Jordan Cardona

Puede utilizar las sentencias **INSERT, UPDATE, MERGE, TRUNCATE y DELETE** para manipular los datos de Db2. Puede utilizar sentencias SQL para crear, modificar o descartar objetos Db2, utilizando varias sentencias CREATE, ALTER y DROP. Todos los programas SQL se ejecutan como parte de un proceso de aplicación.

```
MERGE Locations T
USING Locations_stage S ON T.LocationID=S.LocationID
WHEN MATCHED AND T.LocationID =3 THEN
DELETE
WHEN MATCHED AND T.LocationID =1 THEN
UPDATE SET LocationName=S.LocationName;

select * from Locations
```

100 %

Results Messages

	LocationID	LocationName
1	1	Richmond Cross
2	2	Brigade Road

Fuente: <https://www.sqlshack.com/sql-server-merge-statement-overview-and-examples/>

3.18 Restricción de DB2

Por: Jose Carlos Jordan Cardona

Una *restricción de comprobación* es una regla que especifica los valores permitidos en una o más columnas de cada fila de una tabla base. Al igual que las restricciones de referencia, las restricciones de comprobación son opcionales y se definen mediante las sentencias **CREATE TABLE** y **ALTER TABLE**.

La definición de una restricción de comprobación limita los valores que una columna específica de una tabla base puede contener. Una tabla puede tener tantas restricciones de comprobación como se desee. Db2 impone una restricción de comprobación aplicando la restricción a cada fila que se inserta, se carga o se actualiza.

Una restricción es que un nombre de columna de una restricción de comprobación de una tabla debe identificar una columna de dicha tabla.

Por ejemplo, puede crear una restricción de comprobación para asegurarse de que todos los empleados ganan un salario de 30.000 \$ o más:

```
CHECK (SALARY >= 30000)
```

ANÁLISIS DE
SISTEMAS
Bimestre IV



4. Windows Server Conceptos

DDP-ITKer Rodolfo Rojas

Windows Server es una distribución de Microsoft para el uso de servidores. Está desarrollado en el **lenguaje de programación C++ y Asembler**. Se trata de un sistema multiproceso y multiusuario que a día de hoy utilizan millones de empresas de todo el mundo gracias a las características y ventajas que ofrece.

1. Qué es un Windows Server

La primera versión del sistema fue Windows 2000 Server, lanzada a principios del nuevo milenio. Fue concebida para ser el servidor de archivos, impresión y web de PYMEs. Una solución extraordinaria para cuando no era necesario contar con un servidor dedicado a cada tarea, pudiendo así tener todo centralizado en un único servidor. Era capaz de soportar hasta cuatro procesadores.

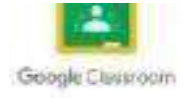


<https://www.informaticalescala.com/es/problema-de-seguridad-critico-detectado-en-sistemas-operativos-windows-ms14-066/>

<https://www.informaticalescala.com/es/problema-de-seguridad-critico-detectado-en-sistemas-operativos-windows-ms14-066/>

1.1 Funciones principales de Windows Server

- **Virtualización de servidores:** Windows Server ofrece a compañías de todo el mundo las herramientas que necesitan para crear y administrar un entorno de servidor virtualizado. Así, tienen la posibilidad de replicar una determinada máquina virtual de un lugar a otro, sin necesidad de almacenamiento compartido.
- **Gestión del servidor:** Server Core y Nano Server ofrecen un entorno mínimo para la ejecución de diversas funciones específicas de servidor.
- **Acceso Directo:** una de las principales características que definen a Windows Server es la conocida como "túnel". Permite el acceso a la red desde cualquier equipo informático. Aporta mayor rapidez que las conexiones VPN tradicionales.



2. Windows Server 2012

Windows Server 2012 R2 es una plataforma de centro de datos y nube de clase empresarial probada que puede escalarse para ejecutar las clases de trabajo más grandes a la vez que activa opciones de recuperación sólidas para protegerse contra interrupciones del servicio. Permite acelerar el tiempo de creación de valor mediante la simplificación de la infraestructura subyacente y la reducción del coste, aprovechando el hardware estándar del sector. Windows Server 2012 R2 te ayuda a crear, desplegar y escalar aplicaciones y sitios web rápidamente, y ofrece flexibilidad para trasladar cargas de trabajo entre los entornos local y en la nube. Permite proporcionar un acceso flexible remoto a los recursos corporativos, mediante la gestión de identidades en el centro de datos y su federación en la nube, y proteger la información crítica de la empresa.



Fuente: <https://www.software-reuse.eu/en/software-kaufen/windows-server-2012-r2/>

3. Diseños de la Red

El diseño de red es una categoría del desarrollo de sistemas que permite determinar la estructura física de una red de computadoras. Se refiere a la planificación de la implementación de una infraestructura de red informática.

El diseño de red implica evaluar, comprender y determinar el alcance de la red que se va a implementar. El diseño completo de una red generalmente se representa como un diagrama que sirve como modelo que orienta a los expertos al momento de poner en marcha la red físicamente.

3.1 ¿Cómo se hace el diseño de una red?

De acuerdo con el artículo *How to Design a Network + Best Network Design Tools* publicado en 2020 en el sitio de **DNSstuff**, una compañía que se enfoca en proporcionar revisiones de software, consejos para la resolución de problemas, mejores prácticas para herramientas y listas completas de los mejores software de negocios, todo diseño de red inicia con el modelo OSI, el cual se conforma de siete capas.



3. Diseño de una red

Por: **Ronaldo Ruano**

La creación inicial de una red y la instalación de Windows Server 2012 y un servidor DHCP son tres temas críticos en el diseño y la implementación de una infraestructura de red eficiente y sólida. En esta investigación formal, se explorarán estos temas en profundidad para brindar una comprensión completa de su importancia y sus implicaciones en la administración de una red.

4. Instalación de una Red

La creación inicial de una red es un proceso crucial que determina el éxito de la infraestructura de la red en el futuro. Es importante tener en cuenta aspectos como la topología de la red, la selección de los equipos y software necesarios, y la definición de los protocolos de red a utilizar. La planificación cuidadosa y la implementación adecuada de la creación inicial de una red asegurarán una infraestructura sólida y eficiente que satisfaga las necesidades actuales y futuras de la organización.

5. Instalación de Windows Server 2012

- a. Los siguientes son los pasos generales para la instalación de Windows Server 2012:
- b. Preparación del hardware: Antes de instalar el sistema operativo, asegurarse de que el hardware cumpla con los requisitos mínimos y que esté configurado correctamente.
- c. Descargar la ISO de Windows Server 2012: Se puede descargar la imagen ISO de la página web de Microsoft.
- d. Crear un medio de instalación: La imagen ISO de Windows Server 2012 debe ser grabada en un DVD o copiada en un dispositivo USB para su uso como medio de instalación.
- e. Iniciar la instalación: Reiniciar la computadora y arrancar desde el medio de instalación para iniciar la instalación de Windows Server 2012.
- f. Seleccionar la versión de Windows Server 2012: Seleccionar la versión de Windows Server 2012 que se desea instalar (Standard, Datacenter, etc.).
- g. Aceptar los términos de licencia: Aceptar los términos de licencia de Microsoft antes de continuar con la instalación.
- h. Seleccionar la partición: Seleccionar la partición en la que se desea instalar el sistema operativo y, si es necesario, crear una nueva partición.
- i. Instalación del sistema operativo: El proceso de instalación comenzará y se completará en unos pocos minutos.
- j. Configuración inicial: Una

vez que la instalación se complete, se deben realizar la configuración inicial, incluida la definición de un nombre para el servidor y la configuración de una contraseña para el administrador.



Windows Server 2012 – Ediciones Datacenter y Standard:

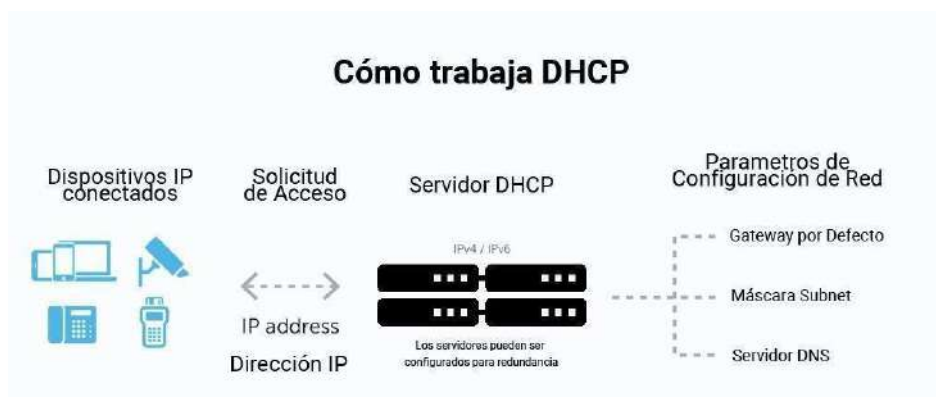
<https://www.internetya.co/windows-server-2012-ediciones-datacenter-y-standard/>

6. Instalación de un servidor DHCP:

Un servidor DHCP (Dynamic Host Configuration Protocol) es un componente crítico en la administración de una red. La función de un servidor DHCP es automatizar la asignación de direcciones IP a los dispositivos en la red. La instalación de un servidor DHCP asegura que cada dispositivo en la red tenga una dirección IP única y reduce el riesgo de errores humanos al asignar manualmente las direcciones IP.

Los siguientes son los pasos generales para la instalación de un servidor DHCP:

- a. Verificar requisitos del sistema: Verificar que el hardware y el sistema operativo cumplan con los requisitos mínimos para ejecutar un servidor DHCP.
- b. Instalar el servicio DHCP: En Windows Server 2012, el servicio DHCP se puede instalar desde el Administrador del Servidor.
- c. Configurar el servidor DHCP: Una vez instalado el servicio DHCP, se deben configurar las opciones necesarias, como la asignación de direcciones IP, el rango de direcciones IP disponibles, etc.
- d. Crear reglas de reserva: Es posible crear reglas de reserva de direcciones IP para asignar direcciones IP específicas a dispositivos específicos en la red.
- e. Habilitar el servicio DHCP: Una vez configurado el servidor DHCP, se debe habilitar el servicio para que pueda proporcionar direcciones IP a los dispositivos de la red.
- f. Prueba y monitoreo: Verificar que el servidor DHCP esté funcionando correctamente y monitorear su rendimiento periódicamente.
- g. Estos son los pasos generales para la instalación de un servidor DHCP. Algunos detalles pueden variar en función de la configuración específica de la red y del sistema operativo utilizado.



¿Qué es DHCP y cómo funciona?

<https://www.infoteknico.com/que-es-dhcp-y-como-funciona/>

7. Configuración de DHCP

Por: Gabriela Santos

El Protocolo de configuración dinámica de host (DHCP) es un protocolo cliente/servidor que proporciona automáticamente un host de Protocolo de Internet (IP) con su dirección IP y otra información de configuración relacionada, como la máscara de subred y la puerta de enlace predeterminada.

1. Selecciona Inicio y, a continuación, escribe configuración.
2. Selecciona Configuración > Red & Internet.
3. Sigue uno de estos procedimientos:
 - Para obtener una red Wi-Fi, selecciona Wi-Fi > Administrar redes conocidas. Elige la red para la que quieres cambiar la configuración.
 - Para una red Ethernet, selecciona Ethernet y, a continuación, elige la red Ethernet a la que estás conectado.
4. Junto a Asignación IP, seleccione Editar.
5. En Editar la configuración IP de red o Editar configuración IP, selecciona Automático (DHCP) o Manual.
 - Especificar manualmente la configuración de IPv4
 - i. En Editar la configuración IP de red o Editar configuración IP, elige Manual y, a continuación, activa IPv4.
 - ii. Para especificar una dirección IP, en los cuadros Dirección IP, Máscara de subred y Puerta de enlace, escriba la configuración de la dirección IP.
 - iii. Para especificar una dirección del servidor DNS, en los cuadros DNS preferido y DNS alternativo, escribe las direcciones de los servidores DNS principales y secundarios.
 - iv. Para especificar si desea usar una conexión cifrada (DNS a través de HTTPS) o no cifrada con el servidor DNS o los servidores que especifique, para DNS a través de HTTPS, elija la configuración que desee:
 - Desactivado: todas las consultas DNS se enviarán al servidor DNS sincifrar en texto sin formato a través de HTTP.
 - Activado (plantilla automática): las consultas DNS se cifrarán y se enviarán al servidor DNS a través de HTTPS. Las consultas DNS usarán la configuración predeterminada para la plantilla automática o intentarán descubrirlas automáticamente.
 - Activado (plantilla manual): las consultas DNS se cifrarán y se enviarán al servidor DNS a través de HTTPS. Usarán la configuración que especifique en el cuadro de plantilla DNS sobre HTTPS.
 - v. Si usa DNS sobre HTTPS (plantilla automática o manual), active o desactive Fallback a texto sin formato:
 - Cuando está activada, se enviará una consulta DNS sin cifrar si no se puede enviar a través de HTTPS.

- Cuando está desactivada, no se enviará una consulta DNS si no se puede enviar a través de HTTPS.

6. Cuando termine, selecciona Guardar.

8. Instalación de DNS

1. Abra Administrador del servidor y haga clic en Agregar roles y características.
2. En el Asistente para agregar roles, si aparece la página Antes de comenzar, haga clic en Siguiente.
3. En la pantalla Tipo de instalación, seleccione Instalación basada en características o basada en roles y haga clic en Siguiente.
4. En la pantalla Selección del servidor, seleccione el servidor y haga clic en Siguiente.
5. En la pantalla Roles de servidor, seleccione Servidor DNS, si se le pide, haga clic en Agregar características y haga clic en Siguiente.
6. En la pantalla Características, haga clic en Siguiente.
7. Lea la información de la página Servidor DNS y, a continuación, haga clic en Siguiente.
8. En la página Confirmación, compruebe que se instalará el rol servidor DNS y, a continuación, haga clic en Instalar.

9. Configurar DNS

1. Abra Administrador del servidor, haga clic en Herramientas y haga clic en DNS.
2. Cree zonas DNS para los mismos nombres de dominio DNS hospedados en los servidores DNS antes de que se produzca un mal funcionamiento crítico. Para obtener más información, vea Agregar una zona de búsqueda directa.
3. Configure los datos DNS tal como existían antes del mal funcionamiento crítico. Por ejemplo:
 - Configure las zonas DNS que se almacenarán en AD DS. Para obtener más información, vea Cambiar el tipo de zona.
 - Configure la zona DNS que es autoritativa para los registros de recursos de localizador de controladores de dominio (localizador de controlador de dominio) para permitir una actualización dinámica segura. Para obtener más



información, consulte Allow Only Secure Dynamic Novedades.



4. Asegúrese de que la zona DNS primaria contiene registros de recursos de delegación (servidor de nombres (NS) y registros de recursos del host de pegado (A) para la zona secundaria hospedada en este servidor DNS. Para obtener más información, vea Crear una delegación de zona.

5. Después de configurar DNS, puede acelerar el registro de los registros de NETLOGON.

6. En el símbolo del sistema, escriba el siguiente comando y presione ENTRAR:

net stop netlogon

7. Escriba el siguiente comando y presione ENTRAR:

net start netlogon

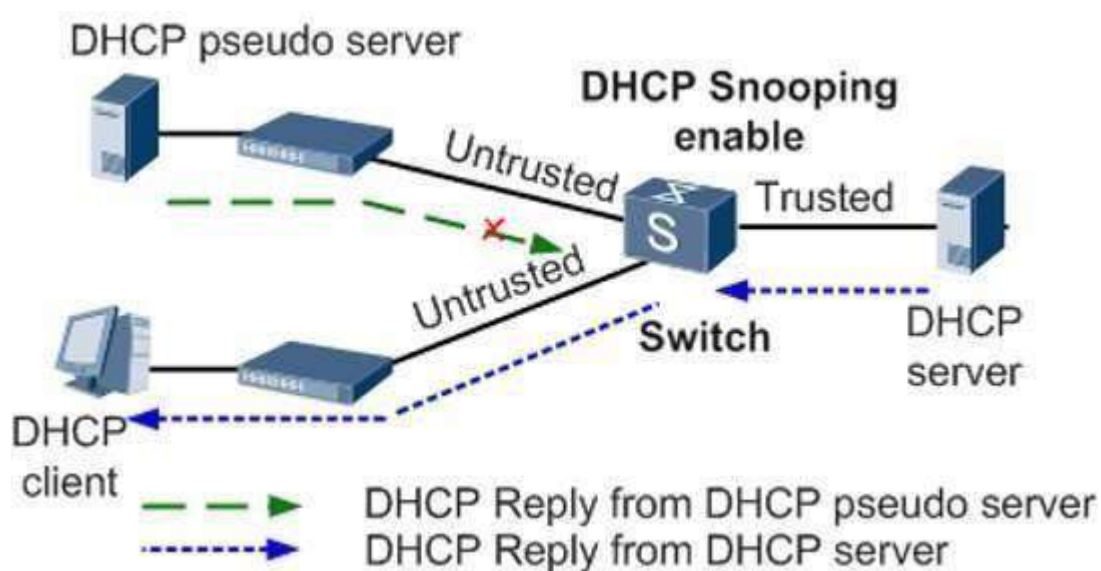


Imagen: Protocolo DHCP.

Fuente: <https://www.redeszone.net/tutoriales/internet/que-es-protocolo-dhcp/>

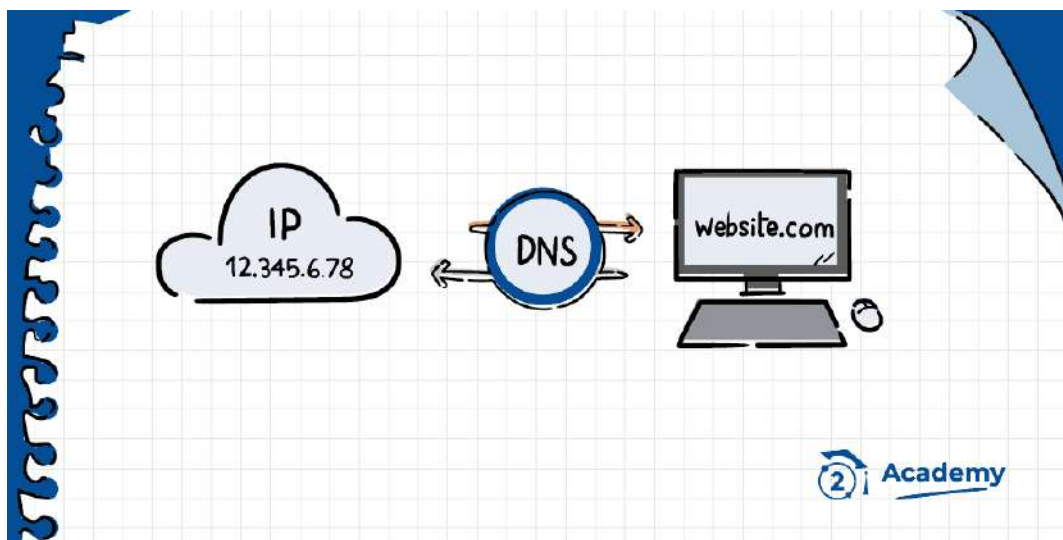


Imagen: Protocolo DNS.

Fuente: <https://academy.bit2me.com/que-es-el-servidor-dns/>

10. Inserción de clientes al dominio de Windows Server 2012

Por: Dilan Sarceño

Un cliente equipado con un sistema operativo Windows 7 se une a un dominio de forma casi idéntica a como lo hace un equipo que incorpora Windows 8.1. Sin embargo, existe algún pequeño matiz que merece que le dediquemos hoy nuestra atención a este proceso.

Básicamente, el trabajo consistirá en establecer las características de red, para que coincidan con las necesidades del dominio, ajustar el nombre del equipo cliente, unir el equipo al dominio y, finalmente, iniciar sesión utilizando una cuenta de usuario de las que ya tenemos definidas en el dominio.

11. Configurar cliente

2. Para asegurarnos de que la configuración de la red del equipo cliente coincide con las necesidades de conexión del dominio, hacemos clic en el botón Inicio y, cuando aparezca el menú, elegimos Panel de control.

2. La ventana del Panel de Control contiene las diferentes herramientas de configuración de Windows 7. Hacemos clic sobre Redes e Internet.

3. Las opciones de la ventana cambiarán para ofrecernos las herramientas solicitadas. Elegimos Centro de redes y recursos compartidos.

4. Aquí disponemos de multitud de opciones de configuración de la red. En particular, a la izquierda encontramos la opción Cambiar configuración del adaptador. Hacemos clic en esta.

5. En la ventana Conexiones de red encontramos un elemento llamado Conexión de área local. Hacemos clic con el botón derecho del ratón sobre él y, en el menú de contexto que aparece elegimos Propiedades.

6. A continuación, veremos que se abre una ventana con las Propiedades de la conexión de área local. En este momento nos debemos plantear qué versión del protocolo de Internet utiliza nuestra red. En mi caso, utilizo la versión 4, por lo que deshabilitaré la versión 6, con el objetivo de ahorrar recursos en el sistema. Después, selecciono la versión 4 y hago clic en Propiedades para configurarla.

7. En la ventana que aparece, me aseguro de que se encuentra marcada la opción Obtener una dirección IP automáticamente. Lo cierto es que, si no dispones de servidor DHCP en la red, también podría asignar las direcciones IP manualmente, pero creo que hoy en día, esta situación es bastante infrecuente. Otra cuestión que debemos tener en cuenta en este punto es que la dirección IP del servidor DNS preferido debe coincidir con la del servidor DNS de nuestra red. En nuestro caso, lo instalamos a la vez que el Dominio, por lo que la dirección IP será la del propio

controlador de dominio, que debe estar asignada manualmente y ser fija.
Una vez establecidos los valores, hacemos clic sobre el botón Aceptar.



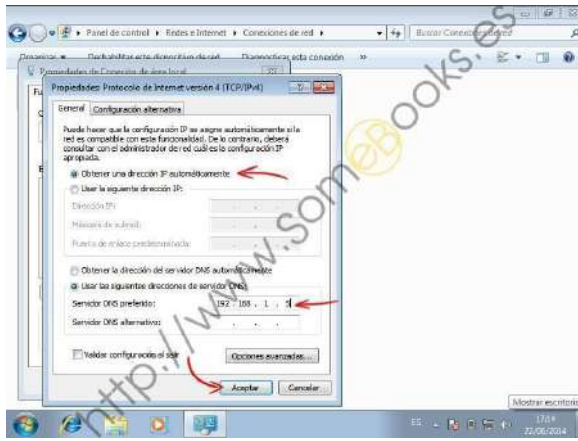


Imagen: Configuración de cliente
Fuente: <http://www.someBooks.es>

1.1 Comprobar que la configuración de red es correcta

1. Después de cerrar todas las ventanas, debemos comprobar que no hemos cometido ningún error. Esto es tan sencillo como abrir una ventana de comandos y hacer un ping al Servidor. Si responde, la configuración es correcta. Si no lo hace, deberemos repasar nuestras últimas acciones. Hacemos clic sobre el botón Inicio y, en el cuadro Buscar programas y archivos, escribimos el texto cmd.

2. Después, se abrirá la ventana de comandos. Escribimos la orden ping server-2012-a (que es como se llama el servidor) y pulsamos la tecla Enter. Si el servidor responde, la salida será como en la imagen.

Imagen: verificación de red
Fuente: <http://www.someBooks.es>



1.2 Cambiar el nombre del equipo y unirlo al dominio

1. El segundo y el tercer paso se harán seguidos y consistirán en ajustar el nombre del equipo cliente, para que coincida con el que debe tener en el dominio. Ya sabes que, si la cuenta existe previamente en el dominio, al unir el cliente se establecerá un vínculo entre ambos. Si no existe, se crea una nueva cuenta de equipo en ese instante. Volvemos a hacer clic sobre el botón Inicio y después sobre Equipo, pero esta segunda vez, con el botón derecho del ratón. En el menú de contexto que aparece, elegimos Propiedades.

2. Aparece una ventana con información del sistema. En la parte inferior, además de algunos detalles sobre el hardware del equipo, podemos encontrar su nombre (tanto el nombre NetBIOS como el nombre DNS), la descripción y el nombre del grupo de trabajo o dominio al que

pertenece. Para cambiar estos datos, sólo tenemos que hacer clic en el enlace [Cambiar configuración](#).



3. En la solapa Nombre de equipo de la ventana Propiedades del sistema, que aparece a continuación, podríamos escribir un texto para la descripción (si te fijas, en la imagen anterior aparecía en blanco). Sin embargo, lo dejaremos así. Lo que sí haremos será un clic sobre el botón Cambiar, para escribir el nuevo nombre del equipo y nombre del dominio.

4. Cuando los datos sean correctos, hacemos clic en Aceptar.

5. En ese momento, Windows 7 busca en la red el dominio especificado. Si no lo encuentra, aparecerá un mensaje de aviso (probablemente habremos cometido algún error en los datos introducidos). Si lo encuentra, deberemos escribir un nombre de usuario y una contraseña, perteneciente al dominio, que tenga privilegios suficientes para unir el equipo cliente (observa que, aunque en la imagen anterior hemos usado el nombre DNS, en la ventana de autenticación de la imagen siguiente aparece el nombre NetBIOS. Una vez introducidos los datos, hacemos clic en Aceptar.

6. La ventana de autenticación se cierra y en su lugar aparece un mensaje indicando que el equipo se ha unido correctamente al dominio. Si cometemos algún error en el nombre de usuario o en la contraseña, en lugar del mensaje siguiente, aparecerá uno de error y tendremos que volver a intentarlo. Hacemos clic sobre Aceptar.

7. A continuación, aparece una nueva ventana informativa indicando que deberemos reiniciar el equipo para que se apliquen los cambios, pero que antes deberemos cerrar todos los programas y guardar todos los archivos que tengamos abiertos. Hacemos clic en Aceptar.

8. Cuando cerremos la ventana de Propiedades del sistema, detectará que ya no hay programas en ejecución y nos dará la oportunidad de reiniciar en ese momento. Hacemos clic en Reiniciar ahora.

9. Para iniciar sesión con un usuario diferente, hacemos clic sobre Cambiar de usuario. Y ya podemos terminar.

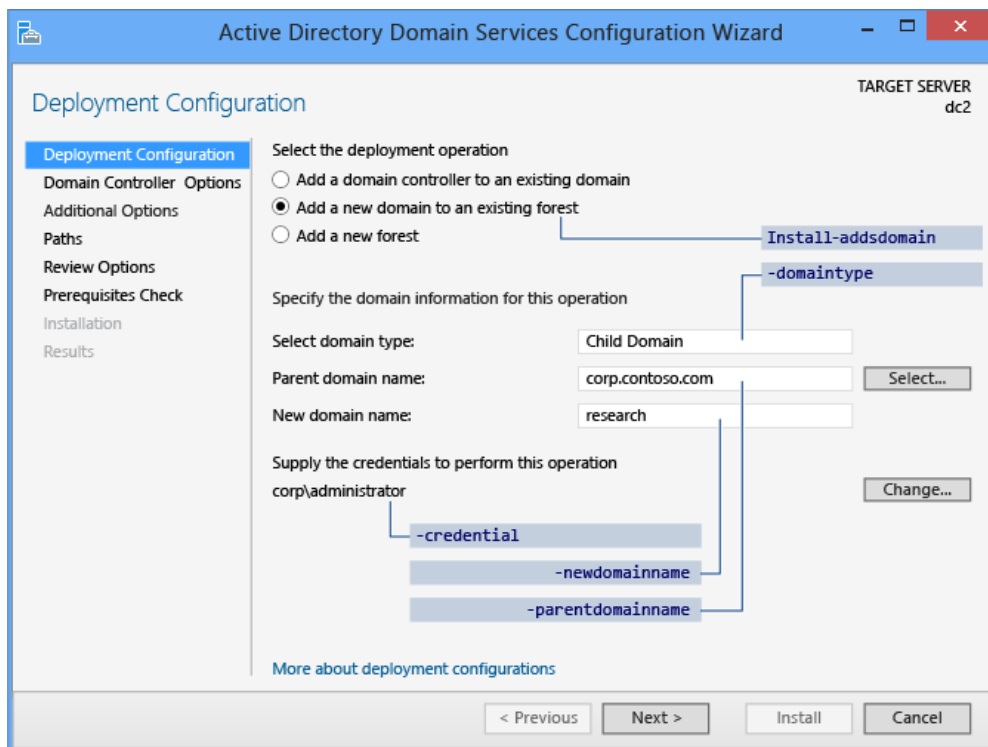


Imagen: Cambiar de usuario en el SO. Fuente: <http://www.someBooks.es>

13. Instalación ADDS en subdominio

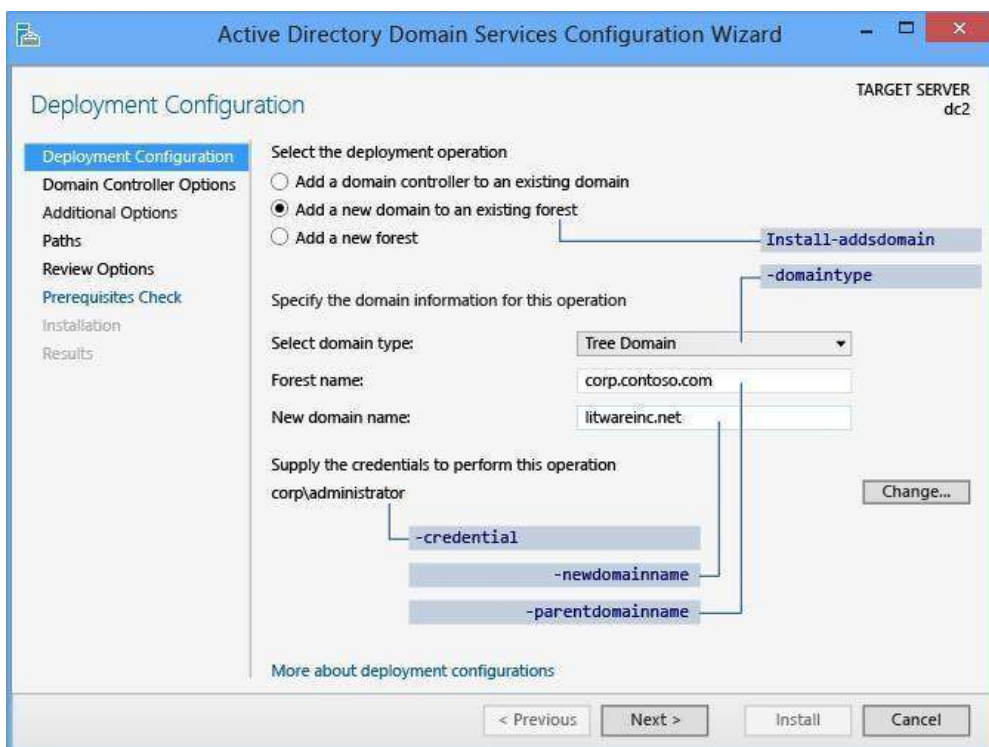
Por: Dulce María Shuyá Baten

- En la siguiente captura de pantalla verás las opciones para agregar un dominio secundario:



https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/deploy/media/install-a-new-windows-server-2012-active-directory-child-or-tree-domain--level-200-/adds_smi_tr_childdeployconfig.png

- En la siguiente captura de pantalla verás las opciones para agregar un dominio de árbol:



https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/deploy/media/install-a-new-windows-server-2012-active-directory-child-or-tree-domain--level-200-/adds_smi_tr_childdeployconfig.png

[/adds_smi_tr_treedeployconfig.png](#)

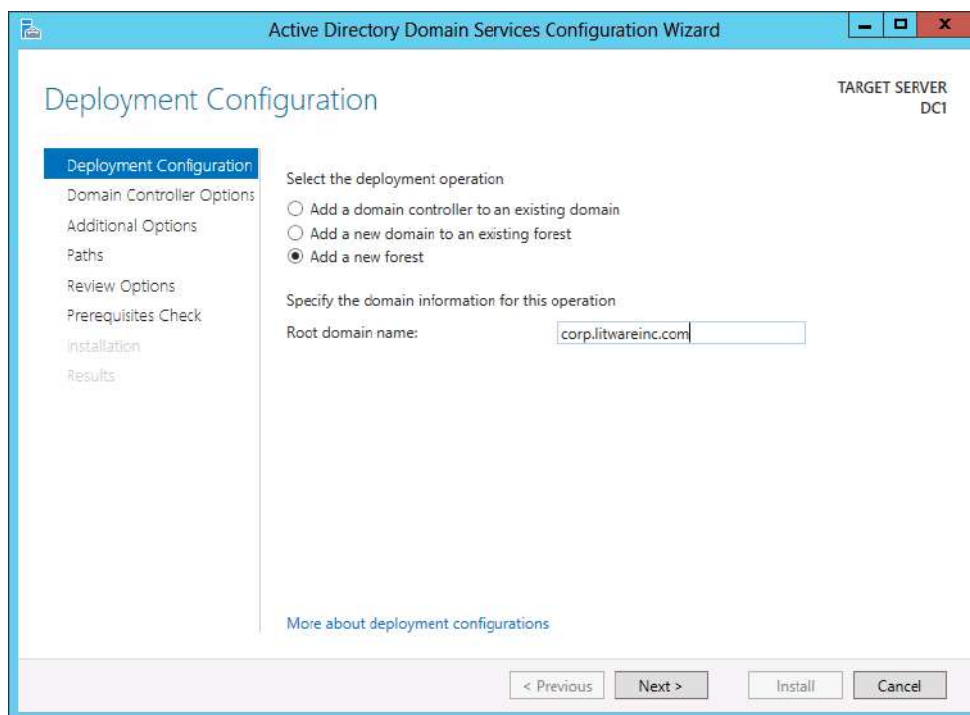
- Las opciones restantes y los campos requeridos cambian en esta página y en las páginas siguientes, según qué operación de implementación se seleccione.

- En este tema se combinan dos operaciones discretas: la promoción de dominios secundarios y la promoción de dominios de árbol. La única diferencia entre las dos operaciones es el tipo de dominio que crees.
- Para crear un nuevo dominio secundario, haz clic en Agregar un nuevo dominio a un bosque existente y selecciona Dominio secundario. Para un nombre de dominio primario, escribe o selecciona el nombre del dominio primario. A continuación, escribe el nombre del nuevo dominio en el cuadro Nuevo nombre de dominio. Especifica un nombre de dominio secundario de etiqueta única y válido; el nombre debe cumplir los requisitos de nombre de dominio DNS.
- Para crear un nuevo dominio de árbol, haz clic en Agregar un nuevo dominio a un bosque existente y selecciona Dominio de árbol. Escribe el nombre del dominio raíz del bosque y después el nombre del nuevo dominio. Especifica un nombre de dominio raíz completo y válido; el nombre no puede ser de etiqueta única y debe cumplir los requisitos de nombre de dominio DNS.

14. Configuración ADDS para subdominio

El Administrador del servidor comienza todas las instalaciones de controlador de dominio con la página Configuración de implementación. Por ejemplo, si crea un bosque, la página Opciones de preparación no aparece, pero sí si instala el primer controlador de dominio que ejecuta Windows Server 2012 en un bosque o dominio existente.

Se realizan algunas pruebas de validación en esta página. Por ejemplo, si intenta instalar el primer controlador de dominio de Windows Server 2012 en un bosque que tiene el nivel funcional de Windows 2000, aparece un error en esta página. Cuando se crea un nuevo bosque aparecen las siguientes opciones.

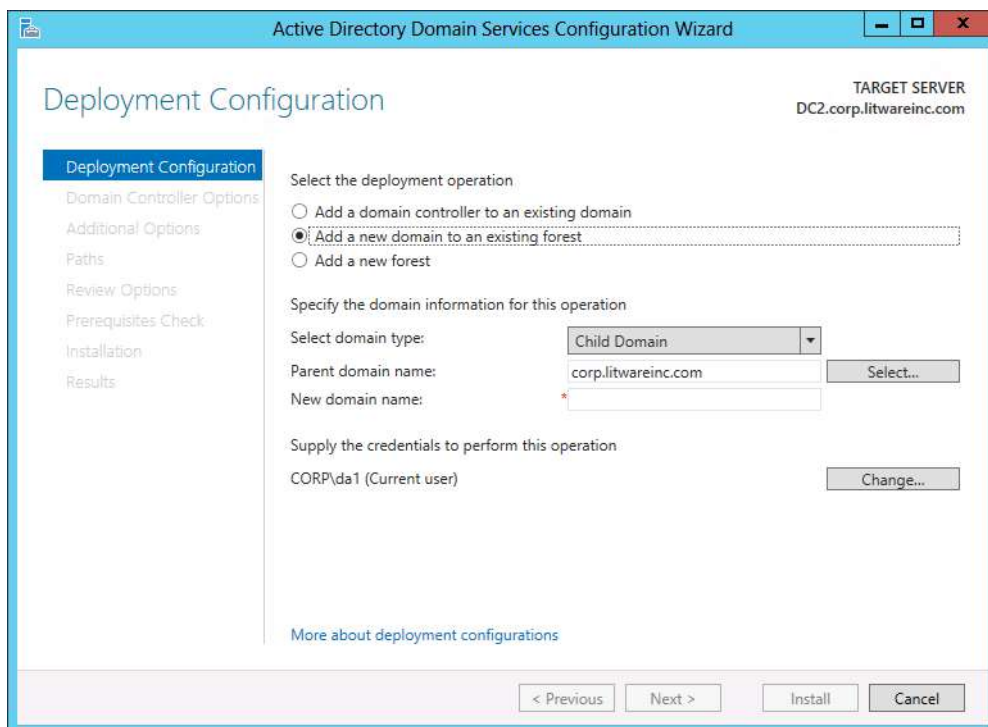


https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/deploy/media/install-a-new-windows-server-2012-active-directory-forest--level-200-/adds_smi_deploymentconfiguration_forest.gif

- Cuando se crea un nuevo bosque, se debe especificar un nombre para el dominio raíz del bosque. El nombre de dominio raíz del bosque no puede etiquetarse de forma única (por ejemplo, debe ser "contoso.com" en lugar de "contoso"). Debe usar convenciones de nomenclatura de dominio DNS permitidas. Se puede especificar un nombre de dominio internacionalizado (IDN).

- No cree bosques nuevos de Active Directory con el mismo nombre que el del DNS externo. Por ejemplo, si la dirección URL de DNS de Internet es <http://contoso.com>, debe elegir otro nombre para el bosque interno para evitar problemas de compatibilidad futuros.

- Debe ser miembro del grupo Administradores en el servidor donde desea crear el nuevo bosque.



Fuente: https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/deploy/media/ad-ds-installation-and-removal-wizard-page-descriptions/adds_smi_deploymentconfiguration_childdomain.gif

- Haga clic en Seleccionar para buscar el dominio primario o el árbol de Active Directory, o escriba un dominio primario o nombre de árbol válido
- Dominio de árbol: proporcione un nombre de dominio raíz completo y válido.
- Dominio secundario: proporcione un nombre de dominio secundario de etiqueta única y válido.
- El Asistente para configuración de Active Directory Domain Services pide confirmación de las credenciales de dominio si las credenciales actuales no corresponden al dominio.
- Las siguientes opciones aparecen cuando se agrega un controlador de dominio nuevo a un dominio existente.

Active Directory Domain Services Configuration Wizard

Deployment Configuration

TARGET SERVER
DC2.corp.litwareinc.com

- Deployment Configuration
- Domain Controller Options
- Additional Options
- Paths
- Review Options
- Prerequisites Check
- Installation
- Results

Select the deployment operation

- Add a domain controller to an existing domain
- Add a new domain to an existing forest
- Add a new forest

Specify the domain information for this operation

Domain:

Supply the credentials to perform this operation

CORP\da1 (Current user)

[More about deployment configurations](#)

< Previous Next > install Cancel

Fuente: https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/deploy/media/ad-ds-installation-and-removal-wizard-page-descriptions/adds_smi_deploymentconfiguration_forest.gif

- Haga clic en Seleccionar para buscar el dominio, o escribe un nombre de dominio válido.
- El Administrador del servidor solicita que escribas credenciales válidas si es necesario. La instalación de un controlador de dominio adicional requiere la pertenencia al grupo Admins del dominio.
- Además, la instalación del primer controlador de dominio que ejecuta Windows Server 2012 en un bosque requiere credenciales que incluyan pertenencias a grupos en los grupos Administradores de empresa y Administradores de esquema.

15. Inserción clientes al subdominio

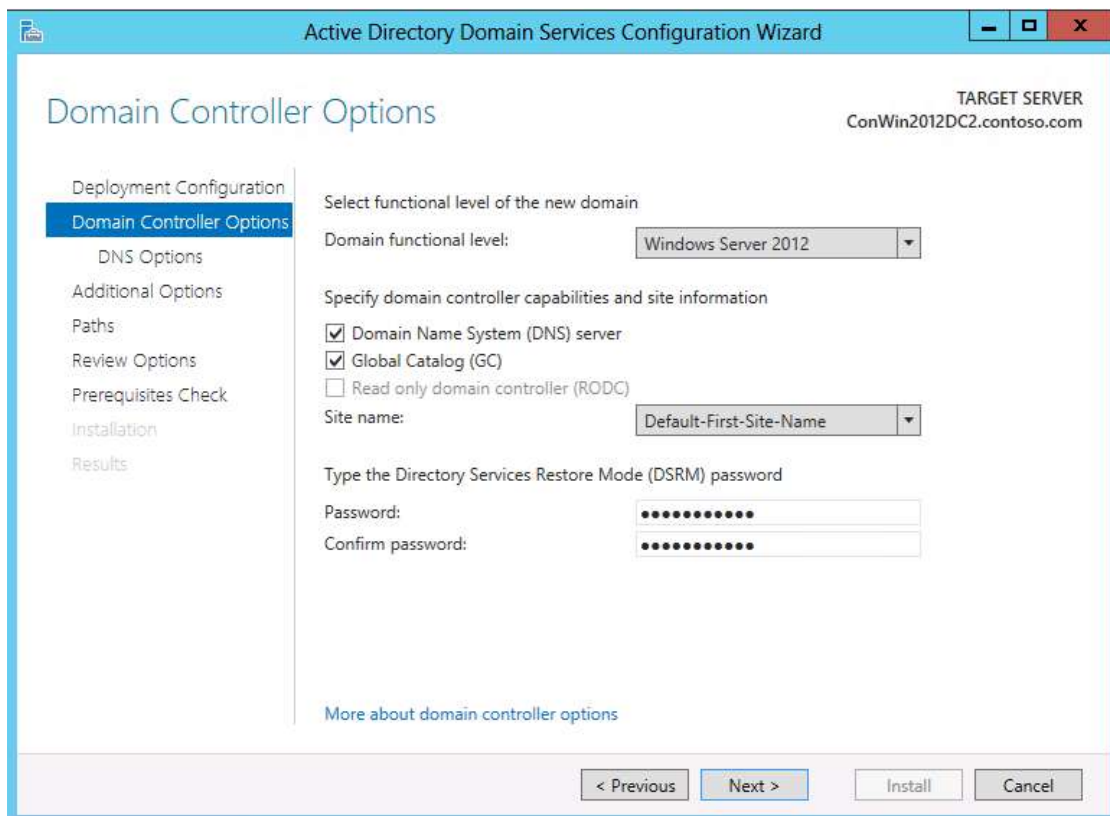
Si está creando un nuevo bosque, la página Opciones del controlador de dominio tiene estas opciones:

En la página Opciones del controlador de dominio se especifican las opciones del controlador de dominio para el nuevo controlador de dominio. Las opciones del controlador de dominio que pueden configurarse incluyen Servidor DNS y Catálogo global.

Microsoft recomienda que todos los controladores de dominio proporcionen servicios de DNS y GC para alta disponibilidad en entornos distribuidos. GC siempre está seleccionado de manera predeterminada, excepto si el dominio actual ya hospeda DNS en sus controladores de dominio basándose en la consulta Inicio de autoridad, en cuyo caso la opción predeterminada será Servidor DNS. También es necesario especificar un Nivel funcional de dominio. El nivel funcional predeterminado es Windows Server 2012 y se puede elegir cualquier otro valor que sea igual o superior al nivel funcional del bosque actual.

La página Opciones del controlador de dominio también permite elegir el nombre de sitio lógico apropiado de Active Directory de la configuración del bosque. De manera predeterminada, se selecciona el sitio con la subred más correcta. Si solo hay un sitio, este se seleccionará automáticamente.

La Contraseña del modo de restauración de servicios de directorio debe cumplir la directiva de contraseñas aplicada al servidor. Siempre elija una contraseña segura y compleja.



https://learn.microsoft.com/es-es/windows-server/identity/ad-ds/deploy/media/ad-ds-installation-and-removal-wizard-page-descriptions/adds_smi_dcoptions_forest.gif

TEORÍA DE INFORMACIÓN BIMESTRE 1

1.1 MecaNet

Por: Angel juarez

MecaNet es un programa para aprender mecanografía que nos ayudará a mecanografiar de forma rápida y con soltura, sin cometer errores y que nos permitirá adquirir una mayor velocidad en la escritura, este programa con voz gratuito para aprender mecanografía en el consta de 20 lecciones y cada lección cuenta con las instrucciones correspondientes sobre la posición y movimiento de los dedos.



Programa gratuito fuente

<https://www.compartolid.es/mecanet/>

Pueden usarlo al mismo tiempo varios usuarios porque permite guardar y cargar los diferentes progresos de cada uno de ellos.

Autor de MecaNet: Carlos Miguel Cáceres García.

Idioma: español

Licencia: Es de uso gratuito

1.1.1 ventajas:

Se trata de una interfaz gráfica con múltiples paneles diseñados para ser usados de forma visual. No resulta operativo para usuarios que precisen de un lector de pantalla.

Permite cierto grado de personalización en las vistas y fuentes de letras.

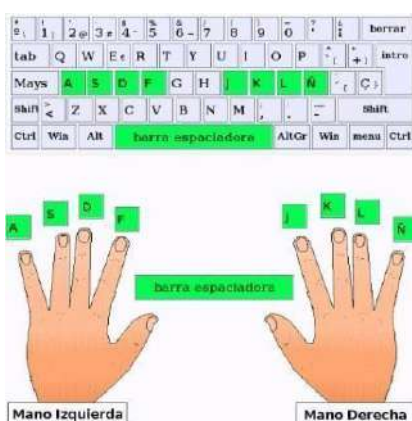
Dispone de las siguientes funciones para el seguimiento de los progresos del alumno.

Permite generar nuevas lecciones para adaptarlas a ciertos textos e incluso crear perfiles o grupos de alumnos.

1.1.2 Requisitos

Sistema Operativo Microsoft Windows 8.1, 8, 7, Vista, ME, XP, 98, 95. (32 y 64 bits). Se dispone de varias versiones para adaptarse mejor a las necesidades del usuario, entre ellas destaca la versión portable que nos permite ejecutarlo en cualquier equipo sin precisar instalar nada en él

La utilidad de la mecanografía La utilidad de las máquinas de escribir es múltiple. Su estudio consiste fundamentalmente en el desarrollo de destrezas y habilidades para operarlas con el propósito de obtener una producción efectiva en tiempo razonable.



Posición correcta de las manos fuente :

[Posición de las manos en el teclado QWERTY – ProfJulio \(grecobcc.org\)](http://grecobcc.org)

1.1.3 ¿Desventajas de MecaNet?

Cansancio visual. Dolores en la espalda si no se adopta una postura correcta. Dolores en dedos, cuello y/o muñecas.

1.1.4 ¿Cuáles son las características de la mecanografía?

La mecanografía Es el arte de escribir a máquina y el proceso de introducir texto en un dispositivo por medio de un teclado como los que poseen las máquinas de escribir, los ordenadores o las calculadoras. El término se acuñó cuando se empezaron a utilizar las primeras máquinas de escribir mecánicas.

¿Cuáles son los beneficios de la mecanografía computarizada?

Ventajas de la Mecanografía para niños y adultos:

Adquirir el hábito. Si una persona automatiza una forma de escribir errónea, es muy difícil cambiarla para adoptar la correcta.

Tendrán precisión y velocidad.

Agilidad mental.

Aumento de la productividad de las tareas que se han realizado.

Escribir sin errores.

¿Cómo ingresar a MecaNet?

PASOS PARA INGRESAR Primer paso:

En el escritorio está un ícono de programa llamado MecaNet, sitúan el puntero y le dan doble clic con el botón primario o izquierdo del mouse. Al darle doble clic muestra la siguiente ventana.

Después despliega esta ventana.

¿Cuál es la velocidad media de mecanografía?

La velocidad con que una persona introduce textos en un ordenador o una máquina de escribir se mide en palabras por minuto (ppm), o caracteres por minuto (ppm). La cantidad de palabras por minuto que una persona puede llegar a conseguir a través del entrenamiento depende de muchos factores.

¿Cuál es la importancia de la mecanografía en el ámbito escolar?

Si se traslada al aspecto ocupacional o profesional, prepara al estudiante para ingresar en el campo laboral, ya que después de toda una preparación en el área mecanográfica estará capacitado para realizar las diferentes tareas que se realizan en una oficina con mayor rapidez. El lector promedio está entre 200 y 300 palabras por minuto, pero es susceptible de mejora con una serie de ejercicios.

1.1.5 riesgos:

Son muy pocos los riesgos ya que tendremos que revisar si la página es segura para saber si la página es verídica y segura es muy fácil podremos ver un candado arriba en el url, ya que hay otro tipo de compañías o personas que se dedican a duplicar las pestañas y poder meter malware para robar información.



Tipos de malware fuente:

[7 tipos de malware informático y cómo prevenirlos en 2022 - REPARACION ORDENADORES MAC MADRID \(reparacion-portatiles-madrid.net\)](#)

1.2 Sistemas operativos

Por: Gustavo Rafael Juaréz Mendoza

Un sistema operativo es un conjunto de programas que permite manejar la memoria, disco, medios de almacenamiento de información y los diferentes periféricos o recursos de nuestra computadora, como son el teclado, el mouse, la impresora, la placa de red, entre otros.

Encontramos diferentes sistemas operativos como Windows, Linux, Mac Os, en sus diferentes versiones. También los teléfonos tablets poseen un sistema operativo.

Partes de un sistema Operativo

Un sistema operativo consta de muchos componentes o partes y de características distintas. Estas características se definen como parte del sistema operativo varias con cada sistema operativo. Sin embargo, las tres partes del SO más fácilmente definidas y usadas por los SO son:

Kernel: es el programa del SO que podríamos decir que es el corazón de tu sistema operativo, por ese motivo se llama el "NÚCLEO" del SO

Kernel El Kernel o núcleo, es una parte fundamental del sistema operativo que se encarga de conceder el acceso al hardware de forma segura para todo el software que lo solicita, El Kernel es una pequeña e invisible parte del sistema operativo, pero la más importante, ya que sin esta no podría funcionar. Todos los sistemas operativos tienen un Kernel, incluso Windows 10, pero quizá el más famoso es el Kernel de Linux, que ahora además está integrado en Windows 10 con sus últimas actualizaciones.

Tipos de Sistemas Operativos

A nivel del usuario de un PC, podemos decir que hay dos tipos de Sistemas Operativos: SO para PC y SO Para Teléfonos Móviles. Pero también hay una amplia familia de sistemas operativos categorizados según los tipos de ordenadores que controlas y el tipo de aplicaciones que admiten.

Las 4 categorías para este tipo de SO son SO en tiempo real, SO Usuario Único Tarea Única, SP Usuario Único Tarea Única, SO Usuario Único Multitarea y SO Multitarea y SO Multiusuario..

También podemos clasificarlos como sistemas operativos de pago y sistemas operativos libres.

los de pago, tienen un costo económico y los libres son gratuitos para usar, modificar y descubrir.

SO Según el Ordenador y el Tipo de Aplicaciones

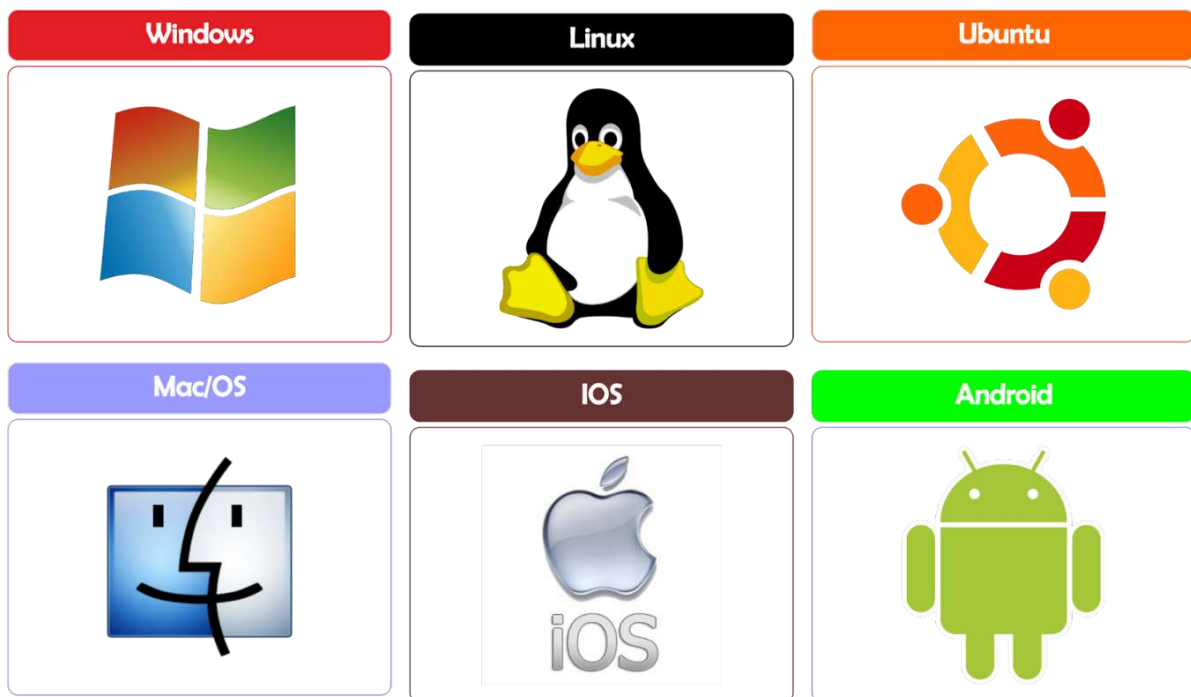
Sistema Operativo en tiempo real (RTOS): los sistemas operativos en tiempo real se utilizan para controlar maquinaria, instrumentos científicos y sistemas industriales

Una parte muy importante de un RTOS es administrar los recursos de la computadora para que una operación particular se ejecute exactamente en la misma cantidad de tiempo, cada vez que ocurre.

Usuario único, tarea única: Como su nombre lo indica, este sistema operativo está diseñado para administrar la computadora de modo que un usuario pueda hacer una cosa a la vez.

Usuario único, multitarea: Este es el tipo de sistema operativo que la mayoría de la gente usa en sus computadoras de escritorio y portátiles en la actualidad.

Windows de Microsoft y las plataformas MacOS de Apple son ejemplos de los sistemas operativos que permitirán que un solo usuario tenga varios programas en funcionamiento al mismo tiempo.



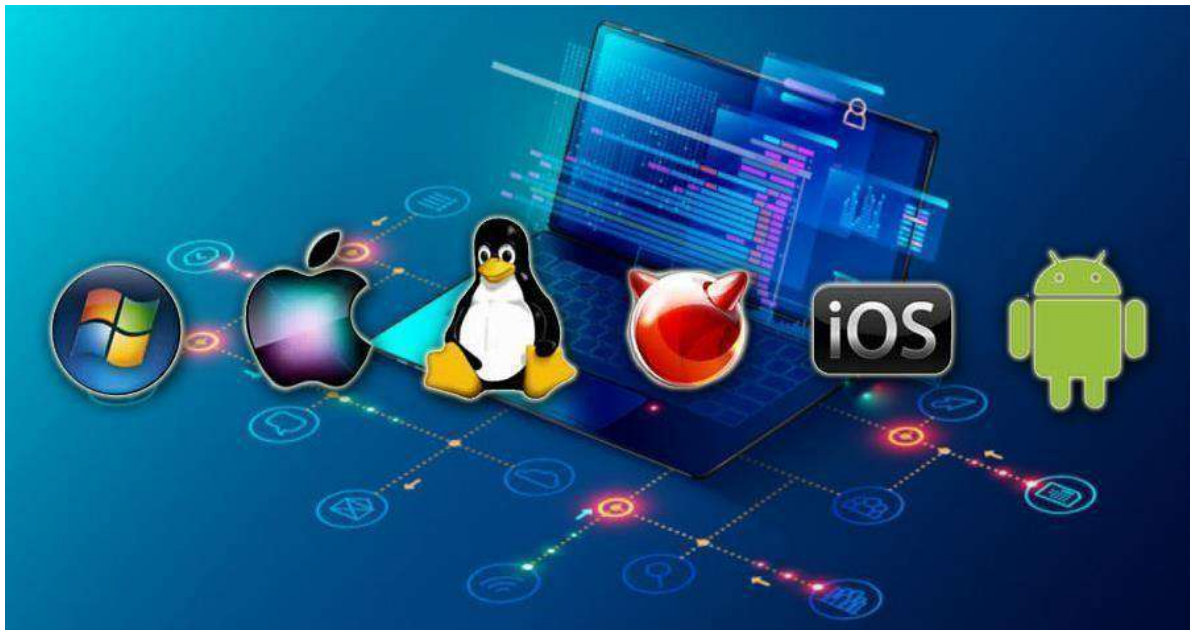
Fuente:

<https://sites.google.com/site/informatica2d38/entorno-al-sistema-operativo>
<https://www.areatecnologia.com/sistemas-operativos.htm>

Compatibilidad de los SO

La mayoría de los programas de software están diseñados para funcionar con el sistema operativo de una sola empresa, por ejemplo Windows (Microsoft) o solo macOS (Apple).

Un software indicara claramente que sistemas operativos admite y sera muy especifico si es necesario incluyendo la versión o versiones de ste SO que admite



Fuente:

<https://www.google.com/url?sa=i&url=https%3A%2F%2Funiversodigital.org%2Ftag%2Fclasificacion-de-los-sistemas-operativos%2F&psig=AOvVaw2SKBoPqosXJJqZG29GnEHs&ust=1675907196168000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCNCcyfvmhP0CFQA AAAAdAAAAABAS>

1.2 Historia De Los Sistemas Operativos

El sistema operativo es el principal software que debemos tener instalado en nuestra computadora, ya que sin el es como si no tuvieramos un ordenador activo

Cual fue el primer sistema operativo que existio y porque se creo

Lo que hoy conocemos como la informática proviene de una constante evolución a través del tiempo. los primeros pasos se dieron a partir de la II Guerra Mundial, a mediados de los 40, una época en la que aún no existía ni siquiera el término de 'Sistema Operativo'

En los 70 salieron a la luz los sistemas operativos como MacOS, MS-DOS, y Windows. En competencia, nuevas empresas se sumaron en la creación de nuevos sistemas y nació Linux Siendo esto ya en los años 90, que luego se unió al proyecto GNU, que es un sistema operativo completamente libre.

¿Cuál fue el impacto de este SO en la Historia de la informática?

Como ya mencionamos, Unix no fue el primer sistema operativo, pero si es el que impactó más en la sociedad y en la evolución de los sistemas operativos. Nació específicamente en 1969, creado por Ken Thompson, Dennis Richie y Ruc Canaday, entre otros.

Al principio se bautizó como UNICS, pero por ser fallido se modificó a UNIX,

desarrollado por los laboratorios Bell de AT&T y la General Electric, aunque su primer lanzamiento fue discriminado un poco por ser un sistema operativo caro y lento.



1.2.2 Versiones de sistemas operativos

Un sistema operativo es un conjunto de programas que controla el funcionamiento de las partes físicas de un equipo electrónico, facilitando al usuario el uso de la computadora u otro equipo. Es decir, es el software que maneja al hardware. a cra

Sistema operativo MS/DOS

El sistema operativo DOS, por Disk Operating System o MS/DOS fue diseñado por Microsoft para las computadoras personales IBM en 1981. MS/DOS podía administrar discos floppy y archivos, memoria y dispositivos de entrada y salida. Se controla a través de comandos.

Versiones del sistema operativo Windows

El sistema operativo más conocido es Windows, ampliamente utilizado en las computadoras personales PC de la compañía Microsoft. Microsoft Windows es una familia de sistemas operativos gráficos que han evolucionado a lo largo de los años:

- Windows 1.0 en 1985 • Windows 2.0 en 1987 • Windows 3.0 en 1990
- Windows 3.1 en 1992
- Windows 95 en 1995
- Windows 98 en 1998
- Windows Millenium ME en 2000
- Windows XP en 2001
- Windows Vista 2006
- Windows 7 en 2009
- Windows 8 en 2012
- Windows 10 en 2015
- Windows 11 en 2021

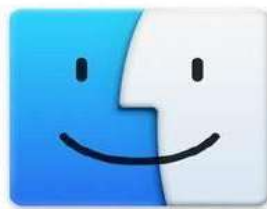


Imagen: logo de windows

Fuente: <https://concepto.de/windows-2/>

Versiones de sistema operativo macOS

- macOS Ventura 13.2
- macOS Monterey 12.6.3
- macOS Big Sur 11.7.3
- macOS Catalina 10.15.7
- macOS Mojave 10.14.6
- macOS High Sierra 10.13.6
- macOS Sierra 10.12.6
- OS X El Capitan 10.11.8
- OS X Yosemite 10.10.8
- OS X Mavericks 10.9.8
- OS X Mountain Lion 10.8.8
- OS X Lion 10.7.8
- Mac OS X Snow Leopard 10.6.8
- Mac OS X Leopard 10.5.8
- Mac OS X Tiger 10.4.11
- Mac OS X Panther 10.3.9
- Mac OS X Jaguar 10.2.8
- Mac OS X Puma 10.1.5
- Mac OS X Cheetah 10.0.4



MacTM OS

Imagen: Logo de MacOS

Fuente: <https://conceptoabc.com/mac-os/>

Versiones de sistema operativo Linux

- Ubuntu MATE
- Apricity OS

- Linux Mint
- Antergos
- ZORIN
- Elementary OS
- Xubuntu • Manjaro Linux

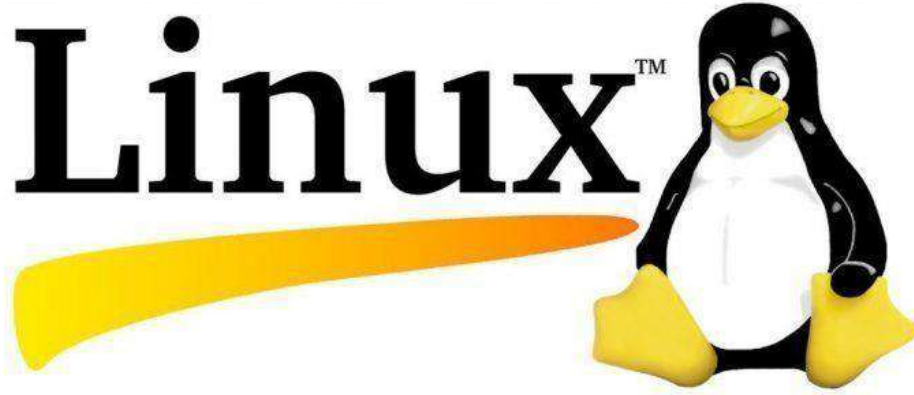


Imagen: Logo de Linux

Fuente: <https://urko.net/producto/sistema-operativo-linux-a-2/>

1.2.3 Distribuciones de los sistemas operativos

Consiste en crear una metadistribución de sistema operativo libre con identidad nacional, que favorezca la autonomía tecnológica y establezca estándares de seguridad y desarrollo propios.

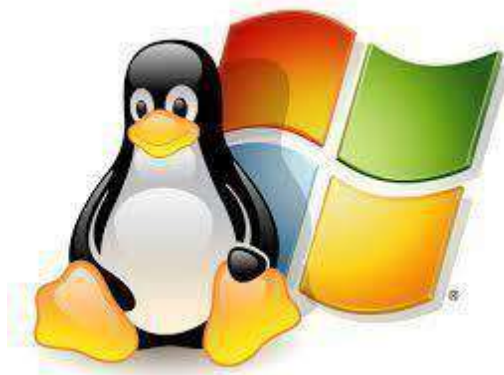


Imagen: Sistemas operativos

Fuente: <https://www.timetoast.com/timelines/sistemas-operativoscomerciales-y-de-libre-distribucion>

4.1 Distribución de sistema operativo windows

4.1.1 Versiones para PC

- Windows Home
- Windows Pro
- Windows Enterprise
- Windows Education

4.1.2 Versiones para dispositivos móviles

- Windows Mobile
- Windows Mobile Enterprise
- Version especial para dispositivos de tipo Raspberry Pi

4.2 Distribución de sistema operativo Linux

Las distribuciones de Linux son sistemas operativos instalables diseñados a partir del kernel de Linux, que admiten programas de usuarios, repositorios y bibliotecas. La versión de cada comunidad o proveedor es una distribución.

Las distribuciones de Linux están disponibles en versiones comunitarias o empresariales. Las comunitarias son distribuciones gratuitas de Linux cuyo soporte y mantenimiento están a cargo principalmente de la comunidad open source. En cambio, la distribución empresarial (o comercial) de Linux está disponible a través de una suscripción con un proveedor y no depende únicamente del respaldo de la comunidad.

La principal diferencia entre ambas es quién decide lo que es importante para los usuarios. El rumbo de las distribuciones comunitarias depende de los colaboradores, quienes eligen y mantienen paquetes de varias opciones de software open source. En cambio, las empresariales están a cargo de los proveedores, quienes toman las decisiones teniendo en cuenta las necesidades de los clientes.

- Android
- Arch Linux
- CentOS
- Debian
- Elementary OS
- Fedora Linux
- Gentoo Linux
- Kali Linux
- Linux Lite
- Linux Mint • Manjaro Linux

1.3 Unidades de medida

Por: María Emilia Larios Contreras

1.3.1 Concepto

Las unidades de medida de la información son aquellas unidades de medición que permiten determinar cuánto espacio hay disponible en una unidad de memoria. Con ellas podemos obtener datos sobre la unidad de almacenamiento, capacidad de tarjetas de memorias, pendrives y de todo aquello que sirva para almacenar.

Se le llama unidad de medida de la información al registro del espacio que hay en un dispositivo dado para grabar datos e información de manera permanente o temporal. También se puede decir que una unidad de medida de la información es aquella práctica que se realiza con el interés de optimizar el rendimiento y aprovechar todo el espacio que existe dentro de una unidad.

Sirven para cuantificar y obtener información esencial. En las unidades de almacenamiento informático también se aplican las mismas magnitudes que en las unidades de medida física, como son kilo, mega, giga o tera; cada una equivale a mil unidades de la anterior.

1.3.1 Múltiplos

os Bit

El bit (Binary Digit) significa dígito binario, se representa con la longitud "b", corresponde a la unidad más pequeña de información de un equipo. Este tiene un valor que puede ser 0 o 1, y con ella se puede representar una cantidad infinita de condiciones binarias que pudiese ser encendido y apagado, verdadero y falso, presente y ausente. Por lo general, estos son manejados para describir velocidades de transmisión. Una sucesión de bits representa una cadena de información o pulsos eléctricos.

Las magnitudes del bit son las siguientes:

- **Kilobit;** que equivale a **1000 bits**.
- **Megabit;** que equivale a **1000 kilobits**.
- **Gigabit;** que equivale a **1000 megabits**.

Es importante saber que el sistema binario es un sistema de base 2 (dos valores) por lo que para determinar cuántas combinaciones de bits podemos hacer, solamente tendríamos que elevar la base a la n-ésima potencia según los bits que queramos.

Bit = 0 • 1

Imagen: Representación gráfica del Bit.

Fuente: <https://internetpasoapaso.com/wp-content/uploads/que-es-un-bit.jpg>

En la medida de los bits, existen algo llamado **bit más significativo (MSB)** que se le

denomina al bit de mayor peso o mayor valor y **bit menos significativo (LSB)** al bit de menor peso o menor valor, mientras mayor sea la posición hacia la derecha, mayor valor tendrá el bit.

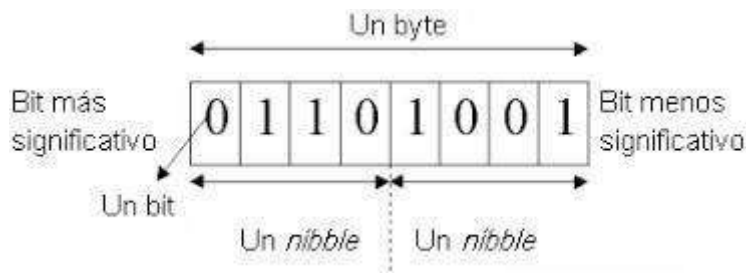


Imagen: Ejemplo de bit más significativo y menos significativo. Fuente:

<https://www.google.com/url?sa=i&url=https%3A%2F%2Facademicos.azc.uam.mx%2Fjfg%2Fdiapositivas%2Ficomp%2FUnidad2.pdf&psig=AOvVaw1EqxIVi6g9YFtvx5R1ml5J&ust=1675913060953000&source=images&cd=vfe&ved=0CAMQjB1qFwoTCMi0tuL8hP0CFQAAAAAdAAAAABAE>

Byte

El byte (B) corresponde a la unión de 8 Bits consecutivos, por lo que esta se considera una combinación del primero. Cabe destacar que con ella se pueden lograr hasta 256 conjugaciones, siendo suficiente para representar todo el alfabeto, signos de puntuación, números y más. Es decir, cada carácter que se introduce en un ordenador es un byte, siguiendo las equivalencias del código ASCII.

Las magnitudes del byte son las siguientes:

- **Kilobyte**; que equivale a **1000 byte**.
- **Megabyte**; que equivale a **1000 kilobyte**.
- **Gigabyte**; que equivale a **1000 megabyte**.
- **Terabyte**; que equivale a **1000 gigabyte**.

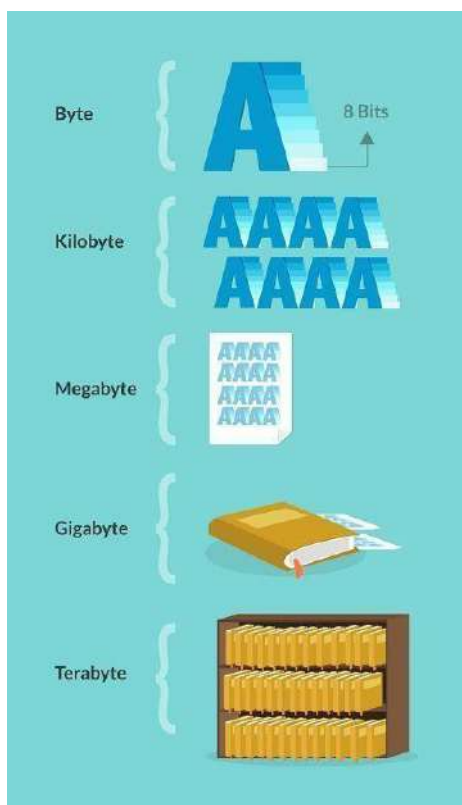


Imagen: Ejemplo gráfico de magnitudes de almacenamiento del Byte.

Fuente:

https://aprendelibvrefiles.blob.core.windows.net/aprendelibvrecontainer/article/medidas_de_almacenamiento_de_informacion/image/byte-01_l.png

Hercio

Se utilizan para medir la velocidad de procesamiento de la información, también es el número de oscilaciones que sufre una onda electromagnética o sonora en un segundo. Un hercio es el equivalente a un ciclo, es decir, algo que se repite una vez, como podrían

ser los giros de un procesador en un determinado espacio de tiempo. Este valor se mide en hertzios cuya magnitud es la frecuencia. Dicho esto, se puede hablar de hertzios por segundo, minuto e incluso horas.

Las magnitudes de los hertzios son las siguientes:

- Kilohertz; que equivale a 1000Hz.
- Megahertz; que equivale a 1000Khz.
- Gigahertz; que equivale a 1000Mhz.
- Terahertz; que equivale a 1000Ghz.

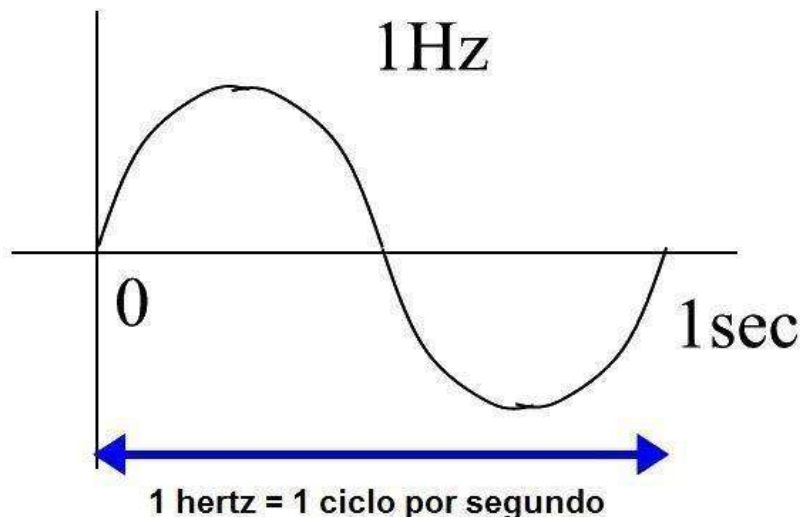


Imagen: Representación gráfica de un Hertz.

Fuente: <https://saullainez.files.wordpress.com/2017/01/1hzhertz.jpg?w=600>

El Baudio

El baudio es una unidad de medida derivada directamente del bit, se mide en "bit per second" (bps) ó bit por segundo, pero para mayor comodidad se utilizan los Kilobits por segundo (Kbps) y los Megabits por segundo (Mbps). Esta unidad se suele utilizar para definir las velocidades de transferencia en las redes basadas en cable, redes inalámbricas, módems y en algunos casos en dispositivos como memorias USB. Esta unidad mide la cantidad de bits que se transmiten en cuestión de un segundo.

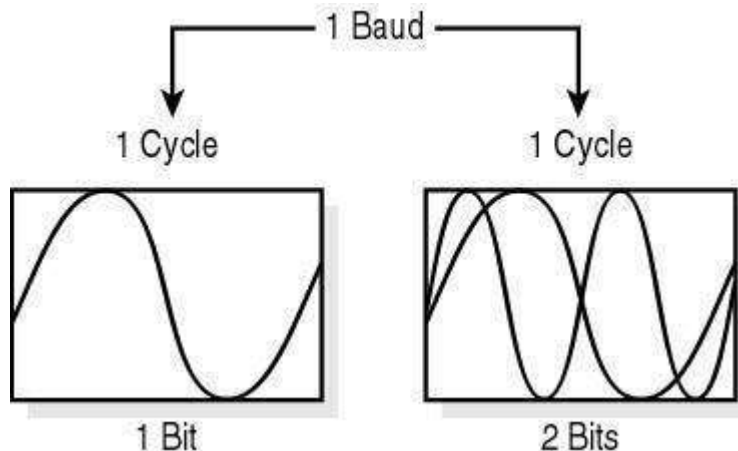


Imagen: Representación gráfica de un Baudio.

Fuente: <http://www.brainbell.com/tutors/A+/Hardware/images/f12xx07.jpg>

Comparación de unidades binarias y decimales y valores

Nombre (Binario)	Símbolo	Valor (base 2)	Nombre (Decimal)	Símbolo	Valor (base 10)
kibibyte	KiB	2^{10}	kilobyte	KB	10^3
mebibyte	MiB	2^{20}	megabyte	MB	10^6
gibibyte	GiB	2^{30}	gigabyte	GB	10^9
tebibyte	TiB	2^{40}	terabyte	TB	10^{12}
pebibyte	PiB	2^{50}	petabyte	PB	10^{15}
exbibyte	EiB	2^{60}	exabyte	EB	10^{18}

1.3.4 Conversión de unidades

Es importante recordar que el bit es la unidad mínima, el byte es equivalente a 8 bits, el Kilobyte equivale a 1024 bytes, el Megabyte equivale a 1024 kilobytes, el Gigabyte equivale a 1024 megabytes y el Terabyte equivale a 1024 gigabytes.

Entonces si se desea convertir de una unidad menor a una unidad mayor por ejemplo 2Kb (kilobytes) a Megabyte se dividiría entre 1024 o sea $2/1024=0.001953125$

Si es convertir a una unidad mas grande como Gigabyte se tiene que volver a dividir entre 1024 y si es a Terabyte se divide de nuevo pero si es a bit primero hay que convertir a byte y en bit se divide entre 8.

Si se quiere convertir una unidad mayor a una menor se multiplican las veces necesarias por 1024 por ejemplo 12Mb a bits

- Megabyte-Kilobyte: $12 \cdot 1024 = 12288$
- Kilobyte-byte: $12288 \cdot 1024 = 12582912$
- Byte-bit Se multiplica por 8 no por 1024: $12582912 \cdot 8 = 100663296$

1.4. Sistema de numeración

Por: Glendy Lopez

La utilidad de los sistemas de numeración en el tema que nos ocupa es determinante. La razón es que todos los ordenadores utilizan el sistema de numeración binaria para representar la información (no sólo los números). Los humanos han estado contando durante mucho tiempo. Para ello, utilizamos sistemas que relacionan símbolos únicos con valores específicos. A esto se le llama sistema numérico, y es la técnica que usamos para representar y manipular números.

Un sistema numérico debe tener símbolos únicos para cada valor, ser coherente, proporcionar valores comparables y ser fácilmente reproducible.

Tipos

Hay varios tipos de sistemas numéricos y los más utilizados son el sistema numérico decimal, el sistema numérico binario, el sistema numérico octal y el sistema numérico hexadecimal.

Importancia

El sistema numérico ayuda a representar números en un pequeño conjunto de símbolos. Las computadoras, en general, usan números binarios 0 y 1 para mantener los cálculos simples y mantener menos la cantidad de circuitos necesarios, lo que resulta en la menor cantidad de espacio, consumo de energía y costo.

Características

Símbolos únicos, consistencia, resultados de valores comparables y son fáciles de reproducir. El sistema decimal sigue siendo el sistema más básico utilizado por los humanos con un poder base de 10, pero las mismas cosas no pueden aplicarse a las máquinas.

Las máquinas interpretan los números de manera diferente a los humanos, por lo que requieren un sistema completamente diferente. Cada vez que escribimos algo en un dispositivo, esas letras se convierten en ciertos números que solo la computadora puede entender. Un sistema de números posicionales es lo que necesita un dispositivo donde hay dígitos y diferentes valores que forman un número entero. Cada dígito tiene su propia posición única y un símbolo. El valor de cada dígito se puede determinar por: el dígito, la posición del dígito en el número y la base del número.

Sistemas numéricos que admite una computadora, los cuales son;

- Binario
- octales
- Decimal

- hexadecimal



1.4.2. Sistema binario

La unidad de almacenamiento más básica en un dispositivo está representada por un bit. Una computadora usa un bit para mostrar cualquier tipo de información. Los transistores son una parte importante de un sistema informático que permite que la corriente fluya en un dispositivo. Puede estar encendido o apagado.

Cada número que vemos en una computadora es una señal eléctrica que antes se representaba mediante el encendido y apagado creando un interruptor binario. El estado encendido y apagado usa 1 y 0 para representar la situación, lo que hace que el sistema numérico binario sea una base de 2.

Solo estos dos símbolos representan todos los números. 0 es para la tarifa más baja mientras que 1 es para la tarifa más alta. Los números no son una unidad individual aquí sino que están hechos de grupos de 1 y 0. Cada dígito binario es un bit y los valores posicionales son potencias ascendentes de dos de izquierda a derecha. El bit menos significativo está en el extremo derecho, mientras que el bit más significativo está en el extremo izquierdo.

Ejemplo:

= 11010
= $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
= $16 + 8 + 0 + 2 + 0$
= 2610

La tabla de conversión del bit es:

1 byte (B) = 8 bits
1 kilobyte (KB) = 1024 bytes
1 Megabyte (MB) = 1024 KB
1 Gigabyte (GB) = 1024 MB
1 Terabyte (TB) = 1024 GB
1 exabyte (EB) = 1024 PB
1 zettabyte = 1024 EB
1 Yottabyte (YB) = 1024 ZB

Convertir Decimal a Binario

Se divide el número decimal por 2 se mantenga el cociente entero para la siguiente iteración mantenga el resto para el dígito binario se repite los pasos hasta que obtengas 0 como tu cociente

Ejemplo:

División	Resto (R)
----------	-----------

$112 / 2 = 56$	0
$56 / 2 = 28$	0

$28 / 2 = 14$	0
$14 / 2 = 7$	0
$7 / 2 = 3$	1
$3 / 2 = 1$	1
$1 / 2 = 0$	1

Cada dígito binario se denomina bit . Por ejemplo,01100110 se compone de 4 bits que son00,11,11, y00. Cuando se trata de colocar valores a los bits (es decir, dígitos) en este sistema, colocamos valores correspondientes a potencias ascendentes de 2 de derecha a izquierda.

El bit más a la derecha se denomina bit menos significativo (LSB) y el bit más a la izquierda es el bit más significativo (MSB).

Puede manipular bits de izquierda a derecha con operadores bit a bit para cambiar de manera eficiente el valor de un número en el nivel de código de máquina.

1.4.3. Sistema Octal

El sistema numérico octal, como su nombre indica, tiene una base de ocho números. Los dígitos utilizados son: 0,1,2,3,4,5,6,7. Cada posición en un número octal tiene 0 potencia de la base.

Tienen muchas aplicaciones e importancia, ya que se utilizan en computadoras y sistemas de numeración digital. En el sistema numérico, los números octales se pueden convertir en números binarios, los números binarios en números octales al convertir primero un número binario en un número decimal y luego un número decimal en un número octal.

Similar al sistema numérico octal, el sistema numérico binario está representado por la base 2, el sistema numérico decimal está representado por la base 10 y el sistema numérico hexadecimal está representado por la base

Algunos de los ejemplos de estos sistemas numéricos son:

(10) ₂ (10) ₂ es un numero binario
(119) ₁₀ (119) ₁₀ es un numero decimal
(51) ₆ (51) ₆ es un número hexadecimal

La razón detrás del desarrollo de este sistema fue hacer el binario más compacto. Hace dígitos binarios en un grupo de tres en lugar de cuatro. El sistema base de ocho proviene únicamente del sistema numérico decimal. Por ejemplo – 125708

Al convertir al sistema decimal

$$\begin{aligned} &125708 \\ &= ((1 \times 8^4) + (2 \times 8^3) + (5 \times 8^2) + (7 \times 8^1) + (0 \times 8^0))_{10} \\ &= (4096 + 1024 + 320 + 56 + 0)_{10} \\ &= 5496_{10} \end{aligned}$$

Conversión de números octales a binarios

Para el proceso de conversión, necesitamos convertir cada número del número octal al número binario. Cada dígito debe convertirse en un número binario de 3 bits y, por lo tanto, llegar al equivalente binario del número octal. A continuación, se muestra una tabla de representación de los números binarios a los números octales y viceversa.

Ejemplo:

Convertir $(14)_8(14)_8$ en un número binario - Dado $(14)_8(14)_8$ es un número octal, escribimos $(14)_8(14)_8 = (001100)_2(001100)_2$. Los ceros a la izquierda no tienen ningún significado. Por eso, $(14)_8(14)_8 = (001100)_2(001100)_2$.

Conversión de número decimal a octal

Para convertir números decimales a octales, se utiliza un método diferente. En este método, el número decimal se divide por 8 cada vez que se obtiene un recordatorio del dígito anterior. El primer resto obtenido es el dígito menos significativo (LSD) y el último resto es el dígito más significativo (MSD).

Por ejemplo convierta el número decimal 321 a su forma octal. se divide el número

321 por 8
321/8 da cociente 40 y el resto es 1
40/8 da cociente 5 y el resto es 0

Entonces, aquí el cociente es 5 y el resto es 0. El número octal comienza desde MSD a LSD, es decir, 501

Por lo tanto, $(321)_{10} = (501)_8$

1.3.4 Convertir binario a octal

- El primer paso es agrupar los dígitos binarios en el conjunto de 3.
- Multiplique cada grupo agregando ceros para hacerlo divisible por 3.
- Escribe debajo un símbolo octal para cada grupo.
- Esto le dará un número octal que llegó de un número binario.
- Invertir los dos últimos pasos dará un número binario de uno octal.

Ejemplo: 1010111100
= (1010111100)₂
= (001 010 111 100)₂

$$= (1\ 2\ 7\ 4)_8$$

$$= (1274)_8$$

1.4.5. Sistema Hexadecimal

El sistema hexadecimal es una forma más compacta de representar números en una computadora, ya que requiere solo 4 bits para representar el valor de un dígito.

El sistema numérico hexadecimal evolucionó poniendo números binarios en un grupo de cuatro bits. Es un enfoque compacto para representar números en la computadora, ya que requiere solo 4 bits. El sistema numérico hexadecimal tiene una base de 16, lo que significa que usa 16 símbolos en total.

Este sistema tiene 10 números del sistema decimal y A, B, C, D, E y F como símbolos adicionales. Estas letras son solo para representar los números que vienen después de 10. Cada valor posicional en el número representa la potencia 0 de la base, mientras que los últimos dígitos tienen la potencia z de la base. Al convertir al sistema decimal:

$27FB \times 16 = 2 \times 16^3 + 7 \times 16^2 + 15 \times 16^1 + 10 \times 16^0$
$= 8192 + 1792 + 240 + 10$
$= 1023410$

Convertir binario a hexadecimal

El primer paso es agrupar los dígitos binarios en el conjunto de 4. Y luego cada cuarteto es reemplazado por una representación hexadecimal. Esto le dará un número hexadecimal que llegó de un número binario.

Ejemplo: 1010101101001
$= (1010101101001)_2$
$= (1\ 0101\ 0110\ 1001)_2$
$= (0001\ 0101\ 0110\ 1001)_2$
$= (1\ 5\ 6\ 9)_{16}$
$= (1569)_{16}$

Tabla de Relación del sistema numérico

HEXADECIMAL	DECIMAL	OCTAL	BINARIO
0	0	0	0000
1	1	1	0001
2	2	2	0010

3	3	3	0011
4	4	4	0100

5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
mi	14	dieciséis	1110
F	15	17	1111

1.4.5. Conversiones

La conversión es el proceso de convertir datos de computadora de una representación específica a otra cambiando bits de un formato de archivo a otro, generalmente para compatibilidad entre diferentes aplicaciones o sistemas.

Para convertir de cada sistema digital a decimal, este proceso es el proceso de proceso: el valor de cada número se multiplica por el peso del sistema de acuerdo con la base del sistema tratado.

Para convertir de un pequeño número de sistemas entre sí, el proceso es lo contrario: un pequeño número se divide entre el fondo del sistema, que desean recoger y dividir los restos de cada capítulo. Luego, estos residuos se toman en orden inverso, comenzando con el último cociente

La información fue obtenida de:

<https://byjus.com/maths/numbersystem/#:~:text=In%20general%2C%20the%20binary%20number,systems%20are%20also%20used%20sometimes>

<https://www.educative.io/blog/computer-number-systems-binaryhexadecimal-conversions>

<https://data-flair.training/blogs/number-systems-in-computer/> <https://www.includehelp.com/computer-number-systems.aspx>

1.5 Sistemas De Archivos

Por: **Didier Lopez**

Un sistema de archivos es el sistema de almacenamiento de un dispositivo de memoria, que estructura y organiza la escritura, búsqueda, lectura, almacenamiento, edición y eliminación de archivos de una manera concreta.

El objetivo principal de esta organización es que el usuario pueda identificar los archivos sin lugar a un error y acceder a ellos lo más rápido posible.

Los sistemas de archivos también otorgan a los archivos, entre otras, las siguientes características:

- Convenciones para nombrar a los archivos
- Atributos de archivo
- Control(es) de acceso

Sistemas de archivos más importantes:

Hay diversos sistemas de archivos estándar para Windows, macOS, Linux, Unix y el resto de sistemas operativos.

Se han ido diferenciando cada vez más: por ejemplo, se han creado sistemas de archivos adecuados para los dispositivos de almacenamiento *flash*, cada vez más populares, entre los que se incluyen las memorias USB y las unidades SSD.

FAT32

Uso:

Medios de almacenamiento extraíbles.

Sistema operativo (compatibilidad):

- Windows
- Mac OS X/macOS
- Linux (si se instalan los correspondientes controladores)

Características:

- Sin funciones de cifrado ni compresión
- No garantiza particularmente la seguridad de los datos
- Ideal para particiones más pequeñas
- Volumen máximo de datos: 4 GB

exFat

Uso:

Medios de almacenamiento extraíbles.

Sistema operativo (compatibilidad):

- Windows
- Mac OS X/macOS (compatible a partir de 10.6.4)
- Linux (si se instalan los correspondientes controladores)

Características:

- No permite gestionar derechos
- No permite comprimir los datos
- Ideal para memorias flash más pequeñas, a partir de 32 GB (memorias USB, tarjetas SD)
- Tamaños y particiones ilimitados (según el estado actual de la tecnología)

NTFS

Uso:

Discos duros internos y externos.

Sistema operativo (compatibilidad):

- Windows
- Mac OS X/macOS (de forma integral instalando un software adicional) - Linux (instalando controladores)

Características:

- Se especializa en archivos extensos y en grandes capacidades de almacenamiento
- Inadecuado para discos pequeños y particiones de menos de 400 MB (demasiada potencia)
- Volumen máximo de datos: 256 TB

APFS

Uso:

Unidades SSD.

Sistema operativo (compatibilidad):



- macOS (el estándar desde la versión 10.13 High Sierra)



- Versiones anteriores de Mac OS y Windows (instalando un software adicional)

Características:

- Optimizado para unidades de estado sólido (SSD) y otros dispositivos de almacenamiento *flash*
- También funciona en unidades mecánicas e híbridas
- Permite el cifrado de datos

HFS+

Uso:

Discos duros internos y externos.

Sistema operativo (compatibilidad):

Mac OS X/macOS

Características:

- Sistema de archivos maduro y probado
- Especialmente indicado para discos mecánicos
- No optimizado para tecnologías de almacenamiento modernas (SSD, *flash*) - Mejor compatibilidad con versiones anteriores que APFS

ext4

Uso:

Linux

Sistema operativo (compatibilidad):

- Linux
- Windows (solo con *software* adicional)
- Mac OS X/macOS (solo con *software* adicional)

Características:

- Mejora del rendimiento
- Mejora de la seguridad de los datos
- Incorpora cifrado (desde Linux Kernel 4.1)
- La nueva función *extents* aumenta la velocidad de procesamiento de archivos grandes y evita la fragmentación

- Gestión de derechos



Fuente: <https://www.ionos.es/digitalguide/servidores/know-how/sistemas-de-archivos/>

1.5.1 Rutas y nombres de archivos (Windows, Linux)

El Explorador de archivos en los sistemas operativos Windows tiene un límite de ruta de 255 caracteres.

Con el tiempo, los usuarios tienden a profundizar la estructura de carpetas y exceden el límite de ruta de 255 caracteres al que están limitados el Explorador de archivos y muchas aplicaciones en Windows.

Sin embargo, existen algunas alternativas que usan MyWorkDrive para escenarios que requieren profundidades de ruta más profundas.

Alternativas con MyWorkDrive

Los servidores MyWorkDrive que ejecutan la versión 6.2 y posteriores pueden admitir archivos con una ruta de más de 255 caracteres utilizando MyWorkDrive cliente web, si se realizan los ajustes apropiados en el sistema operativo en el servidor MyWorkDrive y los servidores de archivos.

Consideración/Limitaciones

- Para una interacción consistente y sin preocupaciones con archivos que tienen rutas de más de 255 caracteres, MyWorkDrive cliente web debe ser usado. La razón por la que Web Client funciona es porque aprovecha la API de Windows y no utiliza el Explorador de archivos de Windows para sus operaciones.
- Si algún usuario, incluido un administrador, intenta interactuar con archivos usando el Explorador de archivos de Windows (ya sea directamente a través de SMB o mientras usa MyWorkDrive Desktop Client), es probable que ese usuario experimente problemas para acceder o interactuar con los archivos.
- Microsoft Office tiene un límite estricto de 250 caracteres para los nombres de archivo. Si un usuario intenta guardar, abrir o modificar un archivo con más caracteres, fuera de MyWorkDrive Web Client, se producirá un error y fallará.

Configuración



- Se requiere MyWorkDrive Server 6.2 o posterior.



- SSO debe estar habilitado y utilizado para el inicio de sesión de los usuarios.
- El siguiente valor de clave de registro debe establecerse tanto en su servidor MyWorkDrive y su(s) servidor(es) de archivos. Después de establecer el valor de la clave de registro, es necesario reiniciar.
-

Linux

Conocer el nombre de un archivo (o de un directorio) no es suficiente para acceder a él; un mismo nombre puede estar asociado a varios archivos en directorios diferentes.

Existen tres tipos de rutas: absolutas, relativas y personales. Pueden utilizarse indiferentemente en la forma de nombrar archivos en la línea de comando.

Rutas absolutas

Una ruta absoluta se basa en la raíz del árbol de Linux. Toda ruta absoluta empieza, pues, por /.

Sea cual sea la ubicación actual donde se encuentre, se podrá referenciar el archivo *notas* del ejemplo anterior con la ruta `/home/willy/notas`.

Rutas relativas

Las rutas relativas dependen del directorio actual en el que se encuentra el usuario.

Sabiendo que cada directorio en el sistema contiene los archivos `.` (punto) y `..` (punto-punto) que referencian respectivamente el propio directorio actual y el directorio padre, existen varias rutas relativas que designan el archivo *notas* en el ejemplo:

Directorio actual	Ruta relativa correspondiente
<code>/home</code>	<code>willy/notas</code>
<code>/home/willy</code>	<code>./notas</code>
<code>/home/willy/colores</code>	<code>../notas</code>
<code>/</code>	<code>home/willy/notas</code>
<code>/home...</code>	

Imagen: Tabla de rutas relativas.



Fuente:

<https://www.edicioneseni.com/open/mediabook.aspx?idR=cca5c07859681efac234687bd0c71c75>

1.5.2. Clasificación: de disco, de red, de propósito especial

Un disco duro es una unidad de hardware que se usa para almacenar contenido y datos digitales en las computadoras.

Todas las computadoras tienen un disco duro interno, pero también hay discos duros externos que pueden usarse para ampliar el almacenamiento de una computadora. Aquí, exploraremos los distintos tipos de discos duros, y sus ventajas y desventajas.

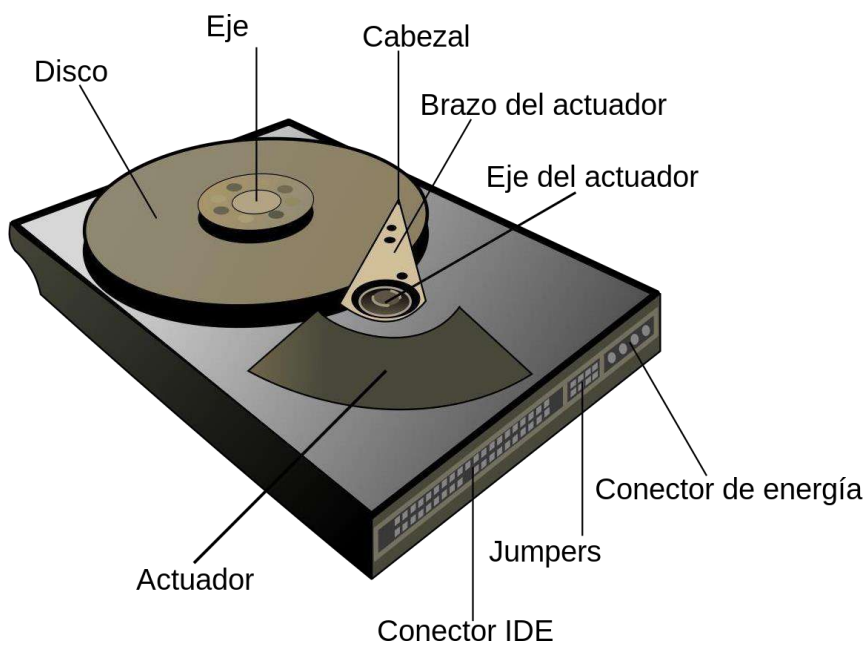


Imagen: Partes de disco duro.

Tipos de almacenamiento secundario

Todas las computadoras necesitan que los discos almacenen datos a largo plazo. Esto se conoce como almacenamiento secundario y la RAM (memoria de acceso aleatorio) de una computadora es su almacenamiento primario.

En general, el almacenamiento secundario se presenta en dos formas: las unidades de disco duro (HDD) y las unidades de estado sólido (SSD).

Tarjeta de red

La clasificación de las redes informáticas se refiere a la nomenclatura y clasificación de la infraestructura de una red.

Esta clasificación incluye redes LAN (Local Area Network), MAN (Metropolitan Area Network), WAN (Wide Area Network) y GAN (Global Area Network). Los ejemplos de redes LAN, MAN y WAN incluyen la Internet, una red bancaria nacional, redes empresariales transnacionales, redes satelitales militares y redes

de televisión. Las características principales de una red informática son el intercambio de datos entre dispositivos conectados a través de un medio físico o inalámbrico.



Imagen: Tarjeta de Red

Fuente:

https://es.wikipedia.org/wiki/Sistema_de_archivos
[¿Qué es un disco duro? - Dropbox](#)

1.5.3 FAT

FAT es un sistema de archivos desarrollado para MS-DOS. Es el sucesor de FAT16, que a su vez es sucesor de FAT12.

El tamaño máximo soportado por este sistema es de 4 GB-1 Byte y se utiliza para el intercambio de datos entre distintos sistemas operativos de un mismo equipo. Además, también es utilizado en tarjetas de memoria y dispositivos similares.



Imagen: Ejemplo de FAT.

NTFS

Está incluido en las versiones de Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, Windows Vista, Windows 7 y Windows 8. El tamaño mínimo recomendado para las particiones de este tipo de sistemas de archivos es de 10 GB, siendo posibles tamaños mayores.

En cuanto al rendimiento, NTFS es mucho más rápido en el acceso a los archivos que una partición tipo FAT. Esto se debe a que utiliza un árbol binario

de alto rendimiento para localizar los archivos. El tamaño límite de una partición es de $17 \cdot 10^9$ Bytes



Imagen: Ejemplo de NTFS.

EXT2

En Ext2, al contrario que en FAT, las entradas de la tabla de entrada de ficheros apuntan a un bloque especial llamado i-nodo el cual incluye información de tamaño, permisos, propietario y localización en disco, es decir, los metadatos del fichero. Además de los metadatos los i-nodos contienen dos tipos de referencias a los bloques del fichero:

- 12 referencias directas a bloques.
- 3 tipos de referencias indirectas: simple, doble y triple (en base al numero de tablas de bloques de cada referencia, desde 1 hasta 3).

Inconvenientes

- El principal inconveniente de un enlace duro es que los bloques de datos no se liberan hasta que el contador de número de referencias al i-nodo es cero. Por lo que el usuario no libera espacio en la unidad de almacenamiento hasta que todos los enlaces duros previamente creados no hayan sido borrados.
- Solo puede enlazarse a i-nodos dentro del mismo volumen.

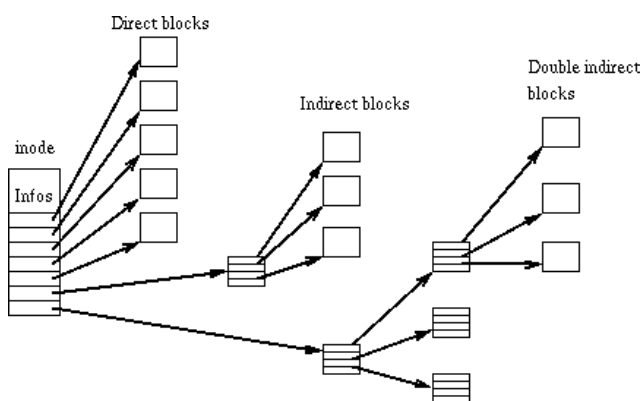


Imagen: Ejemplo de EXT2.

EXT3

Es un sistema de archivos principalmente utilizado en distribuciones Linux con registro por diario (journaling). Está siendo reemplazado por su sucesor, ext4, aunque todavía se utiliza.

¿Qué es el journaling?

El journaling se basa en llevar un registro diario en el que se almacena la información necesaria para reestablecer los datos del sistema afectados por un cambio, en caso de que falle.

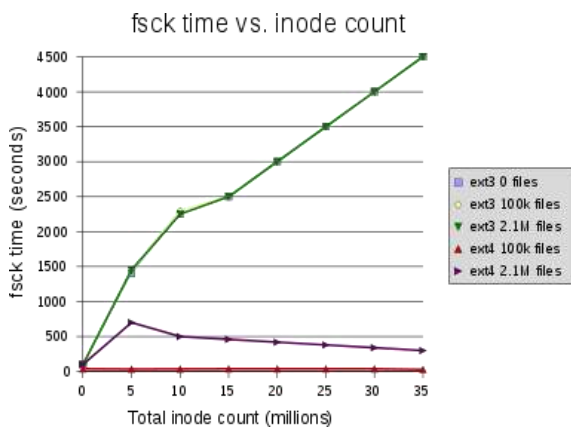


Imagen: Ejemplo de EXT3

EXT4

Es una mejora compatible de ext3 que utiliza menos CPU y mejora la velocidad de lectura y escritura. Además, soporta volúmenes de hasta 1024 PiB (PebiByte) (1 PiB = 2^{50} Bytes).

En ext4 se introducen los exents, que se utilizan para reemplazar al tradicional esquema de bloques utilizado por ext2 y ext3. Los exents mejoran el rendimiento al trabajar con ficheros de gran tamaño.

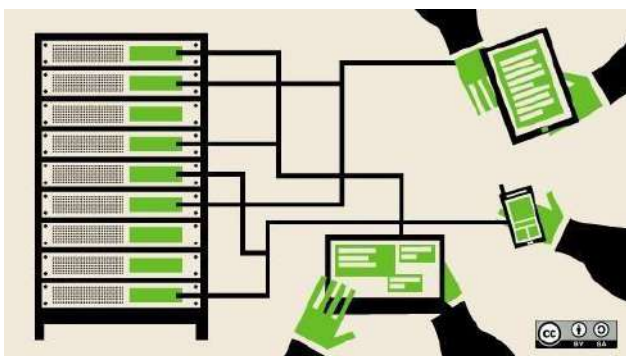


Imagen: Ejemplo de EXT4

Fuente: <https://geekytheory.com/sistemas-de-archivos-fat32-ntfs-ext3-y-ext4/c>
<https://1984.lsi.us.es/wiki-ssoo/index.php/EXT2>

TEORÍA DE LA INFORMACIÓN

Bimestre II

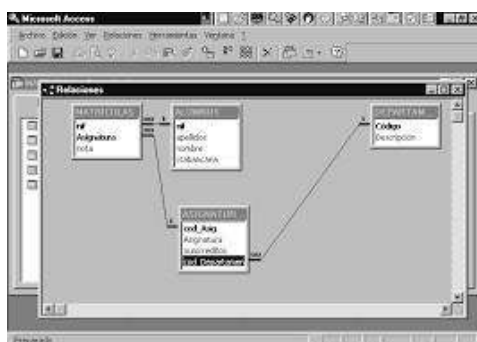
1. Teoría de Bases de Datos, Estructura de un Sistema de Bases de Datos y Seguridad de la Información.

Por: Maryori Omeydi Sipac Sesam.

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico. Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de

2.1 Teoría de Bases de Datos.

Para un tratamiento "mecanizado" de la información es fundamental que los datos se organicen de forma que se facilite su gestión, es decir, que el acceso a los mismos sea rápido y eficaz. Convencionalmente, el trabajo en informática de gestión consistía en realizar una serie de programas en determinados lenguajes, para creación, actualización, consulta y listados de datos, que trabajan con información de entrada para obtener unos resultados o datos de salida. Una base de datos es un conjunto de datos relacionados entre sí, que están estructurados de forma que puede accederse a ellos automáticamente e independientemente de los programas que gestionan esos datos. Esta independencia se refiere a la posibilidad de modificar la estructura de los datos sin necesidad de cambiar los programas que los manipulan, evitando con ello los problemas de actualización de datos previamente existentes.



Fuente:

<https://si.ua.es/es/documentos/documentacion/office/access/teoria-de-bases-de-datos.pdf>

2.1.1 DBMS.

El DBMS proporciona a los usuarios y programadores una forma sistemática de crear, recuperar, actualizar y administrar datos. Un DBMS también permite a los usuarios finales crear, leer, actualizar y eliminar datos en una base de datos. El DBMS esencialmente sirve como una interfaz entre la base de datos y los usuarios finales o programas de aplicación, asegurando que los datos estén organizados de manera consistente y permanezcan fácilmente accesibles. El DBMS gestiona tres cosas importantes: los datos, el motor de la base de datos que permite acceder a los datos, bloquearlos y modificarlos, y el esquema de la base de datos, que define la estructura lógica de la base de datos. Estos tres elementos fundamentales ayudan a proporcionar concurrencia, seguridad, integridad de datos y procedimientos de administración uniformes.

2.1.2 Tipos de DBMS.

Un sistema de administración de bases de datos, DBMS por sus siglas en inglés (Database Management System) es un software que se usa para definir, manipular, recuperar, almacenar y gestionar datos en bases de datos.

En resumen, los sistemas de bases de datos se encargan de:

Definir reglas para validar y manipular datos.

Interactuar con bases de datos, aplicaciones y usuarios finales. Recuperar, almacenar y analizar datos.

Actualizar datos. MySQL

MariaDB

Microsoft SQL Server

Oracle DBMS

PostgreSQL MongoDB

2.1.3 Diferencia entre BD y DBMS.

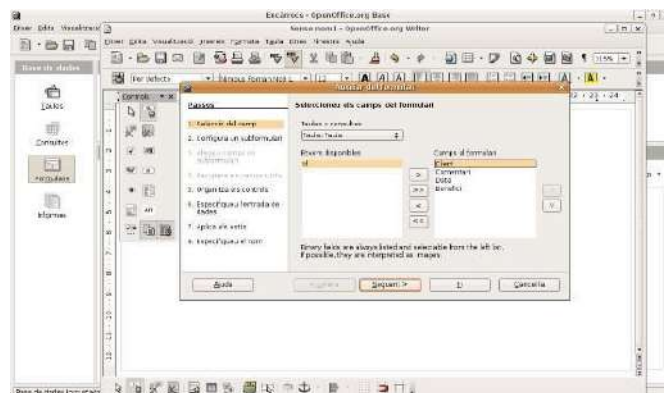
Un sistema destinado a organizar, almacenar y recuperar grandes cantidades de datos fácilmente, se denomina base de datos. En otras palabras, una base de datos contiene un conjunto de datos organizados (generalmente en forma digital) para uno o más usuarios. Las bases de datos, a menudo abreviadas DB, se clasifican según su contenido. Sin embargo, un sistema de gestión de bases de datos (DBMS, por sus siglas en inglés) es en realidad el sistema completo utilizado para administrar bases de datos digitales que permiten el almacenamiento del contenido de la base de datos, la creación y el mantenimiento de datos, la búsqueda y otras funcionalidades.

Una base de datos puede contener diferentes niveles de abstracción en su

arquitectura. Normalmente, los tres niveles: externo, conceptual e interno conforman la arquitectura de la base de datos. El nivel externo define cómo los usuarios ven los datos. Una sola base de datos puede tener múltiples vistas. El nivel interno define

cómo se almacenan físicamente los datos. El nivel conceptual es el medio de comunicación entre niveles internos y externos. Proporciona una vista única de la base de datos, independientemente de cómo se almacene o vea. Las bases de datos están formadas por tablas y contienen filas y columnas, como las hojas de cálculo en Excel. Cada columna corresponde a un atributo, mientras que cada fila representa un solo registro.

DBMS, a veces llamado simplemente administrador de base de datos, es una colección de programas informáticos que se dedica a la administración (es decir, la organización, el almacenamiento y la recuperación) de todas las bases de datos instaladas en un sistema. Existen diferentes tipos de sistemas de administración de bases de datos en el mundo, y algunos de ellos están diseñados para la administración adecuada de las bases de datos configuradas para propósitos específicos.



Fuente: https://es.wikipedia.org/wiki/Base_de_datos



2.1.4 DBMS Access

Por: Edgar Soc

Un sistema de administración de bases de datos (DBMS) es un software de sistema para crear y administrar bases de datos. El DBMS proporciona a los usuarios y programadores una forma sistemática de crear, recuperar, actualizar y administrar datos. Un DBMS también permite a los usuarios finales crear, leer, actualizar y eliminar datos en una base de datos. El DBMS esencialmente sirve como una interfaz entre la base de datos y los usuarios finales o programas de aplicación, asegurando que los datos estén organizados de manera consistente y permanezcan fácilmente accesibles.

El DBMS gestiona tres cosas importantes: los datos, el motor de la base de datos que permite acceder a los datos, bloquearlos y modificarlos, y el esquema de la base de datos, que define la estructura lógica de la base de datos. Estos tres elementos fundamentales ayudan a proporcionar concurrencia, seguridad, integridad de datos y procedimientos de administración uniformes. Las tareas típicas de administración de la base de datos compatibles con el DBMS incluyen la gestión de cambios, el monitoreo y/o ajuste del rendimiento, así como las copias de seguridad y la recuperación.

Microsoft Access es un software de gestión de base de datos de nivel de entrada. La base de datos de MS Access no sólo es económica, sino también una poderosa base de datos para proyectos a pequeña escala. MS Access utiliza el motor de base de datos Jet, que utiliza un dialecto de lenguaje SQL específico (a veces denominado Jet SQL). MS Access viene con la edición profesional del paquete de MS Office. MS Access tiene una interfaz gráfica intuitiva fácil de usar.

Características

- Los usuarios pueden crear tablas, consultas, formularios e informes y conectarlos con macros.
- Microsoft Access es una base de datos basada en un servidor de archivos. A diferencia de los sistemas de administración de bases de datos relacionales cliente- servidor (RDBMS) Microsoft Access no implementa desencadenadores de base de datos, procedimientos almacenados o registro de transacciones.



2.1.5 Modelo entidad relacion

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976.

El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. Es Modelo Semántico descriptivo del análisis de una Base de Datos.

El Modelo Entidad - Relación permite Modelar las Bases de Datos, desde lo conceptual, utilizando Entidades, Relaciones y Atributos, para después Modelar el esquema Lógico a través de las Tablas, transformando los atributos en campos y las entidades en tablas.

Tipos de relacion

Una a una (1:1). En este tipo de relación, una vez fijado un elemento de una entidad se conoce la otra. Ejemplo: nación y capital.

Una a muchas (1:N). Ejemplo: cliente y pedidos.

Muchas a una (N:1). Simetría respecto al tipo anterior según el punto de visto de una u otra entidad.

Muchas a muchas (N:N). Ejemplo: personas y viviendas.

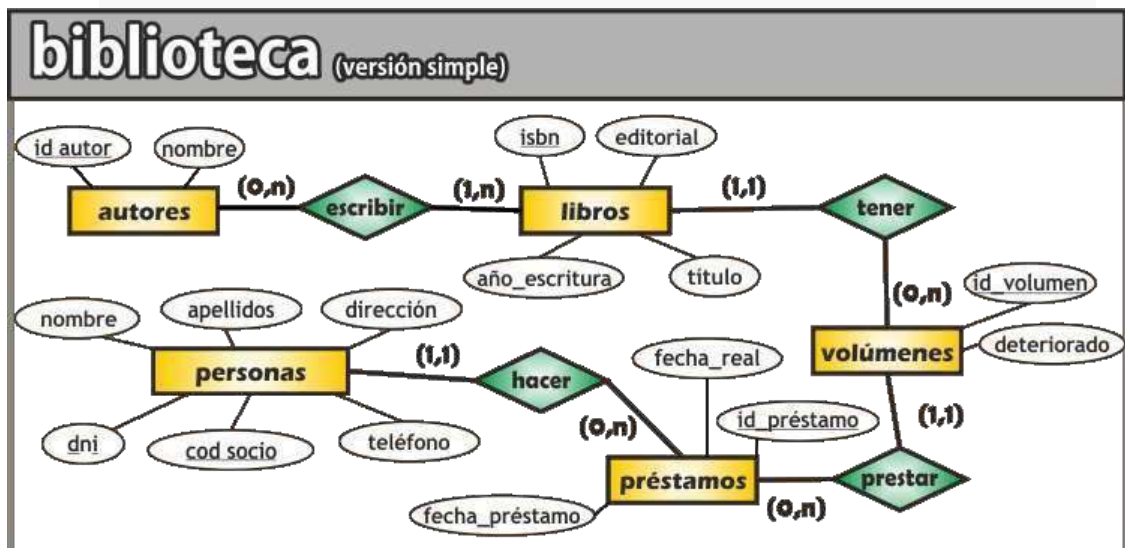


Imagen: Ejemplo entidad relacion(información)

Fuente: <http://formaentic.weebly.com/uploads/1/6/9/2/16924982/559348171.gif>



2.1.5 Reglas

Una Regla de Validación restringe los valores que un usuario puede agregar dentro de un campo en una tabla, consulta o formulario. De esta forma se garantiza el cumplimiento de ciertos requisitos deseados. Cuando este requisito no es satisfecho, el Texto de Validación es el mensaje de error que aparece al tratar de introducir un valor que no cumple con la regla. De esta manera, el usuario puede exigir que un campo de fecha por ejemplo, solo almacene registros posteriores a una determinada fecha; o en un campo de edad se puede restringir la entrada de datos a un rango etario en particular; o en un campo de género se puede limitar a ingresar "Hombre" o "Mujer". Existen en general diversas formas de validación en Access. La primera hace referencia al Tipo de Datos, que establece la clase de valores que se pueden ingresar en un campo de una tabla (como texto, fecha, número, etc.).

Luego viene la Regla de validación, de la cual existen dos tipos: Validación de campo y de registro. En la primera se comprueba el valor del campo al salir de este, y esta validación sólo hace referencia al campo mismo. En cambio, en la validación de registro, la referencia es a otros campos de la misma tabla y se controla cuando se guarda el registro (fila de una tabla).

Pasos para crear una regla de validación

- 2.2 En la vista Diseño de una tabla, seleccione el campo al que desea agregar una Regla de Validación, en la columna Nombre del Campo.
- 2.3 En el cuadro Propiedades del Campo en la sección inferior de la vista, seleccione la pestaña General y luego el campo Regla de Validación (si se pretende crear una regla de validación de registro entonces nos dirigimos al cuadro Hoja de Propiedades).
- 2.4 Luego, agregue la regla de validación directamente o mediante el Generador de Expresiones (pestaña con tres puntos al borde derecho del cuadro).
- 2.5 Finalmente, en el cuadro siguiente, Texto de Validación, agregue el mensaje que desea que aparezca en el cuadro de diálogo de error cuando la regla no es satisfecha.

2.1.7 Relaciones

Una relación en Access le ayuda a combinar datos de dos tablas distintas. Cada relación contiene campos en dos tablas con datos que se corresponden. Por ejemplo, el usuario tiene un campo



Id_Producto en una tabla Productos y en una tabla Detalles de pedido. Cada registro de la tabla Detalles de pedidos tienen un Id_Producto que corresponde a un registro de la tabla Productos con el mismo Id_Producto.

Cuando se usan las tablas relacionadas en una consulta, la relación permite que Access determine qué registros de una tabla se relacionan con qué registros de otra para combinarse en el conjunto de resultados. Una relación también puede impedir que se pierdan datos, al evitar que los datos eliminados dejen de estar sincronizados. Esto se conoce como integridad referencial.

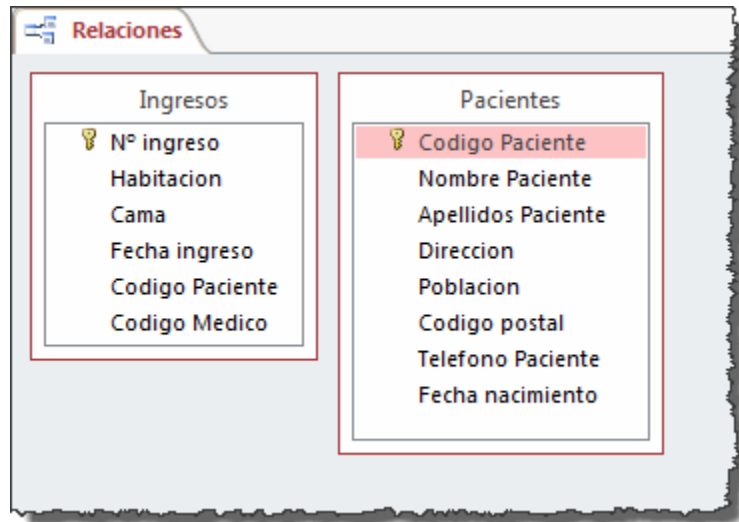


Imagen: Ejemplo de relaciones en Access (información).

Fuente: <https://www.aulacli.com/2016/06/07/relaciones-sin-gif/>

2.1.8 Llave primaria

Una clave principal es un campo o un conjunto de campos con valores únicos en una tabla. Los valores de la clave pueden usarse para hacer referencia a registros completos, porque cada registro tiene un valor diferente para la clave. Cada tabla solo puede tener una clave principal. Access puede crear automáticamente un campo de clave principal al crear una tabla, o puede especificar usted los campos que desea usar como clave principal

Al ser los registros información sobre los atributos de algo o alguien, para no confundirse entre sí se acostumbra a elegir uno de los campos (o a un conjunto de campos) como la clave primaria. Esta clave primaria es la que permite identificar de manera única e



inequívoca un registro. La clave principal no puede contener valores duplicados, ni valores nulos (o en blanco).

Ventajas

- 2.2 Access crea automáticamente un índice para el campo clave principal, esto permite acelerar las búsquedas sobre la tabla.
- 2.3 Cuando se observen los datos ya sea a través de la Hoja de datos o de un formulario, los registros se mostrarán ordenados según la clave principal.
- 2.4 Cuando se adicionen registros, Access no permitirá introducir valores repetidos ni nulos en el campo clave principal, asegurando de esta forma que cada registro sea identificado en forma única.

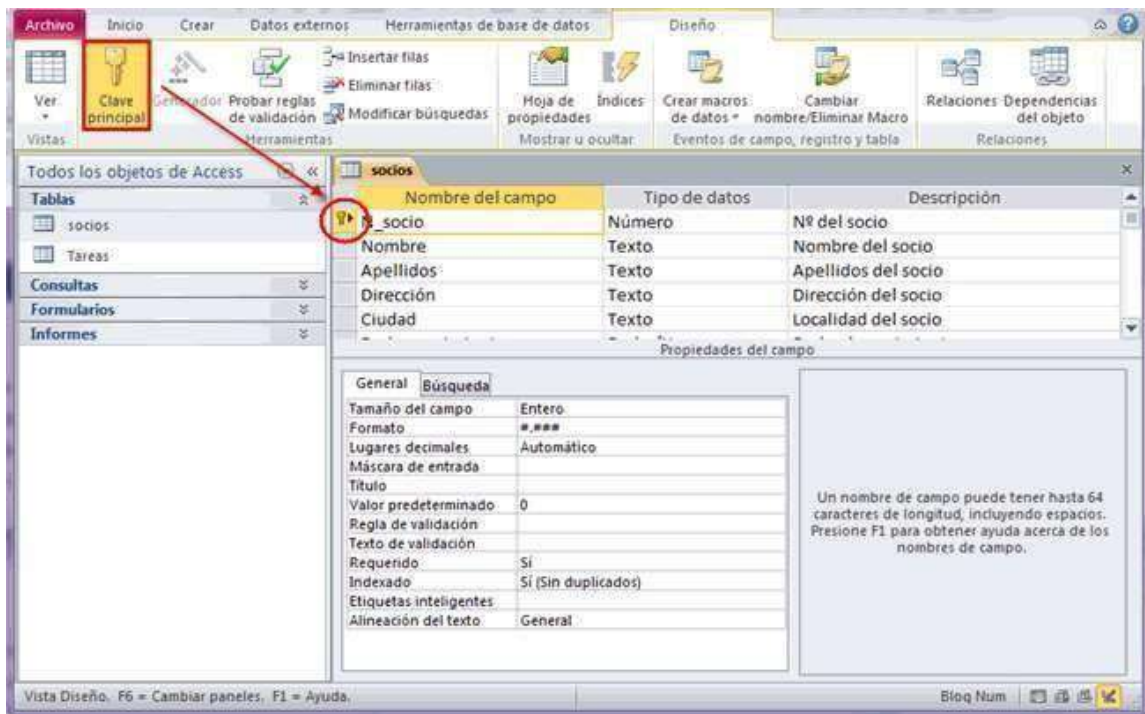


Imagen: Ejemplo de llave primaria en Access (información).

Fuente: https://static.wixstatic.com/media/f36c74_6e7deab2457c43edbdaf1a98a9ee6ac.gif



2.1.9 Llave foranea

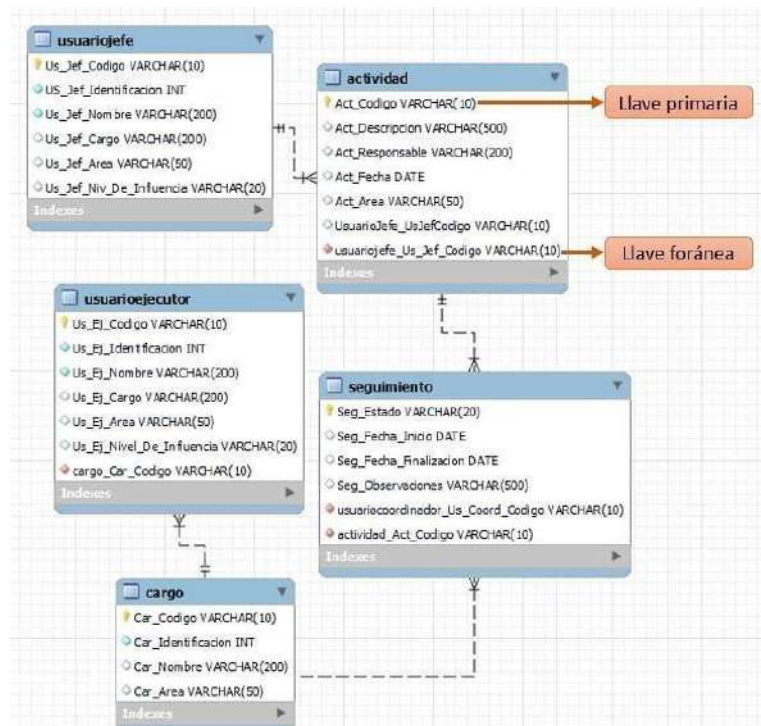
Por: Juan Tiney

llamada clave Externa, es uno o más campos de una tabla que hacen referencia al campo o campos de clave principal de otra tabla, una clave externa indica como esta relacionadas las tablas. Los datos en los campos de clave externa y clave principal deben coincidir, aunque los nombres de los campos no sean los mismos.

Integridad referencial

Estas claves están muy estrechamente ligadas al concepto de integridad referencial, debido a que si una clave foránea contiene un valor, ese valor se refiere a un registro existente en la tabla relacionada. Las columnas referenciadas ya deben contener los valores clave que se insertan en una nueva fila para la tabla de clave foránea. Si no están presentes, no se podrá insertar la fila. Además, todos los valores clave en la tabla de claves foráneas deben borrarse antes de eliminar los valores clave en la tabla a la que se hace referencia.

Se puede crear o eliminar una clave primaria o una clave foránea siempre que sea necesario.



Fuente: https://www.glosarioit.com/Clave_foránea

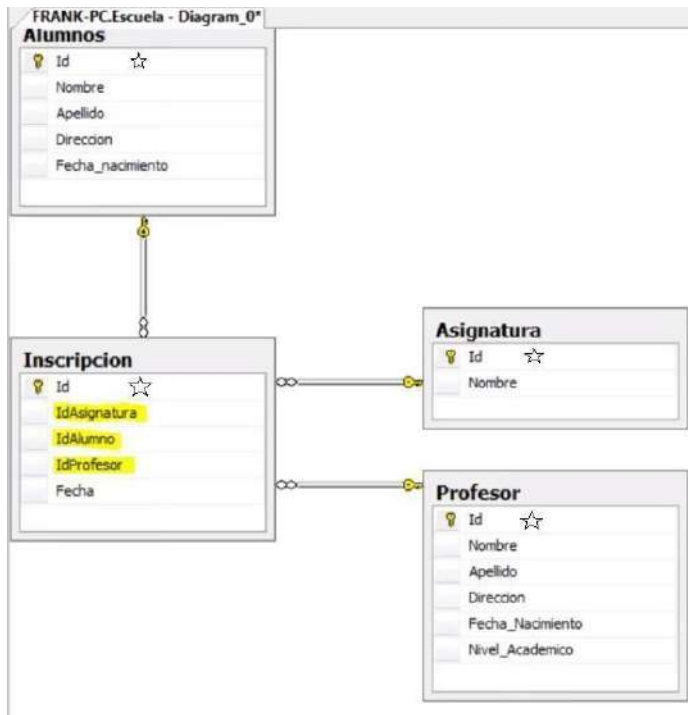
creación de las llaves foráneas

Una **clave foránea** se usa para referirse a otra tabla especificando la referencia y las columnas referenciadas. Tanto las columnas de referencia

como las de referencia se deben asignar entre sí. Las columnas de mapeo deben ser del mismo tipo y longitud. Las columnas a



las que se hace referencia deben ser NOT NULL, pero las columnas de referencia pueden ser NOT NULL o NULL.



fuente: https://www.glosarioit.com/Clave_foránea

2.1.10 Llave candidata

En una relación pueden existir diferentes claves candidatas que se compongan de un número diferente de atributos. - De todas las claves candidatas se elige una que será la Clave Primaria

Objetivo de la llave candidata

- Unicidad: no pueden existir dos registros con el mismo valor en los atributos que forman la llave candidata.
- Irreductibilidad o Minimalidad: no existe ningún otro subconjunto de la llave que cumpla la regla de unicidad.

NO duplicados
NO nulos

núm	grupo	precio
3	A	0.20
1	A	NULL

insert into tabla
values (1,'A',NULL)

Fuentes: <http://dbadixit.com/llaves-candidatas-claves-candidatas>

2.2 Estructura de datos

Las estructuras de datos son una forma de organizar los datos en la computadora, de tal manera que nos permita realizar unas operaciones con ellas de forma **muy eficiente**.

Arrays

Constan de un **índice** para acceder a una posición concreta y del valor que el mismo almacena.

Array :

3	8	1	0	5	-2	32
---	---	---	---	---	----	----

Indices:

0 1 2 3 4 5 6

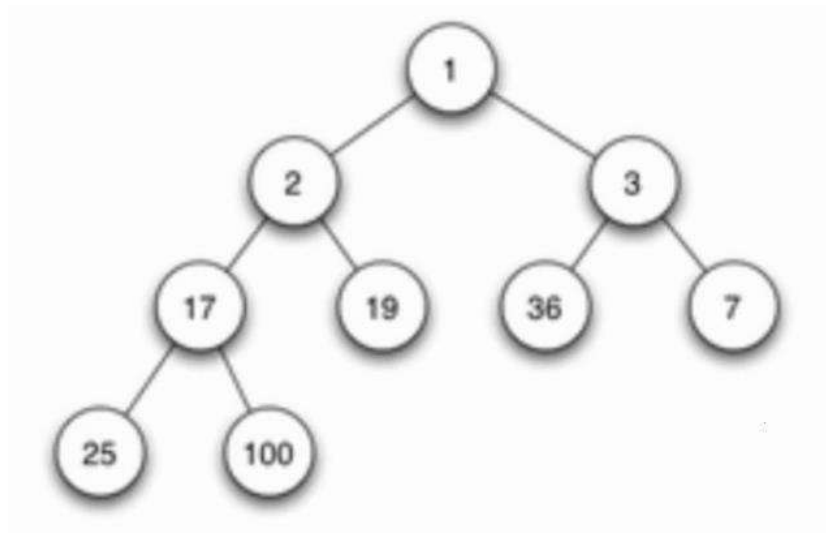
fuentes: <https://openwebinars.net/blog/que-son-las-estructuras-de-datos-y-por-que-son-tan-utiles/>

Montículos binarios

Es una forma de guardar los datos de tal manera, que, aunque no estén ordenados, se puedan retirar de ese conjunto datos **de forma ordenada**.



Esto permite una **gran velocidad**, por ejemplo, a la hora de implementar una cola de prioridades donde queremos que cada elemento que insertemos, si insertamos de repente muchos elementos con una prioridad, el primero que se coja sea el que tenga más o menos prioridad, depende del **tipo de montículo**.



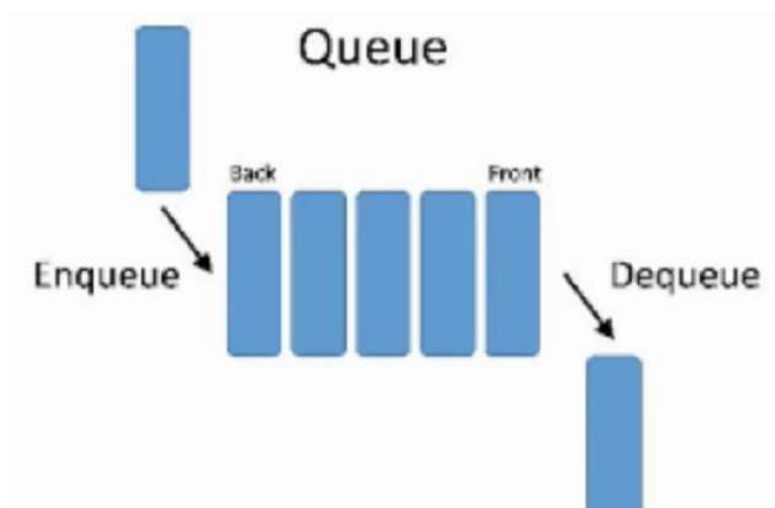
fuelle: <https://openwebinars.net/blog/que-son-las-estructuras-de-datos-y-por-que-son-tan-utiles/>

Pilas

Sirven, por ejemplo, para implementar el **proceso de deshacer**, como cuando escribimos en un editor de texto y pulsamos CTRL+Z, lo que podemos implementar con una pila.

Como vemos en la imagen, tenemos una pila vacía, el bloque 1 sería equivalente a escribir algo. El bloque 2 sería el equivalente a borrar una letra, por ejemplo.

Cuando utilizamos deshacer, lo que haría sería coger la última acción realizada, que tendría una función que haría ciertas operaciones con el hecho de haber borrado una letra y la volvería a poner.

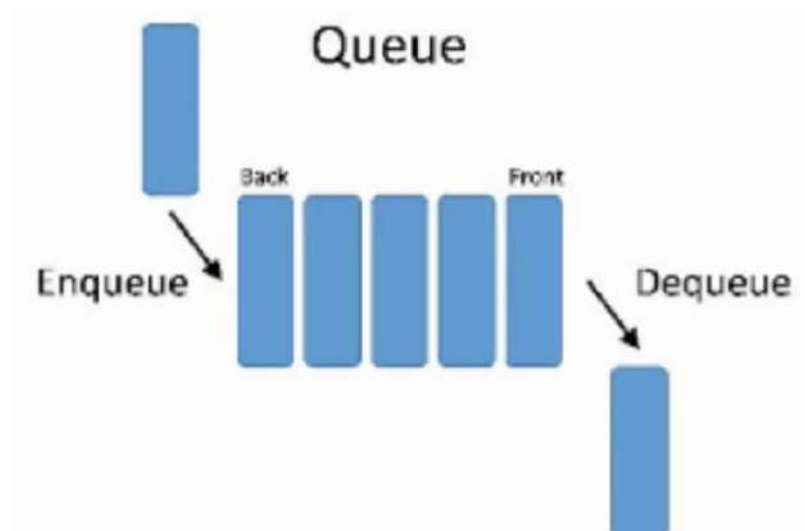


fuente: <https://openwebinars.net/blog/que-son-las-estructuras-de-datos-y-por-que-son-tan-utiles/>



Colas

Es otra estructura de datos muy útil, que sirve, entre otras cosas, para implementar una cola o para comunicar procesos asíncronos.



fuente: <https://openwebinars.net/blog/que-son-las-estructuras-de-datos-y-por-que-son-tan-utiles/>

Para que sirven las estructuras de datos

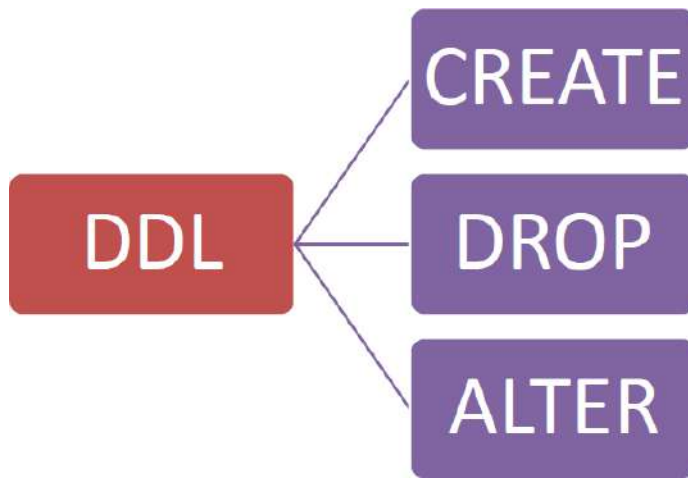
Las estructuras de datos son útiles porque nos permiten tener una batería de herramientas para solucionar ciertos tipos de problemas. Además, nos permiten hacer un software más eficiente optimizando recursos, algo muy útil para IoT y para los entornos que trabajan con Big Data.

2.2.1 DDL

Es un lenguaje de programación para definir estructuras de datos, proporcionado por los sistemas gestores de bases de datos, en este caso PostgreSQL. En inglés, **Data Definition Language**, de ahí sus siglas **DDL**. Te recuerdo que si necesitas un amplio conocimiento en DDL deberías ver nuestro **Curso de Sentencias DDL, DML, DCL y TCL**. Si estás empezando y quieres conocer Postgre, quizás te interese nuestro **Curso de introducción a PostgreSQL**.

Estructuras

- **CREATE**, se usa para crear una base de datos, tabla, vistas, etc.
- **ALTER**, se utiliza para modificar la estructura, por ejemplo añadir o borrar columnas de una tabla.
- **DROP**, con esta sentencia, podemos eliminar los objetos de la estructura, por ejemplo un índice o una secuencia.



fuelle: <https://www.todopostgresql.com/diferencias-entre-ddl-dml-y-dcl/>

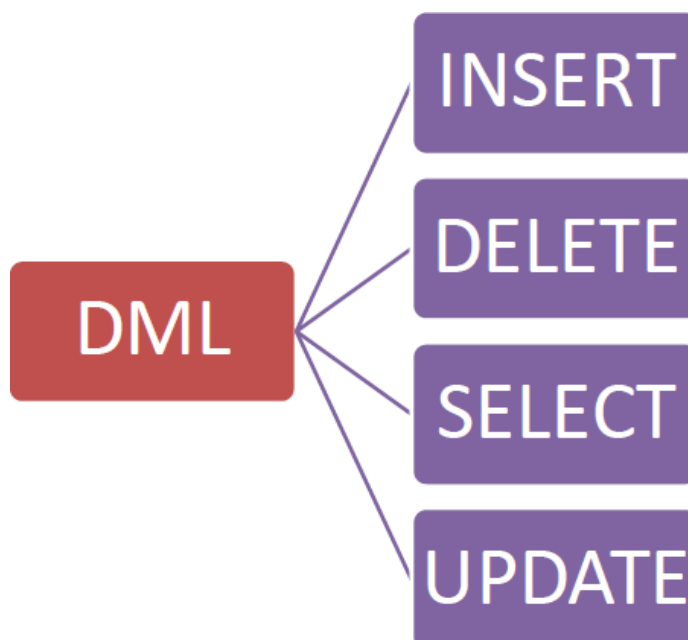
2.2.2 DML

También es un lenguaje proporcionado por los sistemas gestores de bases de datos. En inglés, **Data Manipulation Language (DML)**. Utilizando instrucciones de SQL, **permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación** de los datos que contienen las Bases de Datos.

Los elementos que se utilizan para manipular los datos, son los siguientes:

Elementos

- **SELECT**, esta sentencia se utiliza para realizar consultas sobre los datos.
- **INSERT**, con esta instrucción podemos insertar los valores en una base de datos.
- **UPDATE**, sirve para modificar los valores de uno o varios registros.
- **DELETE**, se utiliza para eliminar las filas de una tabla.



fuente: <https://www.todopostgresql.com/diferencias-entre-ddl-dml-y-dcl/>



2.3 La seguridad informática

Por: Jeremy Tobar

Desde la llegada de Internet y con la transformación digital iniciada en los últimos años, la noción de ciberseguridad se ha convertido en un tema familiar tanto en nuestra vida profesional como personal. La ciberseguridad y las amenazas cibernéticas han estado constantemente presentes durante los últimos 50 años de cambio tecnológico. En las décadas de 1970 y 1980, la seguridad informática se limitó principalmente a la academia hasta la concepción de internet, donde, con una mayor conectividad, los virus informáticos y las instrucciones en la red comenzaron a despegar. Después de la propagación de virus en la década de 1990, la década de 2000 marco la institucionalización de la ciberamenazas y la ciberseguridad.

2.3.1 Concepto

La seguridad informática, la ciberseguridad o la seguridad de la tecnología de la información es la protección de los sistemas informáticos y las redes contra ataques de actores malintencionados que pueden provocar la divulgación de información no autorizada, el robo o el daño del hardware, el software o los datos. Así como de la interrupción o desvío de los servicios que prestan.

El campo ha cobrado importancia debido a la mayor dependencia de los sistemas informáticos, internet, y los estándares de redes inalámbricas como Bluetooth y Wi-Fi, y debido al crecimiento de los dispositivos inteligentes, incluidos los teléfonos inteligentes, los televisores y los diversos dispositivos que constituyen el internet de las cosas.

2.3.2 Vulnerabilidad y ataques

Una vulnerabilidad es una debilidad en el diseño, implementación, operación o control interno. La mayoría de las vulnerabilidades que se han descubierto están documentadas en la base de datos de vulnerabilidades y exposiciones comunes (CVE). Las vulnerabilidades se pueden investigar, realizar ingeniería inversa, cazar o explotar utilizando herramientas automatizadas o secuencias de comandos personalizadas.

2.3.3 Sistema en riesgo

El crecimiento en el número de sistemas informáticos y la creciente confianza en ellos por parte de individuos, empresas, industrias y gobiernos significa que hay un número cada vez mayor de sistemas en riesgo.

- Sistemas financieros

Los sistemas informáticos de los reguladores financieros y las instituciones financieras como la Comisión de Bolsa y Valores de EE. UU. , SWIFT, los bancos de inversión y los bancos comerciales son objetivos importantes de piratería para los ciberdelincuentes interesados en manipular los mercados y obtener ganancias ilícitas.

- Dispositivos de consumo

Las computadoras de escritorio y portátiles suelen ser el objetivo para recopilar contraseñas o información de cuentas financieras o para construir una red de bots para atacar a otro objetivo. Los teléfonos inteligentes , las tabletas , los relojes inteligentes y otros dispositivos móviles , como los dispositivos autónomos cuantificados , como los rastreadores de actividad, tienen sensores como cámaras, micrófonos, receptores GPS, brújulas y acelerómetros que podrían explotarse y recopilar información personal, incluida información de salud confidencial.



Fuente:

https://www.tecnoseguro.com/media/k2/items/cache/210438cfc59c1719a9deba5dc66d5c69_XL.jpg

2.3.4 Protecciones informáticas(contramedidas)

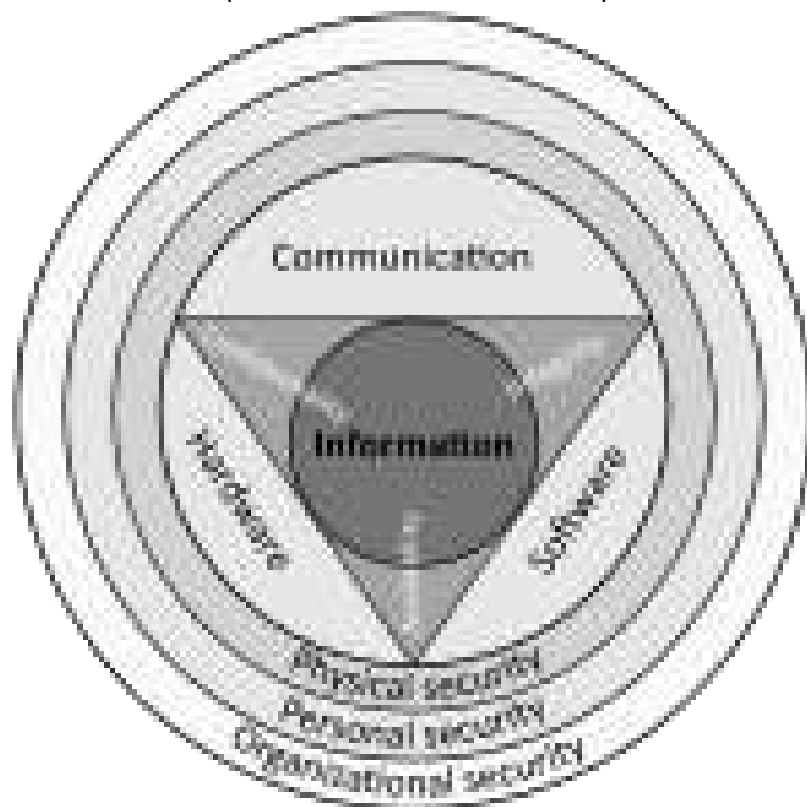
En seguridad informática, una contramedida es una acción, dispositivo, procedimiento o técnica que reduce una amenaza, una vulnerabilidad o un ataque al eliminarlo o prevenirlo, al minimizar el daño que puede causar, o al descubrirlo e informarlo para que se tomen medidas correctivas. Puede ser tomado.

Algunas contramedidas comunes se enumeran en las siguiente secciones:

- Seguridad por diseño
 - El principio de privilegio mínimo , donde cada parte del sistema tiene solo los privilegios necesarios para su función.
 - Demostración automatizada de teoremas para demostrar la corrección de subsistemas de software cruciales
- Medidas de seguridad

Un estado de seguridad informática es el ideal conceptual, alcanzado mediante el uso de los tres procesos: prevención de amenazas, detección y respuesta. Estos procesos se basan en varias políticas y componentes del sistema, que incluyen lo siguiente:

- Los controles de acceso a la cuenta de usuario y la criptografía pueden proteger los archivos y datos de los sistemas, respectivamente.
- Los productos del Sistema de detección de intrusos (IDS) están diseñados para detectar ataques de red en curso y ayudar en el análisis forense posterior al ataque , mientras que los registros y registros de auditoría cumplen una función similar para los sistemas individuales.



Fuente: <https://upload.wikimedia.org/wikipedia/commons/thumb/c/c5/CIAJMK1209-en.svg/150px-CIAJMK1209-en.svg.png>



2.3.4 Como proteger de los virus

Por: Diego Vasquez

Navegar en internet, insertar una memoria USB, CD o DVD en tu computador pueden exponerlo a virus informáticos, programas que afectan el correcto funcionamiento de tu equipo, roban información personal, eliminan u ocultan archivos y más.

Así como hay muchos métodos de infección, también existen formas de prevenir que estos programas maliciosos ingresen a un computador. Conoce algunos ejemplos para evitar que tu equipo sea infectado con virus: Instala un Software antivirus, Configurar un Firewall, No descargar cosas innecesarias, Evita los clics en anuncios publicitarios y Escanea tu computador regularmente.

2.3.5 Medidas de seguridad

Podemos realizar las siguientes recomendaciones: Controles de acceso a los datos más estrictos, una de las principales medidas de seguridad es limitar el acceso a la información. Realizar copias de seguridad, Poseer un sistema de copias de seguridad periódico permite que garantice que puede recuperar los datos ante una incidencia de carácter catastrófico.

2.3.6 Acciones preventivas

Para prevenir los virus informáticos podemos hacer lo siguiente: **Escáner en tiempo real:** Se ejecuta en segundo plano como un servicio del sistema y controla todos los archivos, aplicaciones y servicios. **Analizadores de virus en línea:** Los analizadores en línea comprueban los archivos o todo el equipo a través de Internet. **Escáneres manuales:** La característica especial es la configuración manual del escáner.

2.3.7 Acciones detectivas

Podemos detectar los virus de la siguiente forma: **Velocidad:** los ordenadores van perdiendo velocidad con el tiempo, los componentes envejecen y se deterioran. Si se detecta que repentinamente se ralentiza el sistema es muy posible que sea por causa de un virus. **Mensaje de error:** los mensajes del sistema que advierten de un error pueden indicar que hay un virus. **Buscador:** si notas que el buscador hace cosas que antes no hacía, es probable que sea por causa de un virus.



Imagen: virus informatico

2.3.8 Acciones de restauración

Recuperación de archivos en la copia de seguridad, Para este tipo de recuperación deberás realizar los siguientes pasos:

1. Ve a **"Inicio"**.
2. Selecciona **"Panel de Control"**. Verás que se te abre una lista con todos los elementos que conforman el Panel de control.
3. Selecciona **"Copias de seguridad y restauración"**. Se te abrirá una nueva ventana. Usa la barra de desplazamiento para ir hasta el final de todo de la pantalla.
4. Clica en **"Restaurar mis archivos"**. Se abrirá una nueva ventana.
5. Selecciona o bien **"Buscar"**, **"Buscar archivos"** o **"Buscar carpetas"** (dependerá de la información que se quiera recuperar) y agrega a la lista de recuperación los archivos que quieras recuperar desde el archivo correspondiente. Realizado esto...
6. Dale a **"Siguiente"** para pasar a la ventana de configuración.
7. En la ventana configuración deberás indicar la ubicación dónde quieres que se restauren los archivos. Una vez indicado, dale a **"Restaurar"**.



Imagen: virus informaticos

TEORÍA DE LA
INFORMACIÓN
BIMESTRE



3.1.1. Lenguaje de programación

Por: Byron Aaron Mansilla Fuentes

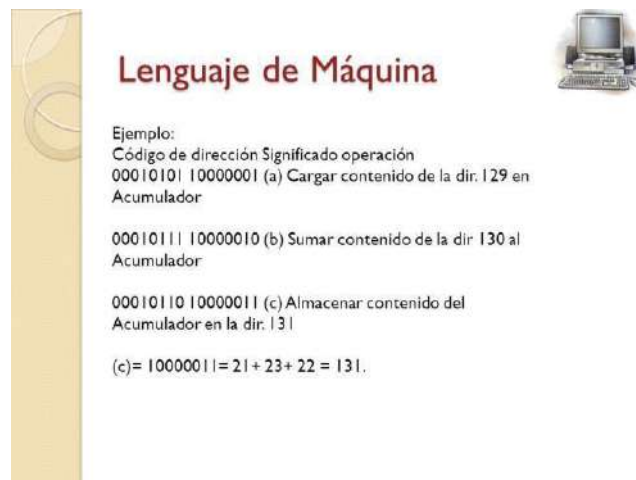
3.1.2. Concepto

Un lenguaje de programación es un lenguaje de computadora que los programadores utilizan para comunicarse y para desarrollar programas de software, aplicaciones, páginas webs, scripts u otros conjuntos de instrucciones para que sean ejecutadas por los ordenadores, especialmente diseñado para describir el conjunto de acciones consecutivas o instrucciones que un equipo informático debe ejecutar.

3.1.3. Clasificación: bajo nivel, alto nivel

Bajo nivel:

- **Código máquina:** El lenguaje de máquina o código máquina es el sistema de códigos directamente interpretable por un circuito micro programable, como el microprocesador de una computadora o el microcontrolador de un autómatas. Está compuesto por un conjunto de instrucciones ejecutadas en secuencia (con eventuales cambios de flujo causados por el propio programa o eventos externos) que representan acciones que la máquina podrá tomar.



Lenguaje de Máquina

Ejemplo:
Código de dirección Significado operación
00010101 10000001 (a) Cargar contenido de la dir. 129 en Acumulador

00010111 10000010 (b) Sumar contenido de la dir. 130 al Acumulador

00010110 10000011 (c) Almacenar contenido del Acumulador en la dir. 131

(c) = 10000011 = 21 + 23 + 22 = 131.

Fuente: <https://slideplayer.es/slide/1824293/>

- **Lenguaje ensamblador:** El lenguaje ensamblador expresa las instrucciones de una forma más natural al programador y a la vez que muy cercana al microcontrolador, ya que cada una de esas instrucciones se corresponde con otra en código máquina. El lenguaje ensamblador trabaja con nemónicos, que son grupos de caracteres alfanuméricos que simbolizan las órdenes o tareas a realizar. La traducción de los nemónicos a código máquina entendible por el microcontrolador la lleva a cabo un programa ensamblador.

- **Lenguaje C/C++:** Es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C y añadir mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido.

C++ ventajas y desventajas

- **Ventajas**
- El lenguaje de programación orientado a objetos.
- El lenguaje muy didáctico, con este lenguaje puedes aprender muchos otros lenguajes con gran facilidad, como C#, Java, Visual Basic, Javascript, PHP, entre otros.
- Es muy potente en lo que se refiere a creación de sistemas complejos, un lenguaje muy robusto.
- Actualmente, puede compilar y ejecutar código de C, ya viene con librerías para realizar esta labor.
- **Desventajas**
- Es mucho más difícil de usar q otros lenguajes de programación.
- Es difícil programar bases de datos
- Los punteros (es muy complicado su sistema).

Fuente: <https://es.slideshare.net/rulvelez/lenguajes-de-programacion-rivas-maldonado>

Alto nivel:

- **PHP:** es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Entre los factores que hicieron que PHP se volviera tan popular es código abierto, esto significa que cualquiera puede hacer cambios en su estructura. En la práctica, esto representa dos cosas importantes:
- es de código abierto, no hay restricciones de uso vinculadas a los derechos.
- está en constante perfeccionamiento, gracias a una comunidad de desarrolladores proactiva y comprometida.

Virtualización

El entorno más profesional lo ofrece la virtualización.

- Ejecutar PHP en un entorno idéntico al de producción
- Disponer de un entorno específico para cada proyecto, sin interferencias con otros proyectos que estés desarrollando
- Compartir la misma configuración, software y librerías entre todos los desarrolladores del mismo proyecto.

Miguel Angel Alvarez
emidesweb

ESCUELA IT

Fuente: <https://desarrolloweb/ventajas-php-entorno-virtual>

- **Python:** es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning. Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

Python (cont.)

Ventajas

- Libre y de fuente abierta.
- Lenguaje de propósito general.
- Gran cantidad de funciones y librerías.
- Sencillo y rápido de programar.
- Multiplataforma.
- Orientado a objetos.
- Portable.

Desventajas

- Lento por ser interpretado.

Fuente: <https://jcstamps.lu/ventajas-de-python-sobre-otros-lenguajes-k.html>

- **JavaScript:** JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales (HTML y CSS) hemos cubierto con mucho más detalle en otras partes del Área de aprendizaje.



Fuente: <https://www.thepowermba.com/es/blog/que-es-javascript-y-para-que-sirve>

- **Visual Basic:** es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo. La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte hasta marzo de 2008.



Fuente: <https://es.slideshare.net/JennyGodoyMaldonado/visual-basic-6-66264948>

- **Cobol:** El lenguaje de programación COBOL surgió de la necesidad de crear un lenguaje más sencillo y universal. Un lenguaje que sirviese para todo tipo de ordenadores, y además que se dirigiese de manera preferente a entornos de empresa. Fue en 1959 cuando se gestó este lenguaje, y a partir de ahí tuvo una gran evolución en la que se han ido

añadiendo diversas funciones.

Programador COBOL 

Funciones <ul style="list-style-type: none">Mantener y desarrollar programas desarrollados en COBOL / MainframesTesting y arreglo de erroresDocumentar requisitos	Formación <ul style="list-style-type: none">Ingeniería informática o educación ITExperiencia previa con COBOL / Mainframes
Habilidades <ul style="list-style-type: none">COBOL, CICS, PL/IBases de datos: SQL Server / DB2Procesos BATCHIBM Mainframe	Salario <ul style="list-style-type: none">Salario EEUU: \$ 40.000Salario España: \$ 55.000Salario Alemania: \$ 80.000

Fuente: <https://www.freelancemap.com/blog/es/que-hace-programador-cobol/>

- **Java:** Java es un lenguaje orientado a objetos, independiente de la plataforma hardware donde se desarrolla, y que utiliza una sintaxis similar a la de C++ pero reducida. Es un lenguaje con una curva de aprendizaje baja (se puede decir que es fácil de aprender) y que dispone de una gran funcionalidad de base (incrementada por la gran cantidad de código de terceros existente). Java, como lenguaje de programación, ofrece un código robusto, que ofrece un manejo automático de la memoria, lo que reduce el número de errores. Java se utiliza en la programación de muchas de las páginas webs más importantes, como Facebook, Amazon o eBay, e incluso algunos videojuegos de éxito como Minecraft están también programados en Java.

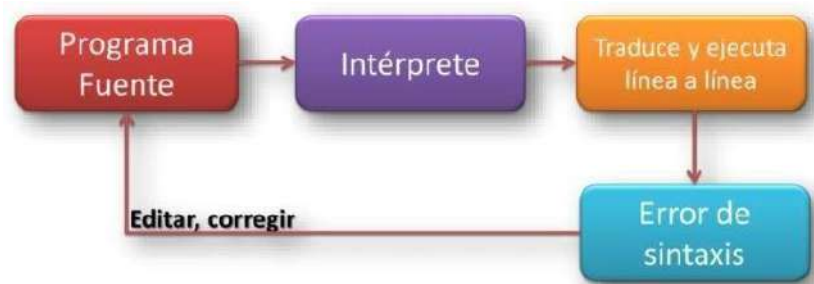
Ventajas y Desventajas

- Retira parte de la responsabilidad sobre punteros
- Multiplataforma gracias a que es interpretado por la JVM
- Facilidad de uso
- Velocidad y uso de recursos (principales desventajas)
- Paso por parámetro, por valor
- Java API

Fuente: <https://es.slideshare.net/alegriadelsol/1-introduccion-java>

3.1.4. Traducción y generación de código

Traducción: la traducción de código se genera mediante un intérprete, un intérprete es a su vez un programa que convierte o traduce un programa, escrito en un lenguaje de programación de alto nivel, instrucción a instrucción o sentencia a sentencia de una forma secuencial, es decir, cada instrucción es traducida una vez traducida la instrucción anterior, así traducción y ejecución se realizan conjuntamente. La principal ventaja de esta forma de trabajar es que la ejecución es inmediata, así pueden corregirse posibles errores que vayan surgiendo durante el proceso y continuar a partir de ese punto.



Fuente: <https://compiladoresugb.wordpress.com/2016/05/31/introduccion-a-los-compiladores-e-interpretes/>

Generación de código: la generación de código es una de las fases mediante el cual un compilador convierte un programa sintácticamente correcto en una serie de instrucciones a ser interpretadas por una máquina. La entrada en esta fase viene representada, normalmente, por

un Árbol Sintáctico, un Árbol de Sintaxis Abstracta, o una Representación Intermedia; la máquina destino puede ser un microprocesador o una

máquina abstracta tal como una máquina virtual o un lenguaje intermedio, legible por un humano. Compiladores más sofisticados realizan múltiples traducciones en cadena (pipelining) con el fin de poder construir código para múltiples plataformas y evitar tener que construir todas las capas del compilador.

En términos más generales, la generación de código es usada para construir programas de una manera automática evitando que los programadores tengan que escribir el código a mano.



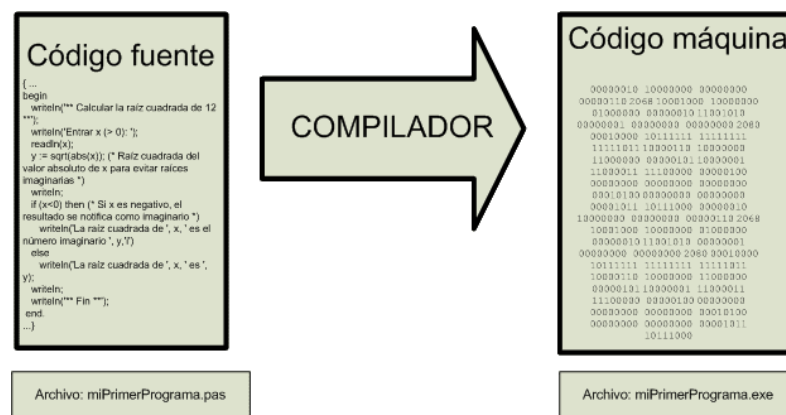
Fuente: <https://fercalan.blogspot.com/2016/06/compilador-e-interprete.html>

3.1.5. Compiladores

Es un Software que traduce un programa escrito en un lenguaje de programación de alto nivel (C / C ++, COBOL, etc.) en lenguaje de máquina. Un compilador generalmente genera lenguaje ensamblador primero y luego traduce el lenguaje ensamblador al lenguaje máquina.

Tipos de compiladores:

- **Compiladores cruzados:** generan código para una plataforma distinta a aquella en la que están funcionando.
- **Compiladores optimizadores:** realizan cambios en el código para mejorar su eficiencia, pero manteniendo la funcionalidad del programa original.
- **Compiladores de una sola pasada:** generan el código máquina a partir de una única lectura del código fuente.
- **Compiladores de varias pasadas:** necesitan leer el código fuente varias veces antes de poder producir el código máquina.
- **Compiladores JIT (just in time):** forman parte de un intérprete y compilan partes del código según se necesitan.



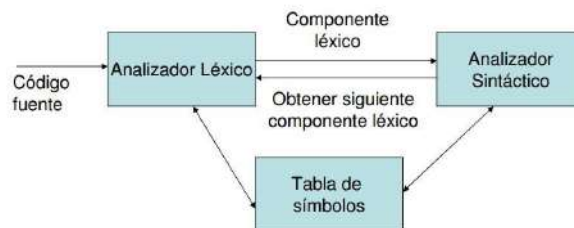
Fuente:

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=895:que-es-un-compilador-c-mejor-ide-o-entorno-de-desarrollo-codelite-codeblocks-geany-kdevelop-cu00506f&catid=82&Itemid=210

3.1.6. Fases de un compilador

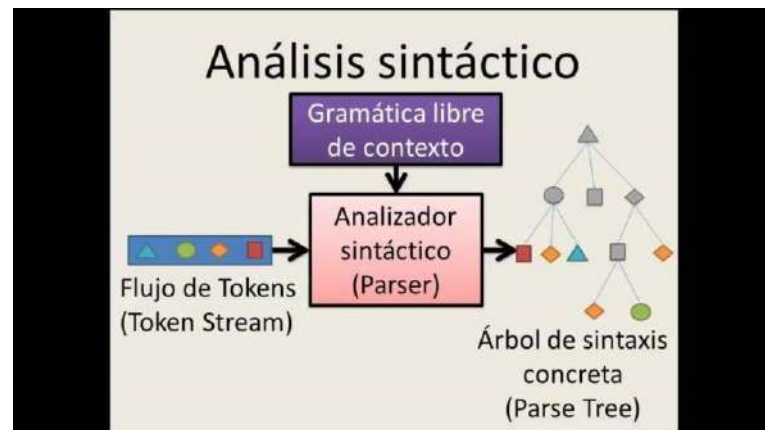
- **Analizador Léxico:** El analizador léxico o lexicográfico (Scanner en inglés) es la primera etapa del proceso de compilación, el cual se encarga de dividir el programa en Tokens, los cuales, según una tabla de símbolos definida por el mismo lenguaje. De esta forma cada token del programa es clasificado según su significado para ser procesados en la segunda etapa del proc

Función del analizador léxico



Fuente: <https://lenguajesyautomatasblog.wordpress.com/2017/05/15/unidad-5-analisis-lexico/>

- **Analizador Sintáctico:** El analizador sintáctico (conocido como parse) es la segunda fase del proceso de compilación y tiene como finalidad la generación de un Árbol sintáctico, el cual no es más que una estructura de datos compleja que permite representar de una forma más simple al programa fuente. Los compiladores modernos utilizan estructuras de objetos para representa a un programa, de esta forma existe una clase específica para representa cada posible token de nuestra tabla de símbolos.



Fuente: <https://www.youtube.com/watch?v=7ptttGTQQmk>

- **Analizador Semántico:** El analizador semántico es el último paso antes de empezar a compilar realmente el código, prepara el programa para ser compilado. El analizador semántico parte del árbol sintáctico abstracto y tiene la finalidad de validar los puntos más finos del programa, como, por ejemplo, validar compatibilidad en tipos de datos, que la variable utilizada en una instrucción este previamente declara o que estén dentro del contexto, si implementamos una interface que todos los métodos estén definidos, etc. El analizador semántico es el que analiza que todo el programa tenga un

significado exacto y que este no pueda fallar en tiempo de ejecución.



Fuente:

https://www.cartagena99.com/recursos/alumnos/apuntes/ININF2_M4_U5_T4.pdf



3.1.7. Interpretes

Por: German Daniel Marquez Estrada

Mientras que el objetivo de los compiladores es obtener una traducción del programa fuente a otro lenguaje, los interpretes tienen como objeto la obtención de los resultados del programa. Para ello deben realizar dos tareas: analizar su entrada y llevar a cabo las acciones especificadas por ella. La parte de análisis puede realizarse de manera idéntica a como se lleva a cabo en los compiladores. Es la parte de síntesis la que se diferencia sustancialmente. En el caso de la interpretación, se parte del árbol de sintaxis abstracta y se recorre, junto con los datos de entrada, para obtener los resultados. En el caso del árbol:



El recorrido consistiría en: Analizar el nodo asignación. Visitar su hijo derecho (la suma) para obtener el valor que hay que asignar:

- Visitar el hijo izquierdo de la suma, recuperar el valor actual de valor.
- Visitar el hijo derecho de la suma, recuperar el valor actual de inc.
- Hacer la suma. Guardar el resultado de la suma en valor.

Actualmente es habitual encontrar híbridos entre la compilación y la interpretación que consisten en compilar a un lenguaje intermedio para una máquina virtual y después interpretar este lenguaje.

Ejemplo:

Java es un lenguaje manejado por este tipo de software, primero se traduce por un compilador-intérprete para generar algo conocido como BYTECODE y posteriormente se interpreta por la máquina virtual de Java.

3.1.8 Fases de un intérprete

Un intérprete funciona "sobre la marcha": recibe el programa (el código que has escrito, por ejemplo, en Python), así como el fichero de datos. Procesa ambos y devuelve la lista de recomendaciones.

Al procesar el código, va haciendo la traducción línea a línea y, al terminar la traducción, inmediatamente la ejecuta. Es decir, la ejecución se hace a la vez que la traducción. Al ser así, la traducción tiene que ser rápida y, por tanto, no puede ser gran cosa: es un trabajo rápido para salir al paso.



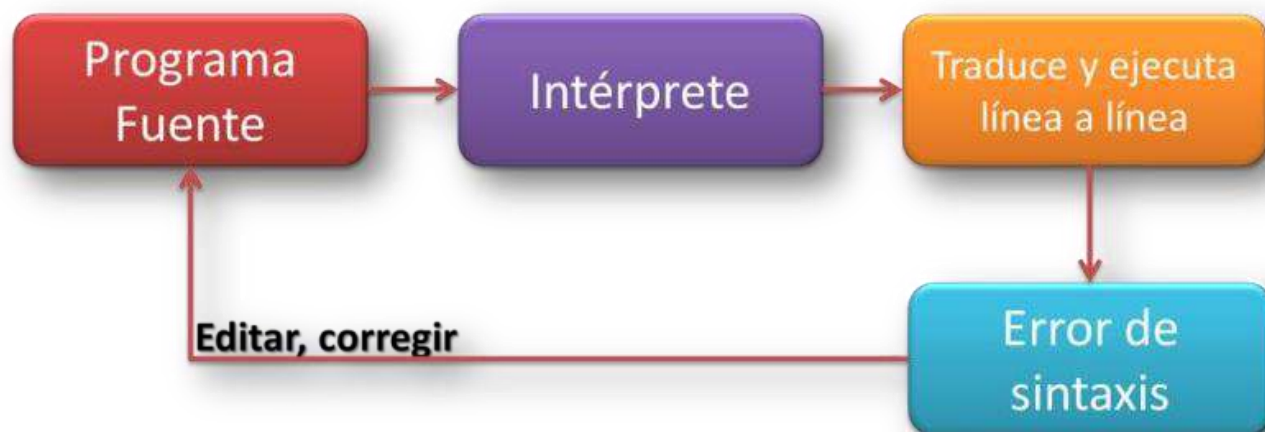


Imagen: Ejemplo de como es el ciclo del interprete (información)

Fuente:

<https://compiladoreseinterpretesugb.files.wordpress.com/2016/05/interprete.png?w=840>

3.1.9. Ciencias auxiliares de la informática.

Las ciencias de la computación son un conjunto de disciplinas que se ocupan de los fundamentos que sustentan la computación: los lenguajes de programación y los propios fundamentos matemáticos y lógicos de esta rama del conocimiento.

1. Inteligencia Artificial

La Inteligencia artificial es el campo científico de la informática que se centra en la creación de programas y mecanismos que pueden mostrar comportamientos considerados inteligentes. En otras palabras, la IA es el concepto según el cual "las máquinas piensan como seres humanos".

2. Arquitectura de Computadoras

La arquitectura de computadoras es el diseño conceptual y la estructura operacional fundamental de un sistema de computadoras. Es decir, es un modelo y una descripción funcional de los requerimientos y las implementaciones de diseño para varias partes de una computadora, con especial interés en la forma en que la unidad central de proceso (CPU) trabaja internamente y accede a las direcciones de memoria.

3. Computación Científica

La computación científica o ciencia computacional es el campo de estudio relacionado con la construcción de modelos matemáticos y técnicas numéricas para resolver problemas científicos, de ciencias sociales y problemas de ingeniería. Típicamente es la aplicación de modelado numérico y otras formas de cálculo de problemas en varias disciplinas científicas.

4. Redes de Computadoras

Es un conjunto de equipos nodos y software conectados entre sí por medio de dispositivos físicos o inalámbricos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios.

Como en todo proceso de comunicación, se requiere de un emisor, un mensaje, un medio y un receptor. La finalidad principal para la creación de una red de ordenadores es compartir los recursos y la información en la distancia, asegurar la con fiabilidad y la disponibilidad de la información.

5. Bases de Datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital. Ejemplo: Acces, DB2, etc.

6. Informática en Salud

La informática en salud o informática médica es la aplicación de la informática y las comunicaciones al área de la salud mediante el uso del software médico, y forma parte de las tecnologías sanitarias. Su objetivo principal es prestar servicio a los profesionales de la salud para mejorar la calidad de la atención sanitaria.

7. Ingeniería de Software

La Ingeniería de Software es una de las ramas de las ciencias de la computación que estudia la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería. Brindando soporte operacional y de mantenimiento, el campo de estudio de la ingeniería de software. Integra ciencias de la computación, ciencias aplicadas y las ciencias básicas en las cuales se encuentra apoyada la ingeniería.

3.1.10. Inteligencia artificial

La Inteligencia Artificial (IA) es la combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano. Una tecnología que todavía nos resulta lejana y misteriosa, pero que desde hace unos años está presente en nuestro día a día a todas horas.

TIPOS DE INTELIGENCIA ARTIFICIAL

Los expertos en ciencias de la computación Stuart Russell y Peter Norvig diferencian varios tipos de inteligencia artificial:

- Sistemas que piensan como humanos

Automatizan actividades como la toma de decisiones, la resolución de problemas y el aprendizaje. Un ejemplo son las redes neuronales artificiales.

- Sistemas que actúan como humanos

Se trata de computadoras que realizan tareas de forma similar a como lo hacen las personas. Es el caso de los robots.

- Sistemas que piensan racionalmente

Intentan emular el pensamiento lógico racional de los humanos, es decir, se investiga cómo lograr que las máquinas puedan percibir, razonar y actuar en consecuencia. Los sistemas expertos se engloban en este grupo.

- Sistemas que actúan racionalmente

idealmente, son aquellos que tratan de imitar de manera racional el comportamiento humano, como los agentes inteligentes.

APLICACIONES PRÁCTICAS DE LA INTELIGENCIA ARTIFICIAL

La IA está presente en la detección facial de los móviles, en los asistentes virtuales de voz como Siri de Apple, Alexa de Amazon o Cortana de Microsoft y está integrada en nuestros dispositivos cotidianos a través de bots (abreviatura de robots) o aplicaciones para móvil, tales como: [Lyli](#)Enlace externo, se abre en ventana nueva., un personal shopper en versión digital; [Parla](#)Enlace externo, se abre en ventana nueva., concebida para ayudarnos con el aprendizaje de idiomas; [Ems](#)Enlace externo, se abre en ventana nueva., diseñada para hacernos un poco más llevadera la ardua tarea de encontrar nuevo piso; o [Gyant](#)Enlace externo, se abre en ventana nueva., un asistente virtual de Facebook que emite 'diagnósticos' médicos. El objetivo de todas ellas: hacer más fácil la vida de las personas.

Los avances en IA ya están impulsando el uso del big data debido a su habilidad para procesar ingentes cantidades de datos y proporcionar ventajas comunicacionales, comerciales y empresariales que la han llevado a posicionarse como la tecnología esencial de las próximas décadas. Transporte, educación, sanidad, cultura... ningún sector se resistirá a sus encantos.

3.1.11. Robótica

La robótica es una disciplina que se ocupa del diseño, operación, manufacturación, estudio y aplicación de autómatas o robots. Para ello, combina la ingeniería mecánica, ingeniería eléctrica, ingeniería electrónica, ingeniería biomédica y las ciencias de la computación, así como otras disciplinas.

La robótica representa la cumbre en la trayectoria de desarrollo tecnológico, es decir, del diseño de herramientas. Su cometido es construir una herramienta que pueda desempeñar muchas de las labores que actualmente desempeña el ser humano, de manera más eficiente y rápida, o en condiciones y ambientes que al ser humano le serían inaccesibles.

El robot es, de algún modo, la herramienta más inteligente posible. Sin embargo, el desarrollo de este tipo de herramientas, ya desde los años de los inicios de la automatización, también se traduce en desempleo y en la sustitución de mano de obra humana por autómatas.

Los robots se clasifican generalmente en base a su pertenencia a las diferentes generaciones de robots construidos, que son:

1. Primera generación. Robots multifuncionales con un sistema simple de control, manual, de secuencia fija o secuencia variable.
2. Segunda generación. Robots de aprendizaje, que repiten secuencias de movimientos previamente ejecutadas por operadores humanos.
3. Tercera generación. Robots de control sensorizado, controlados por algún tipo de programa (software) que envía las señales al cuerpo robotizado para llevar a cabo determinadas tareas mecánicas.

Otra forma de clasificación responde a la estructura del robot, pudiendo hablar de robots:

1. Poliarticulados. Tienen muchas piezas móviles.
2. Móviles. Son de tipo rodante o automotor.
3. Zoomórficos. Imitan la forma de algunos animales.
4. Antropomórficos. Imitan la forma del ser humano.

También existen los robots híbridos, que combinan algunas de las categorías anteriores.

3.1.12. Matemáticas

La programación matemática es una poderosa técnica de optimización utilizada en muchos procesos de toma de decisiones organizacionales. Al igual que otras ramas de la ciencia y la tecnología, la programación matemática utiliza modelos para representar aquellos aspectos de la realidad que tienen impacto en su dominio de interés, en este caso decisiones que optimizan el funcionamiento de un sistema. Hay tres procesos principales que conducen al diseño óptimo del modelo. La primera y más determinante es la identificación de posibles decisiones en el sistema y su representación en forma de variables: variables de decisión. Normalmente estas variables son cuantitativas, aunque también es posible identificar variables cualitativas seleccionadas entre varias posibles.

El segundo proceso implica especificar el conjunto aceptable de valores para las variables de decisión en el sistema, es decir, el conjunto de restricciones que deben obedecer estas variables. Esto se determina teniendo en cuenta las propiedades físicas de los elementos del sistema y sus interrelaciones. El tercer proceso implica desarrollar un modelo de costos del sistema, es decir, identificar los costos/beneficios asociados con cada decisión aceptable. Esto implica idear una función objetivo que asigne a cada conjunto de valores posibles de la variable de decisión su valor de costo/beneficio. Los tres primeros procesos, junto con los datos específicos del sistema, dan como resultado un modelo de optimización.

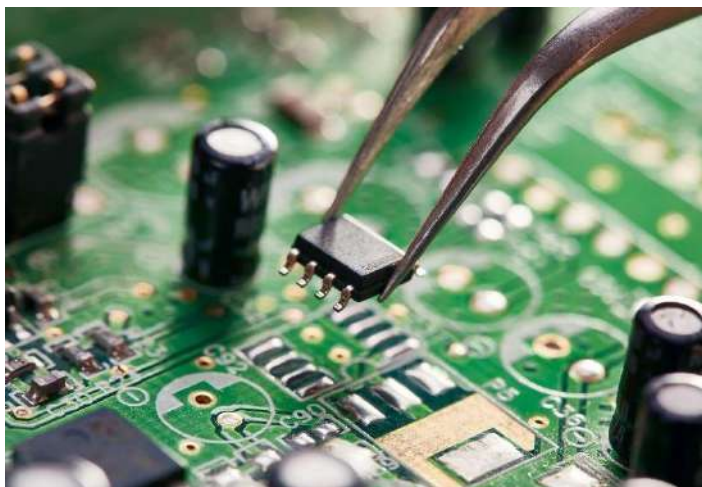
3.1.13. Química

Gracias a la capacidad y velocidad de cálculo de los ordenadores pueden predecirse el comportamiento de las reacciones químicas y las propiedades de los nuevos materiales. Una de las grandes ventajas de la química computacional es el gran ahorro de tiempo y dinero que supone. Repetir experimentos en el laboratorio supone un enorme gasto de reactivos y disolventes, además de grandes quebraderos de cabeza para dar con las condiciones de reacción idóneas. Preguntarle al ordenador suele ser más sencillo. Introduciendo unas cuantas fórmulas y parámetros, los programas de química computacional pueden predecir qué funciona mejor. Y, generalmente, aciertan. La química computacional simula las propiedades de la materia utilizando ordenadores. No sólo sirven para adivinar el futuro. También son muy útiles para explicar qué pasa dentro de los matraces de los químicos experimentales. Muchas veces podemos conocer los reactivos de partida y analizar la composición de los productos pero, ¿cómo se han formado estos últimos a partir de los primeros? Podemos enseñar a los ordenadores las reglas básicas de la química para que, después, calculen cuáles son los intermedios de reacción más probables. Un ejemplo práctico: podemos enseñarle al ordenador unos huevos, harina, azúcar, aceite y un bizcocho. Él será capaz de explicar que para formar el producto (el bizcocho), hemos batido los huevos, hemos mezclado los otros ingredientes, y hemos horneado la mezcla media hora. La química computacional también desentraña los mecanismos de las reacciones, algo fundamental para comprender cómo se comportan las diferentes sustancias y para poder avanzar en el desarrollo de nuevas metodologías y productos.

3.1.14 Electrónica

Por: Elder Martínez

Es una rama de la tecnología que se ocupa del estudio y diseño de sistema electrónicos, que utilizan la manipulación controlada de electrones para procesar información y generar, transmitir y utilizar la energía eléctrica. La electrónica se utiliza en una amplia variedad de aplicaciones, incluyendo la telefonía, la televisión, la computación la medicina, la industria, etc.



En el contexto de la informática, la electrónica permite la interconexión y comunicación de dispositivos a través de redes y sistemas de comunicación

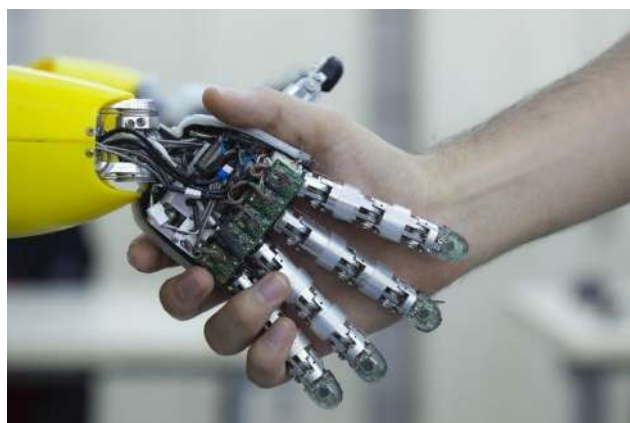
Los componentes electrónicos básicos, como los diodos, transistores y circuitos integrados, son esenciales en la construcción de dispositivos electrónicos, incluyendo computadoras, dispositivos móviles, y otros dispositivos electrónicos de uso cotidiano.

La Electrónica es fundamental en la construcción de sistemas de entrada y salida, como las pantallas y teclados, y en el diseño de sistemas de comunicación y transmisión de datos, como las redes y sistemas inalámbricos.

3.1.15 Mecatrónica

La mecatrónica es una disciplina interdisciplinaria que combina la mecánica, la electrónica y la informática para el diseño de sistemas automatizados y robots. La mecatrónica se utiliza en una amplia variedad de aplicaciones, incluyendo la fabricación, la automoción, la aviación, la robótica entre otras.

La mecatrónica y la informática tienen una relación estrecha y complementaria. La informática es responsable de proporcionar la inteligencia y el control a los sistemas mecatrónicos. Los algoritmos de inteligencia artificial y aprendizaje automático, por ejemplo, se utilizan para permitir que los sistemas mecatrónicos se ajusten y se adapten a su entorno y tareas en tiempo real.



3.1.16 Características

- Control informático: la mecatrónica utiliza sistemas de control informático para monitorear y controlar los procesos mecánicos en los sistemas mecatrónicos.
- Mejora de la productividad y la seguridad: la mecatrónica ha mejorado la productividad y la seguridad en la industria al permitir la creación de robots y sistemas automatizados más avanzados y eficiente.
- Sensores y actuadores: la mecatrónica utiliza sensores y actuadores para recopilar y procesar información sobre el entorno y controlar los procesos mecánicos.
- Automatización: la mecatrónica se autliza para crear sistemas automatizados y robots que realizan tareas repetitivas y precisas con una alta eficiencia y efectividad.

3.1.17 Licenciamiento de software

El licenciamiento de software es el proceso mediante el cual una empresa o desarrollador de software otorga a un usuario final el derecho a usar un software bajo ciertas condiciones y restricciones específicas. El licenciamiento de software protege los derechos de propiedad intelectual del desarrollador y permite que los usuarios finales utilicen el software de manera legal.

Existen diferentes tipos de licencias de software, incluyendo licencias de uso único, licencias de uso múltiples, licencias por suscripción, entre otras. Cada tipo de licencia establece diferentes términos y condiciones para el uso del software, incluyendo el número de usuarios autorizados, la duración de la licencia y las restricciones de distribución y modificación.

Es importante que los usuarios finales lean y comprendan los términos y condiciones de la licencia antes de instalar o usar cualquier software. La infracción de los términos de licencia puede resultar en consecuencias legales y financieras graves para el usuario infractor.

El licenciamiento de software es un proceso esencial que protege los derechos de propiedad intelectual de los desarrolladores de software y permite que los usuarios finales utilicen el software de manera legal y segura.



3.1.18 Tipo de licencia: privada, copyleft, GPL, Free Ware, Shareware.

Licencia privada: Es un tipo de licencia de software que restringe el uso y la distribución del software a un número limitado de usuarios autorizados. Este tipo de licencia es propietaria, lo que significa que el código fuente del software es propiedad exclusiva del desarrollo y no está disponible para el público en general.

Suelen establecer restricciones sobre el número de usuarios que pueden utilizar el software, la duración de la licencia y las condiciones para la distribución. Los usuarios autorizados pueden utilizar el software para sus propios fines, pero generalmente no tienen permiso para modificar el código fuente ni distribuir el software a terceros.

Es importante que los usuarios finales comprendan los términos y condiciones de la licencia antes de instalar o usar el software.



<https://siaguanta.com/c-tecnologia/software-propietario/>

Licencia Copyleft: Es un tipo de licencia de software que permite la distribución y la modificación libres del software siempre y cuando el software resultante siga estando bajo la misma licencia. Esta licencia se utiliza para proteger el software libre y asegurar que siga siendo libre para siempre.

Con una licencia copyleft, los usuarios tienen derecho a utilizar, modificar y distribuir el software de acuerdo con los términos de la licencia. Sin embargo, cualquier software derivado también debe estar disponible bajo los mismos términos y condiciones.

La licencia copyleft es una forma de garantizar que el software libre continúe siendo libre y accesible para todos, y se utiliza comúnmente en

proyectos de software de código abierto.



copyleft

<https://sites.google.com/site/gbicopyright/-cual-es-el-marco-legal-de-los-derechos-de-autor/-que-es-la-sociedad-copyleft>

Licencia GPL: la licencia pública GPL es una licencia copyleft comúnmente utilizada para proteger el software libre y asegura que siga siendo libre para siempre. La GPL permite a los usuarios utilizar, modificar y distribuir el software libremente, siempre y cuando cumplan con ciertas condiciones.

Las condiciones de la GPL incluyen la obligación de hacer disponible el código fuente del software a cualquier persona que lo reciba, así como la libertad de modificar y distribución el software y sus derivados siempre y cuando también estén debajo de los términos de la GPL.



<https://es.vecteezy.com/arte-vectorial/9032410-gpl-logo-gpl-letter-gpl-letter-logo-design-iniciales-gpl-logo-linked-with-circle-and-mayusculas-monograma-logo-gpl-typography-for-technology-business-and-real-marca-inmobiliaria>

Licencia Free ware: se refiere al software que es gratuito y puede ser descargado, usado y distribuido libremente. Esto difiere de otros tipos de software gratuito como el software de código abierto, ya que el software de free ware no requiere que el código fuente esté disponible para los usuarios. Sin embargo algunos programas de free ware pueden incluir limitaciones en cuanto a su uso comercial.



<https://propulsatuweb.com/que-es-freeware/>

Licencia Shareware: Se refiere a un modelo de distribución de software en el cual el usuario puede descargar y probar el software antes de decidir si desea comprarlo o no. Estos programas suelen incluir un período de prueba limitado o funciones limitadas hasta que se compre una licencia completa.

Shareware
software

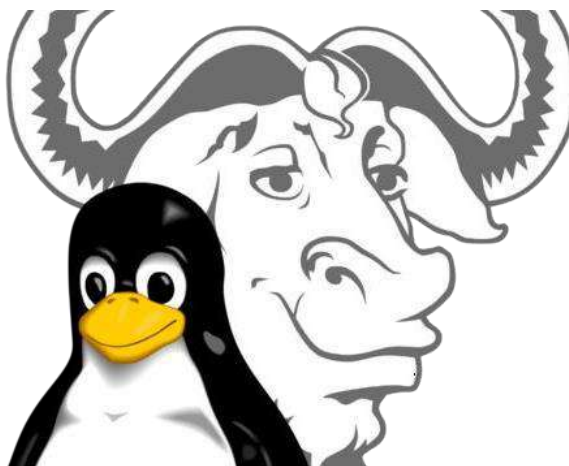
<https://www.timetoast.com/timelines/tipos-de-software-0c0c5ed6-836e-426a-aa0c-ee4cb5f1fa16>

TEORÍA DE LA INFORMACIÓN Bimestre IV

4. Linux

Por: Eduardo Velásquez

Linux es un sistema operativo Open Source diseñado por Linus Torvalds, Linux se lanzó en virtud de la Licencia Pública General GNU, por lo que cualquier persona puede ejecutar el software, estudiarlo, modificarlo y compartirlo. El código modificado se puede redistribuir y vender si así se desea, pero se debe hacer con la misma licencia.



Fuente: <https://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/Gnulinix.svg/1200px-Gnulinix.svg.png>

4.1.1 Historia

Linus Torvalds lo creó y diseñó mientras estudiaba en la universidad de Helsinki, en Finlandia, en el año 1991, se basó en un sistema ya existente llamado Unix, denominado MINIX.

MINIX fue creado en 1987 por Andrew Tannenbaum que era profesor de ciencias de la computación de la Universidad Libre de Ámsterdam en Holanda, principalmente se utilizó para el estudio de sistemas operativos y por lo tanto no tenía muchos recursos ni posibilidad de expansión para la época.

Entonces el sistema operativo que había creado Linus no tenía muchas funciones que las personas necesitaban, Linus frustrado con el esquema MINIX, lo resolvió creando un nuevo sistema, desarrollando un núcleo en el ordenador que rodaba el propio MINIX, usando el compilador GNU C que hoy en día aún se utiliza para los sistemas Linux.



Fuente: <https://grupobcc.com/wp/wp-content/uploads/2015/05/Linus-Torvald-speaker-keynote-linus-conferencias- e1598521766183.jpg>

4.1.2 Distribuciones

Las distribuciones de Linux se basan en el núcleo Linux e incluye determinados paquetes para ofrecer algunas características o funciones al usuario. Por lo tanto, una distribución de Linux es un software que contiene otros paquetes de software, que son de carácter libre y añaden controladores o aplicaciones para satisfacer las necesidades de los usuarios.

Hoy en día existen un montón de distribuciones de Linux. Cada una de ellas va a encajar mejor dependiendo de cada tipo de usuario y entre las distribuciones de Linux más populares tenemos:

- **Debian:** Es una de las distribuciones más antiguas y populares de Linux.
- **Ubuntu:** Es popular debido a su fácil proceso de instalación.
- **Linux Mint:** Es una distribución rápida, ligera y sencilla de manejar.
- **Elementary OS:** Contiene una interfaz gráfica cuidada y muy parecida a macOS.
- **Fedora:** Utiliza una interfaz GNOME y es fácil de utilizar.



Fuente: <https://geekland.eu/wp-content/uploads/2017/09/son-necesarias-las-distribuciones-Linux-1280x720.jpg>

4.1.3 Particionado del disco duro para la instalación de Linux

Las particiones son básicamente divisiones de almacenamiento en mismo disco duro y esto nos permitirá instalar más de un sistema operativo y para ello.

Las particiones que estén en uso no se pueden editar ni redimensionar, así que se tienen que desmontarlas primero. Una vez desmontadas se pueden editar o eliminarlas. Para crear una nueva partición en un espacio libre debemos seleccionar el lugar de la nueva partición, su tamaño y formato en este caso debe de ser ext3 o ext4 para ser compatible con Linux. Una vez creada la partición del disco duro ya podremos instalar cualquier distribución de Linux que queramos.

El gestor de arranque es el que se encarga de dar paso al sistema operativo cada vez que se encienda el ordenador. En caso que tengamos más de un sistema operativo instalado debemos configurar el gestor de arranque para que te pregunte que sistema

debe arrancar.





Fuente: <https://yourgeekweb.com/wp-content/uploads/2018/10/particionar-disco-duro-desde-windows.pngx>



4.1.4 Aplicaciones de Linux

Por: Ricardo Ventura

4.1.4.1 Procesadores de texto

Los procesadores de texto permiten generar tanto 'documentos de texto plano ASCII' como 'documentos binarios'

Los procesadores de texto son la herramienta esencial con la cual hacer trabajos y documentos de texto, pudiendo incluir imágenes, diagramas, y dando formato a los párrafos, añadiéndoles colores, alineaciones, bordes, numeración de páginas y lo que necesites en cada momento.

Algunos de estos son:

- OpenOffice.org Writer: procesador de textos de OpenOffice.org.
- Abiword : procesador de textos de GNOME.
- Kword : procesador de textos de KDE.

4.1.4.2 Visores de imágenes

Los visores de imágenes son útiles cuando sólo desea ver unas pocas fotos u organizar rápidamente una carpeta sin necesidad de utilizar un software de gestión de imágenes complejo.

Algunos visores de imágenes y un ejemplo:

- PhotoQt



Fuente: <https://vidatecno.net/6-visores-de-imagen-qt-ligeros-para-linux/>

- PhotoKit
- Fototónico
- Limoo
- Limoo
- Qiviewer

4.1.4.3 Reproductores de audio y video

Son programas que permiten reproducir archivos multimedia, es decir, archivos de audio o video.

Tenemos algunos ejemplos de los reproductores

- VLC



Fuente: <https://filmora.wondershare.es/video-player/10-open-source-video-player-for-linux-windows-and-mac.html>

- Miro
- MPV
- Kod

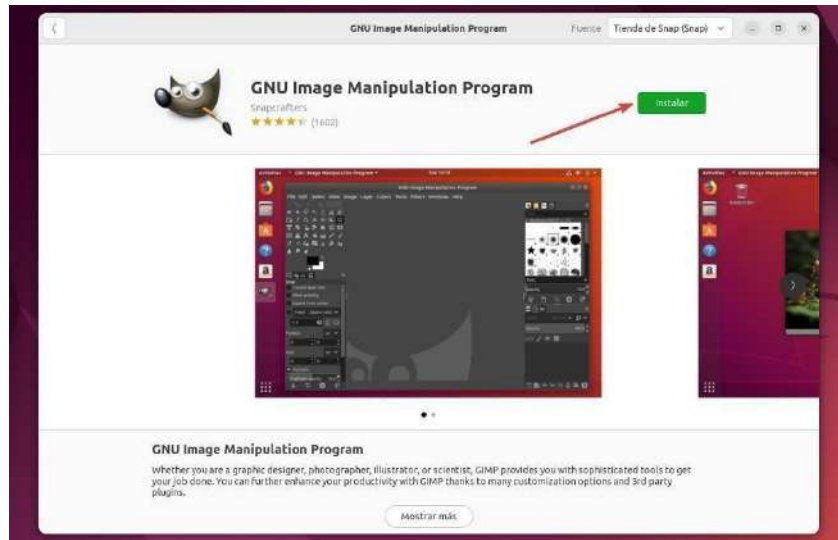
4.1.5 Instalación de software

En Linux podemos instalar aplicaciones utilizando la interfaz gráfica o la línea de comandos. Encontrar una aplicación que funcione tanto a través de la línea de comandos como haciendo uso de una interfaz gráfica es un caso muy raro.

4.1.5.1 Interfaz gráfica

El método más fácil y rápido para instalar aplicaciones en Linux pasa por abrir la tienda de aplicaciones de la distribución Linux que tengamos instalada en nuestro equipo, buscar la aplicación y pulsar en el botón descargar.

Ejemplo de la interfaz:



Fuente: <https://www.softzone.es/linux/tutoriales/descargar-instalar-aplicaciones-linux/>

4.5.1.2 Línea de comandos

Para instalar aplicaciones a través de la línea de comandos, debemos realizar 2 o 3 pasos dependiendo de si esta se encuentra en los repositorios a los que tiene acceso la distribución Linux que tengamos instalada.

lo primero que debemos hacer es añadir el repositorio donde se encuentra la aplicación, utilizando el siguiente comando:

```
sudo add-apt-repository direccióndelrepositorio
```

Fuente: <https://www.softzone.es/linux/tutoriales/descargar-instalar-aplicaciones-linux/>

actualizamos el gestor de paquete

```
sudo apt-get update
```

Fuente: <https://www.softzone.es/linux/tutoriales/descargar-instalar-aplicaciones-linux/>

Y, finalmente instalamos la aplicación utilizando el comando:

```
sudo apt-get install nombreakplicacion
```

Fuente: <https://www.softzone.es/linux/tutoriales/descargar-instalar-aplicaciones-linux/>

4.5.1.3 Tipos de paquetes

Cada tipo de paquetes está destinado a ser utilizado en diferentes versiones de Linux. Por ejemplo, un paquete de Debian, no se puede instalar en Red Hat, al igual que un paquete de Arch Linux tampoco se puede instalar en Ubuntu o Debian, para su solución algunos tipos de paquetes son compatibles con todos los sistemas Linux.

Los tipos de paquetes son:

- Flatpak / Snap
- Appliance
- RPM / Yum
- DPKG / Apt-get / Apt
- Pacman

4.5.1.4 Desde repositorios

Los repositorios son como supermercados, donde tenemos a nuestra disposición una amplia gama de aplicaciones.

4.5.1.5 Vía web

También tenemos la posibilidad de descargar aplicaciones que no se encuentran a través de las demás opciones, aunque cada vez se utiliza menos este método porque muchos usuarios lo consideran anticuado

4.1.6 Bash

Es una herramienta poderosa para todo usuario de Linux o administrador de sistemas.

Bash está disponible en casi todas las versiones de Linux y no requiere de instalación adicional

#!/bin/bash

Es necesario permitir que el shell interactivo sepa qué tipo de intérprete ejecutar para el programa que contiene. La primera línea le dice a Unix que el archivo será ejecutado por /bin/bash.

zach@marigold:~\$./miScriptBash.sh

```
zach@marigold:~$ ./miScriptBash.sh
Hola Mundo!
```

Fuente: <https://www.freecodecamp.org/espanol/news/la-guia-definitiva-de-linea-de-comandos-de-linux-tutorial-completo-de-bash/>

El script solo contiene dos líneas. La primera indica qué intérprete se debe usar para ejecutar el archivo (en este caso, bash.) La segunda línea es el comando el cual queremos usar, echo, seguido por lo que queramos imprimir, es decir "!Hola Mundo!".

Bash cat

Cat es usado para leer un archivo secuencialmente e imprimirlo en la consola de manera estándar. Su nombre es derivado de su función para concatenar archivos.

```
cat [opciones] [nombres_de_los_archivos]
```

Fuente: <https://www.freecodecamp.org/espanol/news/la-guia-definitiva-de-linea-de-comandos-de-linux-tutorial-completo-de-bash/>

Bash cd

Hay unos pocos, pero realmente útiles argumentos que ayudan a su uso:

- hace referencia al directorio actual, tal como **./proyectos**
- puede ser usado para subir un directorio, usa **cd ..**

Bash head

Head se utiliza para imprimir las primeras diez líneas (por defecto) o cualquier otra cantidad especificada de líneas de un archivo o archivos, El comando head te permite ver las primeras N líneas de un archivo.

```
head [opciones] [nombre-de-archivo(s)]
```

Fuente: <https://www.freecodecamp.org/espanol/news/la-guia-definitiva-de-linea-de-comandos-de-linux-tutorial-completo-de-bash/>

Bash ls

ls es un comando en sistemas operativos Unix para listar el contenido de un directorio, por ejemplo, nombres de carpetas y archivos.

```
ls [opciones] [nombre_de_los_archivos]
```

Fuente: <https://www.freecodecamp.org/espanol/news/la-guia-definitiva-de-linea-de-comandos-de-linux-tutorial-completo-de-bash/>

Bash man

Man, o abreviación de manual, es un comando de bash usado para desplegar referencias de manuales en línea de determinado comando.

```
man [opciones] [comando]
```

Fuente: <https://www.freecodecamp.org/espanol/news/la-guia-definitiva-de-linea-de-comandos-de-linux-tutorial-completo-de-bash/>

Bash mv

El primer argumento es el archivo el cual quieres mover, y el segundo es la ubicación a ser movido.

```
mv origen destino
mv origen ... directorio
```

Fuente: <https://www.freecodecamp.org/espanol/news/la-guia-definitiva-de-linea-de-comandos-de-linux-tutorial-completo-de-bash/>

```
mv source_file destination_folder/

mv command_list.txt commands/
```

Fuente: <https://kinsta.com/es/blog/linux-comandos/>

4.1.7 Sistema de usuario, manejo de clave de archivos y permisos

4.1.7.1 Tipos de Usuario

Usuario administrador: los usuarios linux root tiene la posibilidad de realizar todo tipo de labores respecto a la administración del sistema. Los cambios o modificaciones realizados con este usuario pueden causar

inconvenientes en la seguridad del sistema completo.



Demás usuarios: los otros tipos de usuarios en Linux normales o no administradores, son aquellos que pueden utilizar las herramientas del sistema con normalidad, pero que no pueden ejecutar labores administrativas.

Para poder realizar el cambio de usuario normal a usuario root, se necesitarán permisos de administración, y se puede acceder ya sea proporcionando la contraseña del usuario root e iniciando sesión, o establecer una lista de usuarios que pueden tener acceso como administradores.

4.1.7.2 Grupos en Linux

Los grupos en Linux son usados principalmente para otorgar permisos sobre archivos y directorios. es decir, para gestión de usuarios Linux.

Grupo principal: tiene la función de determinar quién será el grupo que posea los archivos creados por el usuario.

Grupos secundarios: los grupos suplementarios son aquellos grupos adicionales a los que el usuario pertenezca.

4.1.7.3 Comandos para gestión de usuarios y grupos de Linux

- adduser o useradd:** permite la creación de nuevos usuarios, es decir, la administración de usuarios linux.
- usermod, chfn, chsh y chage:** usados para la modificación de un usuario.
- passwd:** para cambiar la contraseña de un usuario
- deluser o userdel:** comando para la eliminación de usuarios.
- whoami:** para saber qué usuario somos.
- su:** comando para cambiar de usuario.

4.2 Tecnologías de la Información

Por: Mario Ricardo Yoc Hernandez

Las tecnologías de la información y comunicación, conocidas por su abreviatura como TIC, son un conjunto de tecnologías requeridas para el almacenamiento, recuperación, proceso y comunicación de la información.

4.2.1 Historia

La información y comunicación datan de tiempos prehistóricos. Estas formas han evolucionado con la creación de nuevas tecnologías, que facilitaron el intercambio de la información. Sin embargo, fue hasta los años 70 cuando inició la "era digital"; los avances científicos en el campo de la electrónica causaron el impulso de las TIC, que combinaban esencialmente la electrónica con el software.

Ya en los años 90, las investigaciones desarrolladas permitieron la convergencia de la electrónica, la informática y las telecomunicaciones, lo que hizo posible la interconexión entre redes, y fue así que surgieron las TIC tal como ahora las conocemos.

4.2.2 Características

Algunas de las características fundamentales de las TIC son:

- Inmaterialidad, su materia prima es la información en múltiples códigos y formas.
- Interconexión, aunque se presentan de forma independiente, pueden combinarse y ampliar sus conexiones.
- Interactividad, permite la interacción del sujeto con la máquina y, así, la adaptación de éstas a diversas áreas educativas y cognoscitivas de las personas.
- Instantaneidad, facilita la rapidez de acceso e intercambio de la información.
- Calidad de imagen y/o sonido, lo que da fiabilidad y fidelidad a la información transferida.
- Penetración en diversos sectores de la sociedad: salud, educación, economía, etc.



Imagen: Tecnologías de la Información Fuente:

<https://www.google.com/search?q=1.1%09Tecnologias+de+la+Informaci%C3%B3n+&tbm=isch&ved=2ahU>

[K EwjBt5ip_4n9AhWQVt8KHZamD6QQ2-
cCegQIABAA&og=1.1%09Tecnologias+de+la+Informaci%C3%B3n+&gs_lcp=CgNpbWcQAzoECCMQJzoEC
AQQzoFCAAQgARQhRRYhRRgoRdoAHAAeACAAbkBiAGuApIBAzAuMpgBAKABAaoBC2d3cy13aXotaW1n
wAEB&scient=img&ei=oLfiY8HnA5Ct_QaWzb6gCg&bih=635&biw=1366&rlz=1C1ALOY_esGT1021GT1021#im
grc
=C46D5Bbu6Bx3mM](https://www.google.com/search?q=K+EwjBt5ip_4n9AhWQVt8KHZamD6QQ2-cCegQIABAA&og=1.1%09Tecnologias+de+la+Informaci%C3%B3n+&gs_lcp=CgNpbWcQAzoECCMQJzoEC
AQQzoFCAAQgARQhRRYhRRgoRdoAHAAeACAAbkBiAGuApIBAzAuMpgBAKABAaoBC2d3cy13aXotaW1n
wAEB&scient=img&ei=oLfiY8HnA5Ct_QaWzb6gCg&bih=635&biw=1366&rlz=1C1ALOY_esGT1021GT1021#im
grc
=C46D5Bbu6Bx3mM)

4.2.1 Internet y sus Servicios

El internet es más que una plataforma para el intercambio de datos: es la red mundial que permite el acceso a la información desde cualquier parte del mundo y hace posible la comunicación desde distintos lugares sin la necesidad de estar frente a frente. Todo esto sucede en un lapso casi instantáneo, lo que facilita el intercambio y la obtención de información prácticamente inmediatos.

4.2.1.1 Servicios

Internet nos abre las puertas a miles y miles de distintos servicios, entre ellos nos permitirá:

- Buscar cualquier tipo de información que necesite (por ejemplo en Google).
- Comprar productos de diversa índole (por ejemplo en Amazon o Mercado Libre).
- Comunicarnos con familiares o amigos que estén en otros países o ciudades mediante una vídeo llamada (por ejemplo de Skype o Whatsapp).
- Jugar juegos online (como el League of Legends) con personas de distintas nacionalidades y edades, en tiempo real.

4.2.2 Extranet

Una extranet es una red privada que utiliza protocolos de Internet, protocolos de comunicación y probablemente infraestructura pública de comunicación para compartir de forma segura parte de la información u operación propia de una organización con proveedores, compradores, socios, clientes o cualquier otro negocio u organización. Se puede decir en otras palabras que una extranet es parte de la Intranet de una organización que se extiende a usuarios fuera de ella, usualmente utilizando Internet y sus protocolos.

4.2.2.1 Etimología

La palabra extranet (también extrarred) es una contracción de la frase extraconnected network (red extraconectada).

4.2.2.2 Características

La extranet suele tener un acceso semiprivado. Es por ello que una extranet requiere o necesita un grado de seguridad, para que no pueda acceder cualquier persona. Otra característica de la extranet es que se puede utilizar como una Intranet de colaboración con otras compañías.

4.2.2.3 Creadores

Pero, el término de la extranet fue definido por el que fuera primer ejecutivo de Netscape Communications Corporation Jim Barksdale y el cofundador de dicha empresa Mark Andreessen.

4.2.3 Intranet

Una intranet es una red informática que utiliza la tecnología del protocolo de Internet para compartir información, sistemas operativos o servicios de computación dentro de una organización. Suele ser interna, en vez de pública como internet, por lo que solo los miembros de esa organización tienen acceso a

ella.

4.2.3.1 Etimologia

4.2.4.2 Historia

Los Sistemas Expertos (SE) pueden ser considerados como un subconjunto de la IA (Rossini, 2000). El nombre Sistema Experto deriva del término "sistema experto basado en conocimiento". Hacia los años '80, comienza a surgir la industria de los Sistemas Expertos. Se realizaron importantes inversiones en varios países de Europa, Asia y América, con el fin de lograr generar un sistema capaz de reproducir la actividad de un experto humano en tópicos específicos.

4.2.4.3 Función

Los Sistemas Expertos han sido diseñados para facilitar las tareas en múltiples campos de aplicación y proporcionar equivalentes resultados a los de un experto humano. Se sabe que no son aptos para resolver problemas generales pero su eficiencia aumenta mientras más acotado o específico sea el dominio de aplicación.

4.2.4.4 Estructura

Los SE están compuestos por dos partes principales: el ambiente de desarrollo y el ambiente de consulta. El ambiente de desarrollo es utilizado por el constructor para crear los componentes e introducir conocimiento en la base de conocimiento. El ambiente de consulta es utilizado por los no-expertos para obtener conocimiento experto y consejos.

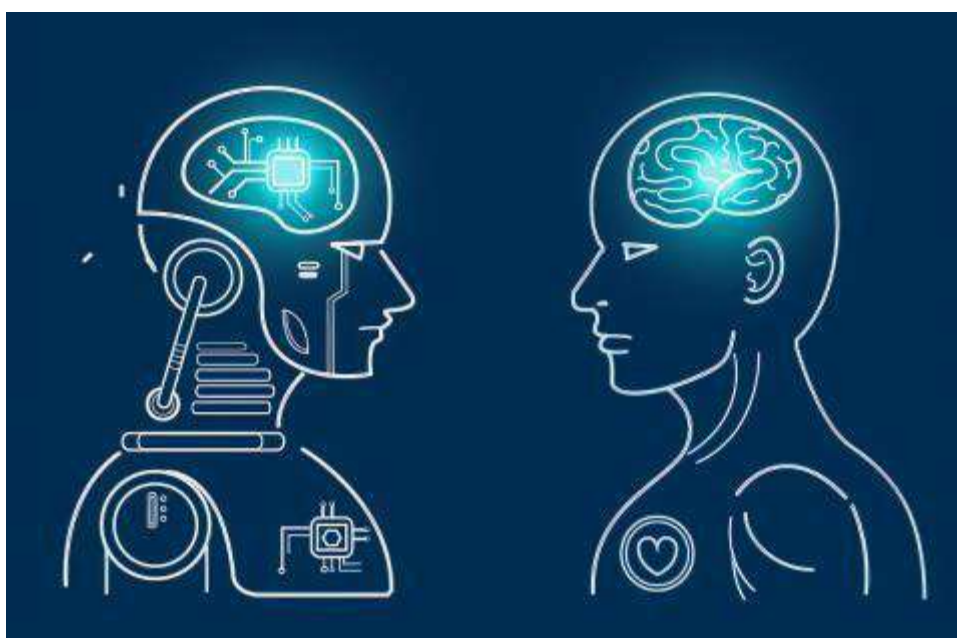


Imagen: Tecnologías de la Información Fuente:

https://www.google.com/search?q=Sistemas+Expertos&tbm=isch&ved=2ahUKewj8nJbdg4r9AhVo-cCegQIABAA&oaq=Sistemas+Expertos&gs_lcp=CgNpbWcQAzIECCMQJzIECAAQQzIECAAQQzIECAAQQzIECAAQQsQMyCwgAEIAEELEDEIMBMgQIABBDMgclABCxAXBDMgUIABCABDIFCAAQgARQowdYowdgyQoAHAAeACAAVuIAVuSAQExmAEOAEBggELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=PrzIY_yFMui2ggeO7pV_I&bih=635&biw=1366&rlz=1C1ALOY_esGT1021GT1021#imgrc=t_C8Zf1YJZvSVM