



CJAVA

The logo consists of a stylized white 'C' and 'JAVA' text. Above the 'C', there is a white graphic element resembling a 3D cube or a series of interconnected squares forming a corner shape.



**CJAVA**  
siempre para apoyarte

010101010101010101010101010101  
001010101010101010101010101010101  
0101010101010101010101010101010101  
01010101010101010101  
1010101010101010  
0101010101010101010101010101010101  
101010101010101010101001010010101010101  
01010101010101

## Visión

Poder aportar al desarrollo del País usando tecnología Java.

# Quienes Somos

Somos una organización orientada a **desarrollar, capacitar e investigar tecnología JAVA** a través de un prestigioso staff de profesionales a nivel nacional.





cjavaperu.com  
info@cjavaperu.com

COMUNIDAD

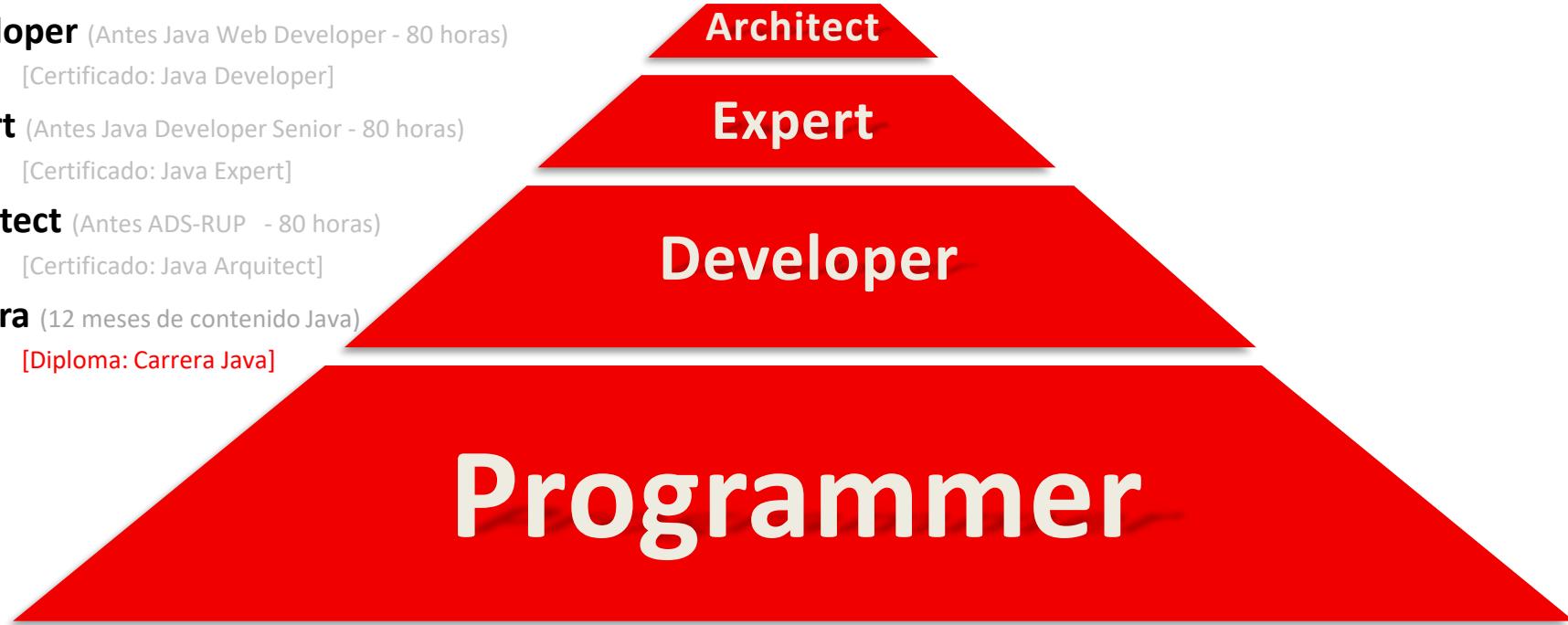
ACADEMICA

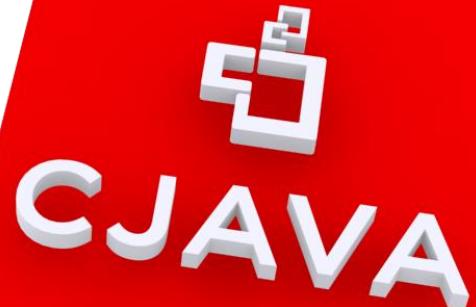


siempre para apoyarte

# Servicio de Capacitación

- **Programer** (Antes Java Developer Junior - 80 horas)  
[Certificado: Java Programer]
- **Developer** (Antes Java Web Developer - 80 horas)  
[Certificado: Java Developer]
- **Expert** (Antes Java Developer Senior - 80 horas)  
[Certificado: Java Expert]
- **Arquitect** (Antes ADS-RUP - 80 horas)  
[Certificado: Java Arquitect]
- **Carrera** (12 meses de contenido Java)  
[Diploma: Carrera Java]

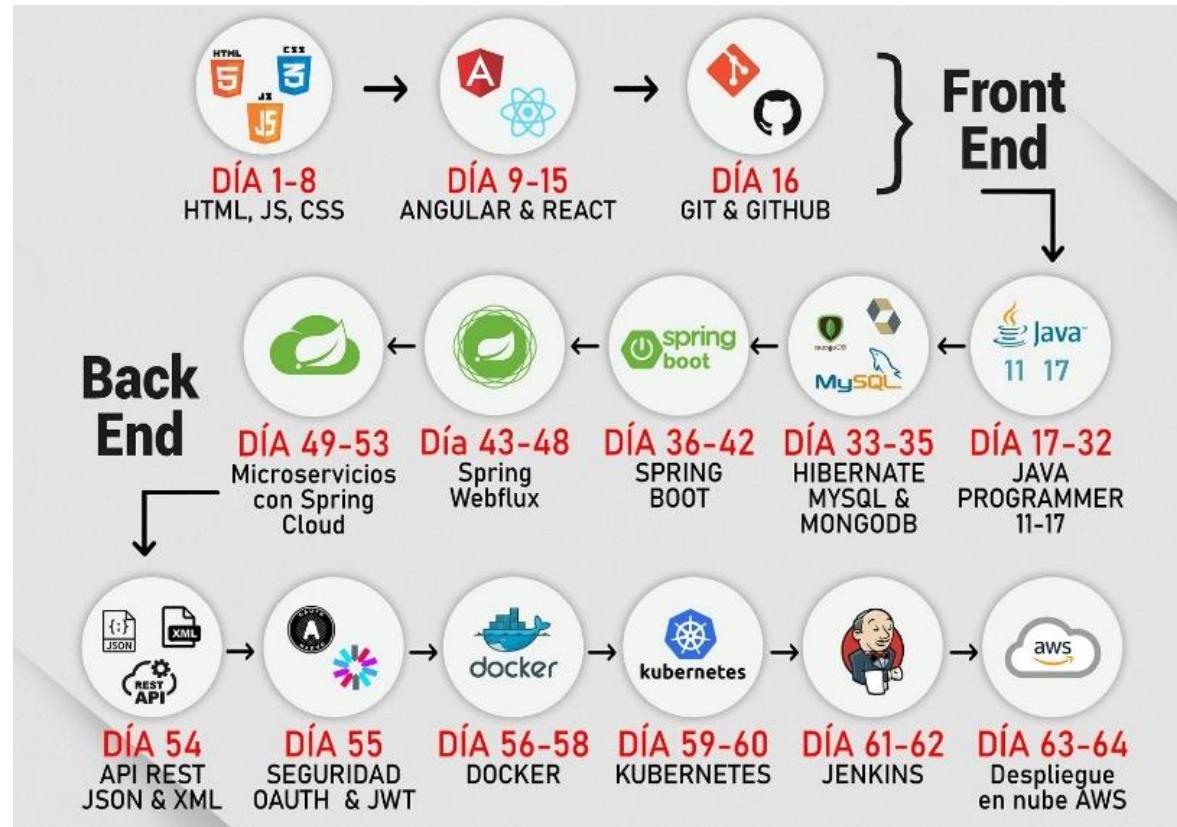


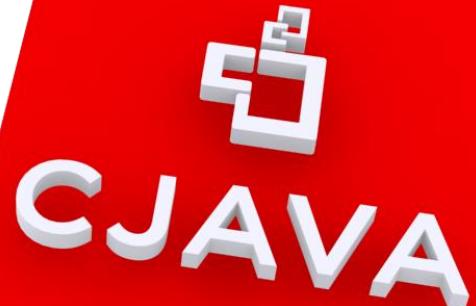


# Java FullStack Developer

emaravi@cjavaperu.com

# Introducción a Full Stack Development





# Maquetación con Bootstrap

Edwin Maravi

# Contenidos o temas

- ¿Qué es Bootstrap?
- ¿Para qué sirve Bootstrap?
- Implementación de Bootstrap

# ¿Qué es Bootstrap?

- Bootstrap es un marco CSS que fue desarrollado por Twitter en 2010 para estandarizar las herramientas de la empresa.
- En un principio se llamó Twitter Blueprint y un poco más tarde en 2011 pasó a ser de código abierto y su nombre se cambió a Bootstrap. Desde entonces, se ha actualizado varias veces y ahora se encuentra en la versión 5.2



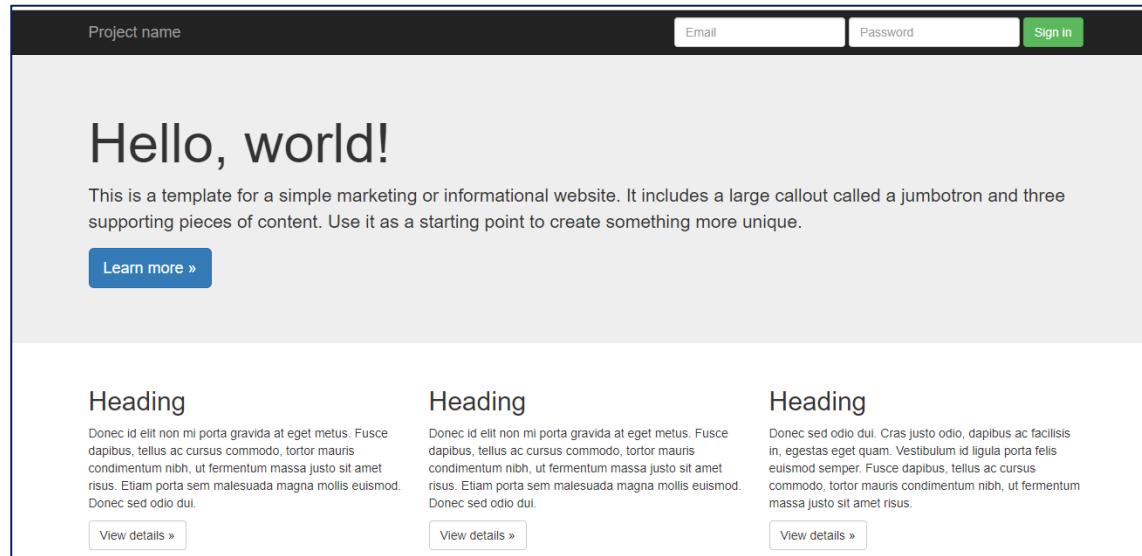
# ¿Qué es Bootstrap?

- Es una herramienta que brinda interactividad en una página, por lo que brinda elementos que facilitan la interacción del usuario, como menús de navegación, controles de página, barras de progreso y más.



# ¿Para qué sirve Bootstrap?

- Bootstrap consta de una serie de archivos CSS y JavaScript que se encargan de definir ciertos atributos de los elementos de la página.



Recuperado de <https://getbootstrap.com/>

# ¿Para qué sirve Bootstrap?

- Una de las características principales de Bootstrap es permitir que la adaptación de la página se realice según el tipo de dispositivo utilizado.
- Para garantizar la responsividad, el framework funciona con:
  - La estilización del elemento <div>
  - El uso del class container



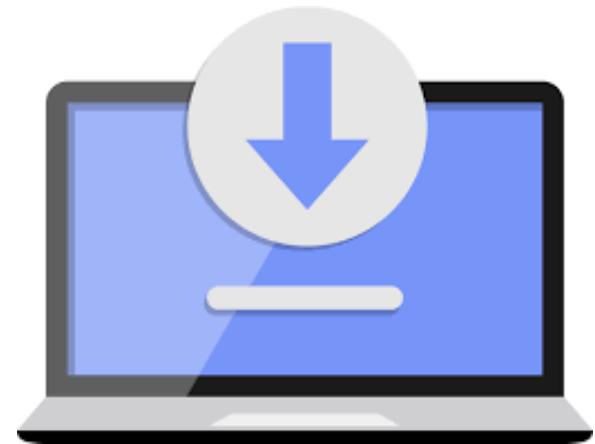
Recuperado de <https://www.ryadel.com/>

# Implementación de Bootstrap

## ¿Cómo descargar Bootstrap?

Hay diferentes formas de descargar este framework:

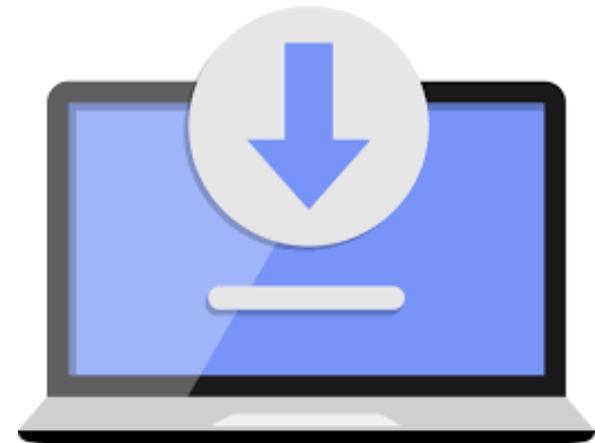
- Una de ellas es descargando la versión compilada de los códigos CSS y JavaScript en la página de Bootstrap.
- También hay una opción para descargar el código fuente del framework, puesto que es una herramienta de código abierto.
- Otra forma de descargar el framework es a través de los administradores de paquetes.



# Implementación de Bootstrap

## ¿Cómo configurar y usar Bootstrap?

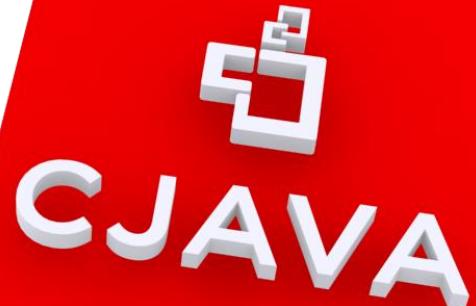
- Hay diferentes formas de configurarlo o estructurarlo para usarlo en una aplicación web. Sin embargo, para que funcione correctamente, es necesario agregar las bibliotecas JQuery y Popper.js, necesarias para la ejecución de algunos componentes de Bootstrap.
- Para comenzar a usar Bootstrap en una página, es necesario agregar las referencias de los principales archivos del framework en la página principal de la aplicación.



# Implementación de Bootstrap

- A continuación, puedes ver el código de una página HTML con todas las referencias necesarias para que la estructura funcione:

```
<!doctype html> <html lang="en"> <head>  <!-- Required meta tags --> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"> <!-- Bootstrap CSS -->  
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8XK3q堰Hxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9lfjh" crossorigin="anonymous"> <title>Hello, world!</title> </head> <body> <h1>Hello, world!</h1> <!--  
Optional JavaScript --> <!-- jQuery first, then Popper.js, then Bootstrap JS --> <script  
src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZf5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"  
crossorigin="anonymous"></script> <script  
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvblyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"  
crossorigin="anonymous"></script><script  
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4lh7YwaYd1iqfkjtj0Uod8GCExl3Og8ifwB6"  
crossorigin="anonymous"></script></body></html>  
  
<Fuente:> Getbootstrap.com
```



# Formularios en HTML y CSS

Edwin Maravi

# Resultado de aprendizaje

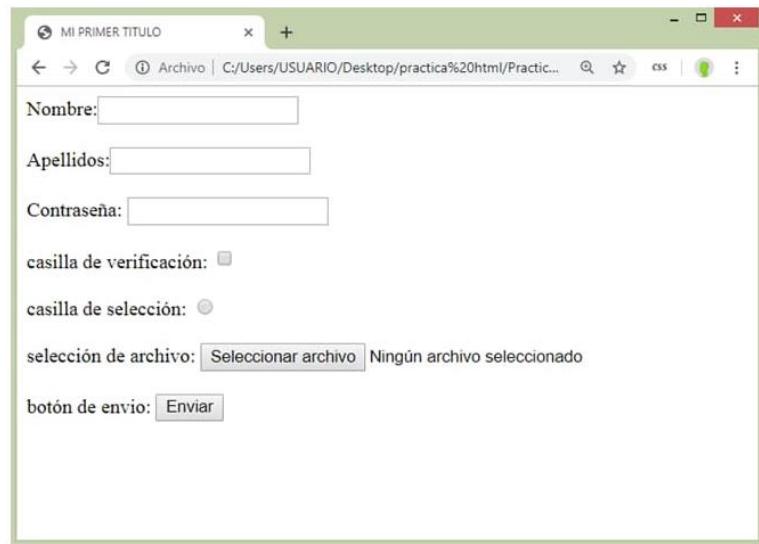
- Conoce sobre los formularios HTML. Asimismo, diseña e implementa formularios según la necesidad del usuario/cliente.

# Contenidos o temas

- ¿Qué son los formularios HTML?
- Diseñar tu formulario
- Implementación de un formulario en HTML y CSS

# ¿Qué son los formularios HTML?

- Un formulario HTML es un documento que contiene: código, elementos especiales llamados controles (casillas de verificación, radio botones y etiquetas para esos controles).
- Un usuario generalmente "llena" un formulario modificando sus controles (ingresando texto, seleccionando un elemento de un menú, etc.) y luego enviando el formulario a un servidor proxy (por ejemplo, servidor web, servidor de correo, etc.).



MI PRIMER TITULO

Nombre:

Apellidos:

Contraseña:

casilla de verificación:

casilla de selección:

selección de archivo:  Seleccionar archivo Ningún archivo seleccionado

botón de envío:

Recuperado de <https://desarrolladoresweb.org/>

# Diseñar tu formulario

CREA TU FORMULARIO DE CONTACTO EN HTML

Formulario de contacto

Escríbenos y en breve los pondremos en contacto contigo

Nombre \*

Email \*

Para generar un buen aspecto a nuestro formulario, primero debemos tener en claro qué datos queremos recopilar.



Elimina elementos adicionales (no podemos abrumar a los visitantes con demasiados problemas) y formula la planificación en papel en diferentes elementos en la página.

# Implementación de un formulario en HTML y CSS

- Para implementar un formulario, se deben considerar los siguientes elementos:

## Etiqueta <form>

Abre tu documento HTML con tu editor de texto preferido. Aquí utilizaremos el elemento `<form>` como contenedor de otro tipo elementos como campos de texto, botones de opción, casillas de verificación, botones de envío, etc.

## Elemento <input>

`<input>`es uno de los elementos más importantes y se puede mostrar de varias formas, dependiendo del atributo “type” que definamos, como “button”, “date”, “color”, “date”, “password”, etc.

# Implementación de un formulario en HTML y CSS

- Para implementar un formulario, se deben considerar los siguientes elementos:

## Elemento <label>

El elemento <label> define una etiqueta para varios elementos de formulario.

## Elemento <select>

El elemento <select> dentro de un <form> permite definir una lista desplegable. ¿Y cómo puedo definir opciones? A través del elemento <option> que define la opción que se puede seleccionar.



# Implementación de un formulario en HTML y CSS

## Ejemplo del elemento select

```
<h3>Elemento select</h3>

<p>El elemento select se utiliza para definir una lista desplegable</p>

<form action="/action_page.php">
    <label for="editors">Elige un editor de texto:</label>
    <select name="editor" id="editor">
        <option value="vsc">visual Studio Code</option>
        <option value="atom">Atom</option>
        <option value="sublime">Sublime Texto</option>
        <option value="vim">Vim</option>
    </select>
    <br><br>
    <input type="submit" value="Submit">
</form>

</body>
</html>
```

# Implementación de un formulario en HTML y CSS

- Para implementar un formulario, se deben considerar los siguientes elementos:

## Elemento <textarea>

El elemento `<textarea>` permite definir un campo de texto de varias líneas. Aquí puedes utilizar los atributos “cols” y “rows” para definir el ancho del área y la cantidad de líneas de texto visibles.

## Elemento <button>

`<button>` nos permite definir un botón cliqueable dentro de nuestro formulario.

# Implementación de un formulario en HTML y CSS

## Ejemplo del elemento button

```
<!DOCTYPE html>
<html>
<body>

<h3>Elemento button</h3>

<button type="button" onclick="alert('Go Henrys!')">Aplica a Henry</button>

</body>
</html>
```

# Implementación de un formulario en HTML y CSS

Para implementar un formulario, se deben considerar los siguientes elementos:

## Elemento <datalist>

Si queremos que los usuarios vean una lista desplegable con opciones predefinidas a medida que ingresan sus datos, lo recomendado es utilizar el elemento `<datalist>` el cual nos permite especificar esa lista de opciones predeterminadas para un elemento `<input>`.

## Elemento <output>

`<output>` es un elemento contenedor en el que un sitio o aplicación puede injectar los resultados de un cálculo o el resultado de una acción del usuario.



# Creación de un portal web ecommerce

Edwin Maravi

# Contenidos o temas

- ¿Qué es un portal de ecommerce?
- ¿Por qué es importante una plataforma ecommerce?
- ¿Qué se debe tener en cuenta antes de crear un ecommerce?
- Creación del portal

# ¿Qué es un portal de ecommerce?

- Es un método de compra y venta que utiliza Internet como medio para realizar transacciones y comunicarse con los consumidores.
- No solo a través de sitios web, sino también a través de las redes sociales.



# ¿Por qué es importante una plataforma ecommerce?

- Una plataforma ecommerce ofrece las siguientes ventajas:

- Globalización y mayor público
- Conocimiento del cliente
- Mayor variedad de productos
- Fidelización
- Disminución de costos



# ¿Qué se debe tener en cuenta antes de crear un ecommerce?

- Antes de su creación se debe tener en cuenta:

•Objetivos claros y plan de acción

•Producto

•Público objetivo

•Tecnología

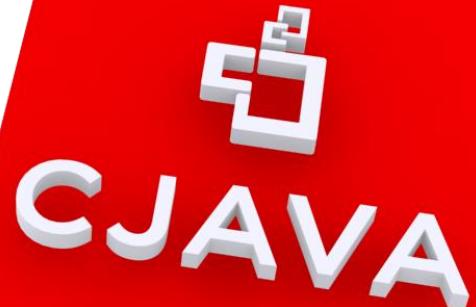
•Logística

•Estrategia de marketing

# Creación del portal

1. Elegir el nombre, dominio y logotipo
2. Establecer un Sistema de Gestión de Contenidos (CMS)
3. Adquirir una opción de hosting
4. Personalizar y organizar tus productos en la tienda virtual
5. Definir los medios de pago
6. Realizar una estrategia de Marketing de contenidos
7. Proporcionar información relevante a los clientes





# Creación de las secciones del portal web

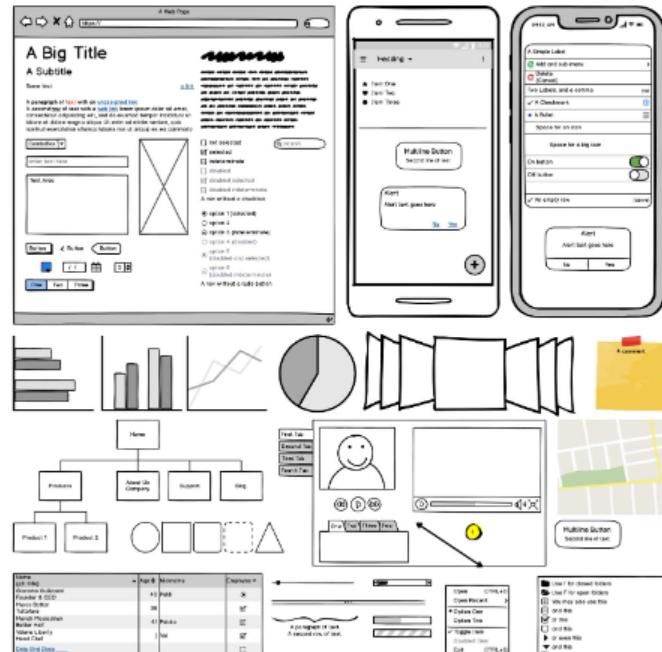
Edwin Maravi

# Contenidos o temas

- Wireframe del portal web
- Creación de secciones
- Uso de HTML, CSS y BOOTSTRAP para la implementación

# Wireframe del portal web

- Un wireframe es una especie de boceto, que sirve para mostrarle al cliente cómo quedaría el producto final (página web) a nivel de su estructura, es decir, cómo se organizarían los elementos en la interfaz.
- De esta forma, se le podría mostrar la propuesta al cliente en una primera fase para obtener su aprobación y garantizar que se satisfagan sus necesidades y expectativas respecto al producto final.



# Wireframe del portal web

## Los pasos para crear un wireframe son los siguientes:

1. Haz una lista numerada con toda la información que quieras incluir en la página.

2. Luego deberías agrupar todos los elementos que tengan relación entre sí, y asignarles algún número para relacionarlas.

3. Decide qué grupo contiene información importante, necesaria, y complementaria.

4. Diseña cada grupo: su ubicación, tanto dentro de la página como dentro del mismo grupo.

5. Junta los grupos en función al diseño y al equilibrio de la página en global.

6. Implementa todo esto por medio de una herramienta. De esto se infiere que los puntos anteriores deberías hacerlos en papel para, a continuación, pasarlos a ordenador.

# Creación de secciones

- Si bien es cierto, existen distintos tipos de páginas web con distintos fines y características, un factor común es que, si hablamos de ecommerce, estas deben poseer las secciones primordiales para cubrir las necesidades de los usuarios:



# Uso de HTML, CSS y BOOTSTRAP para la implementación

- BootStrap es un framework que sirve para crear páginas web con diseño responsive, es decir, un diseño que se adapta a todo tipo de dispositivos.



# Uso de HTML, CSS y BOOTSTRAP para la implementación

Entre sus características, podemos mencionar:

- Facilita a los desarrolladores la maquetación de las páginas.
- Incluye un conjunto de clases CSS para maquetar de manera fácil una página a cualquier resolución.
- Es mantenido y actualizado por Twitter.
- Está optimizado para dispositivos móviles.
- Permite un diseño “Responsive”
- Requiere conocimientos amplios de CSS.



# Crear documentos de BOOTSTRAP

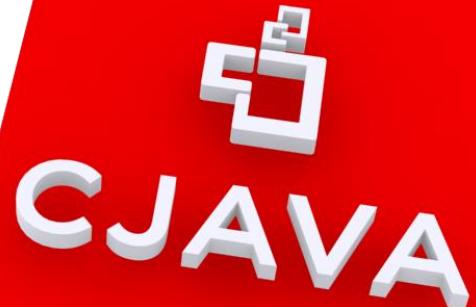
Puedes empezar a diseñar su sitio web con Bootstrap mediante una de las siguientes opciones del cuadro de diálogo Nuevo documento (Adobe, 2021):

## Plantillas de inicio de Bootstrap (Plantillas de inicio > Plantillas de Bootstrap)

Utilice esta opción si quiere empezar rápidamente sin tener que molestarte en diseñar las páginas a partir de cero. Solo tiene que editar el texto y reemplazar los activos, si lo necesita, y habrá creado un sitio web interactivo.

## Crear un documento HTML basado en el framework de Bootstrap (Nuevo documento > HTML > Bootstrap):

Use esta opción si quiere crear su sitio web paso a paso, utilizando componentes de Bootstrap y CSS en Dreamweaver.



# Introducción al JavaScript

Edwin Maravi

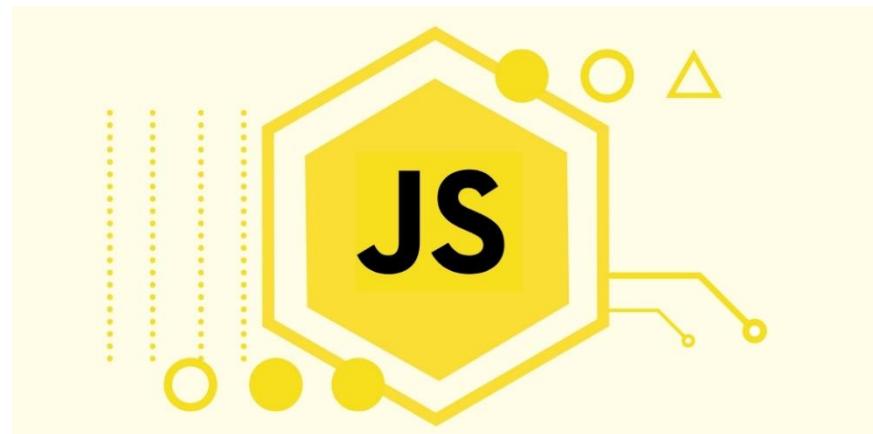


# Contenidos o temas

- Empezando con JavaScript
- Variables en JavaScript

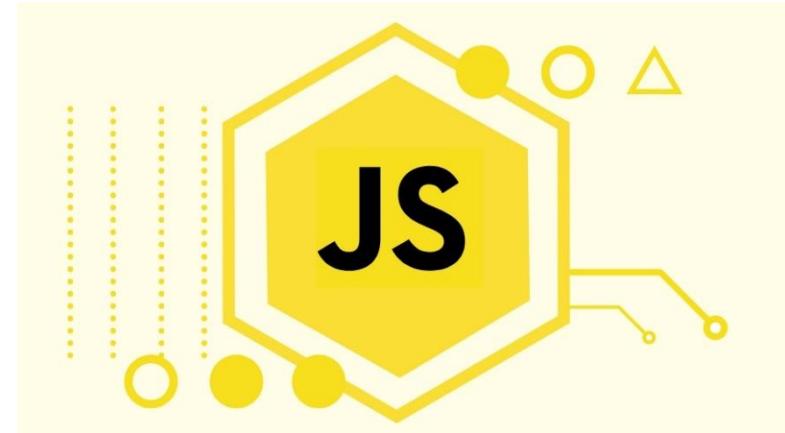
# Empezando con JavaScript

- JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web

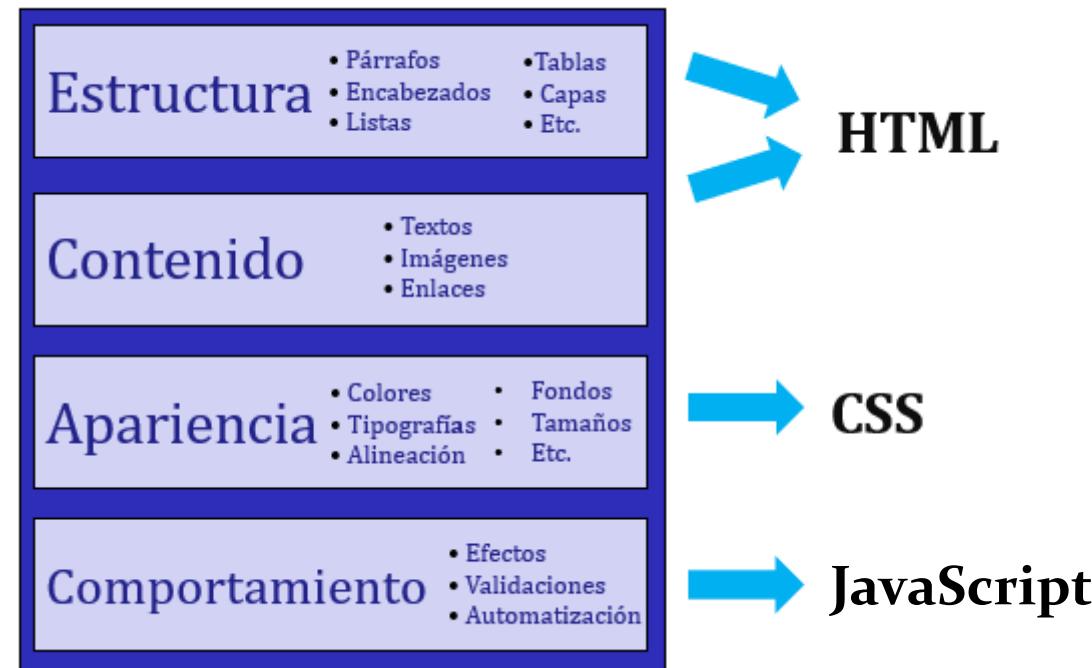


# Empezando con JavaScript

- Cada vez que una página web hace algo más que 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.



# Empezando con JavaScript



# Empezando con JavaScript

## ¿Qué puede hacer JavaScript en el navegador?

✓ Agregar nuevo HTML a la página, cambiar el contenido existente y modificar estilos.

Recordar datos en el lado del cliente con el almacenamiento local (“local storage”).

✓ Enviar solicitudes de red a servidores remotos, descargar y cargar archivos (Tecnologías llamadas AJAX y COMET).

✓ Obtener y configurar cookies, hacer preguntas al visitante y mostrar mensajes.

# Variables

- Una variable es un espacio en memoria donde se almacena un dato, un espacio donde podemos guardar cualquier tipo de información que necesitemos para realizar las acciones de nuestros programas.





# Variables

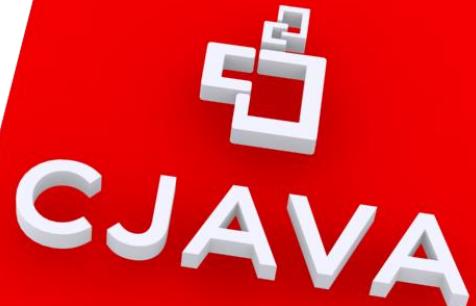
- Aunque todas las variables de JavaScript se crean de la misma forma (mediante la palabra reservada var), la forma en la que se les asigna un valor depende del tipo de valor que se quiere almacenar (números, textos, etc.). Veamos:

•Numéri  
cas

•Cadena  
s de  
texto

•Arrays

•Boolea  
nos



# Funciones básicas de JavaScript

Edwin Maraví



# Contenidos o temas

- Números aritméticos
- Operaciones aritméticas
- Cadenas

# Números aritméticos

## Tipos de números

Usamos diferentes términos para describir diferentes tipos de números decimales, por ejemplo:

- ✓ **Enteros:** son números sin parte decimal, e.g. 10, 400 o -5.
- ✓ **Números con coma flotante (floats):** tienen punto decimal y parte decimal, por ejemplo, 12.5 y 56.7786543.
- ✓ **Doubles:** son un tipo específico de números de coma flotante que tienen una mayor precisión que los números de coma flotante comunes (lo que significa que son más precisos en cuanto a la cantidad de decimales que poseen).

# Operaciones aritméticas

Para realizar operaciones aritméticas, se utilizan operadores básicos:

Operador	Nombre	Propósito	Ejemplo
+	Adición	Suma dos números juntos.	$6 + 9$
-	Resta	Resta el numero de la derecha del de la izquierda.	$20 - 15$
*	Multiplicación	Multiplica dos números juntos.	$3 * 7$
/	División	Divide el número de la izquierda por el de la derecha.	$10 / 5$
%	Sobrante (también llamado módulo)	Retorna el restante después de dividir el número de la izquierda en porciones enteras del de la derecha.	$8 \% 3$ (retorna 2, como tres está dos veces en 8, quedando 2 restantes.)

<https://developer.mozilla.org>

# Operaciones aritméticas

## Operadores de incremento y decremento

Algunas veces necesitarás repetidamente sumar o restar uno de/a una variable numérica. Esto puede hacerse convenientemente usando los operadores de incremento (++) y decremento (--). Prueba esto:

```
var num1 = 4;  
num1++;
```

```
var num2 = 6;  
num2--;  
num2;
```

# Operaciones aritméticas

## Operadores de asignación

- Son operadores que asignan un valor a una variable. Ya usamos el más básico, `=`, muchas veces — simplemente asigna a la variable de la izquierda, el valor de la derecha:

Operador	Nombre	Propósito	_Ejemplo	_Atajo_de_
<code>+=</code>	Adición asignación	Suma el valor de la derecha al valor de la variable de la izquierda y retorna el nuevo valor	<code>x = 3; x += 4;</code>	<code>x = 3; x = x + 4;</code>
<code>-=</code>	Resta asignación	Resta el valor de la derecha, del valor de la variable de la izquierda y retorna el nuevo valor.	<code>x = 6; x -= 3;</code>	<code>x = 6; x = x - 3;</code>
<code>*=</code>	Multiplicación asignación	Multiplica el valor de la variable en la izquierda por el valor en la derecha y retorna el nuevo valor.	<code>x = 2; x *= 3;</code>	<code>x = 2; x = x * 3;</code>
<code>/=</code>	División asignación	Divide el valor de la variable en la izquierda por el valor de la derecha y retorna el nuevo valor.	<code>x = 10; x /= 5;</code>	<code>x = 10; x = x / 5;</code>

<https://developer.mozilla.org>

# Cadenas

Las cadenas son útiles para almacenar datos que se pueden representar en forma de texto. Algunas de las operaciones más utilizadas en cadenas son verificar su length, para construirlas y concatenarlas usando operadores de cadena + y +=, verificando la existencia o ubicación de subcadenas con indexOf() o extraer subcadenas con el método substring().

```
const string1 = "Una cadena primitiva";
const string2 = 'También una cadena primitiva';
const string3 = `Otra cadena primitiva más`;

const string4 = new String("Un objeto String");
```

# Cadenas

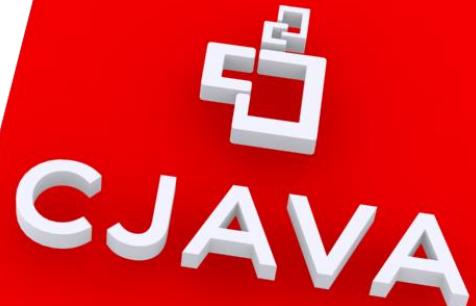
**Hay dos formas de acceder a un carácter individual en una cadena.**

- La primera es con el método `charAt()`:

```
return 'cat'.charAt(1) // devuelve "a"
```

- La otra forma (introducida en ECMAScript 5) es tratar a la cadena como un objeto similar a un arreglo, donde los caracteres individuales corresponden a un índice numérico:

```
return 'cat'[1] // devuelve "a"
```



# Condicionales y bucles con JavaScript

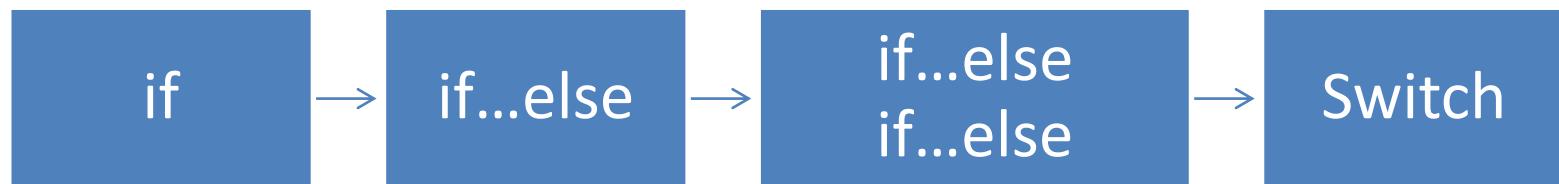
Edwin Maraví

# Contenidos o temas

- Condicionales
- Ciclo While
- Ciclo For

# Condicionales

Las «instrucciones condicionales» se usan para realizar las diferentes acciones según una condición. En JavaScript podemos encontrar las siguientes declaraciones condicionales:



# Condicionales

Instrucción	Descripción	Sintaxis
if	Se usa para ejecutar un bloque de código si la condición es verdadera.	<pre>if (condición) {     bloque de código a ejecutar     si la condición es verdadera }</pre>
if...else	Se usa para ejecutar un bloque de código si la condición es verdadera u otro bloque de código si la condición es falsa.	<pre>if(condición){     bloque de código a ejecutar     si la condición es verdadera } else {     bloque de código a ejecutar     si la condición es falsa }</pre>

# Condicionales

Instrucción	Descripción	Sintaxis
if...else if...else	Se usa para seleccionar uno de los muchos bloques de código a ejecutar.	<pre>if(condición1){     bloque de código a ejecutar     si la condición1 es verdadera } else if(condición2){     bloque de código a ejecutar     si la condición2 es verdadera } else {     bloque de código a ejecutar     si condición1 y condición2     no son verdaderas. }</pre>

# Condicionales

Instrucción	Descripción	Sintaxis
Switch	Se usa para seleccionar uno de los varios bloques de código a ejecutar (igual).	<pre>switch ([evaluación   valor]) {     case 1:         ejecuta este bloque         si el valor o la evaluación         son iguales a 1.         break;     case 2:         ejecuta este bloque         si el valor o la evaluación         son iguales a 2.         break;     default:         ejecuta este bloque si no         ingresa a ninguno         de los "case" anteriores. }</pre>

# Bucles

Los bucles ofrecen una forma rápida y sencilla de hacer algo repetidamente; es decir, repiten una acción varias veces. En esta sesión, revisaremos las declaraciones For y While.

## Ciclo FOR

Un ciclo for se repite hasta que una condición especificada se evalúe como false. El bucle for de JavaScript es similar al bucle for de Java y C.

La declaración for tiene el siguiente aspecto:

```
for ([expresiónInicial]; [expresiónCondicional]; [expresiónDeActualización])  
    instrucción
```

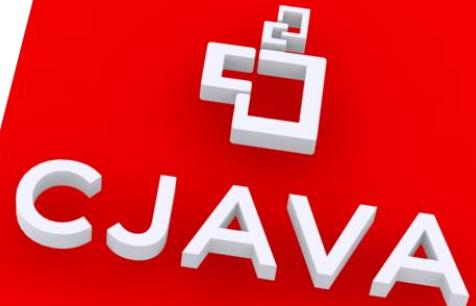


# Bucles

## Ciclo While

Una declaración while ejecuta sus instrucciones siempre que una condición especificada se evalúe como true. Una instrucción while tiene el siguiente aspecto:

```
while (condición)  
    expresión
```



# Métodos con JavaScript

Edwin Maravi



# Contenidos o temas

- ¿Qué son los métodos con JavaScript?
- Métodos para listas, strings y números
- Diferencias entre métodos y funciones en JavaScript

# ¿Qué son los métodos con JavaScript?

- Todos los objetos predefinidos por javascript que conocemos tienen sus propios métodos.
- Los métodos sirven para hacer acciones como una ventana emergente, modificar las propiedades de un objeto, añadir elementos a una lista, quitarlos, transformar un string en un número entero o decimal



# Métodos para listas, strings y números

## Métodos numéricos

Podemos anotar como importantes:

- parseInt() transforma un string en número entero. En caso de ser un texto como, por ejemplo, 5Kg nos devolvería el número 5.
- parseFloat() hace lo mismo que parseInt pero en este caso con un número decimal.

```
//métodos para números  
  
// parseFloat() *■  
  
// parseInt() *■  
  
// toFixed() ■  
  
// toPrecision() ■  
  
// toString() ■
```

# Métodos para listas, strings y números

## Métodos para strings:

Métodos	Descripción
indexOf()	Devuelve el primer elemento de un string que sea igual al valor indicado dentro del paréntesis. En caso de no encontrarlo, devuelve -1.
lastIndexOf()	Hace lo mismo que indexOf() pero empezando por el final de un string
split()	Crea una lista de elementos a partir de un string separando los elementos por el carácter indicado dentro del paréntesis.
Slice(x, y)	Devuelve los caracteres a partir de la posición x. Tener en cuenta que se empieza a contar que el primer carácter siempre ocupa la posición 0.

# Métodos para listas, strings y números

## Métodos para arrays

Métodos	Descripción
length	Es el más importante de todos. Devuelve el número de elementos de una lista.
join()	Junta todos los elementos de una lista separados por el carácter que indiquemos dentro del paréntesis (entre comillas porque será un string)
pop()	Elimina el último elemento de una lista y hace que la lista quede disminuida en una posición su longitud.
sort()	Ordena alfabéticamente o numéricamente los elementos de una lista.

# Diferencias entre métodos y funciones en JavaScript

## Funciones de JavaScript

Una función es un bloque de código escrito para realizar un conjunto específico de tareas. Podemos definir una función usando la palabra clave de función, seguida de Nombre y parámetros opcionales. El cuerpo de la función está encerrado entre llaves.

### Sintaxis:

```
function functionName(parameters) {  
    // Content  
}
```

# Diferencias entre métodos y funciones en JavaScript

## Métodos JavaScript

Un método JavaScript es una propiedad de un objeto que contiene una definición de función. Los métodos son funciones almacenadas como propiedades de objeto.

Se puede acceder al método de objeto con la siguiente sintaxis:

```
object = {  
    methodName: function() {  
        // Content  
    }  
};  
  
object.methodName()
```



# CJAVA

siempre para apoyarte

Gracias