

*Universidad de San Carlos de Guatemala USAC.
División de Ciencias de la Ingeniería.
Centro Universitario de Occidente CUNOC.
Teoría de Sistemas 1
Ing. Pedro*



Estudiante

*Elvis Lizandro Aguilar Tax
Rudy Adolfo Pacheco Pacheco
Brayan Alexander Alonzo Quijivix*

Carnet

*201930304
201930220
201931068*

***Documentación de Tecnologías usadas en el proyecto
"Gamification"***

¿Qué es una página web?

Es un documento digital complejo, que puede integrar y/o contener texto, sonido, video, programas, enlaces, imágenes, hipervínculos y otros elementos, adaptado para la World Wide Web (WWW), y que puede ser accedida y visualizada mediante un navegador web. Esta información se encuentra generalmente en etiquetado HTML o XHTML, y puede proporcionar acceso a otras páginas web mediante enlaces de hipertexto. Frecuentemente también incluyen otros recursos como pueden ser hojas de estilo en cascada, scripts, imágenes digitales, etc.

Las páginas web pueden estar almacenadas en un computador o en un servidor web remoto. El servidor web puede restringir el acceso únicamente a redes privadas, por ejemplo, en una intranet corporativa, o puede publicar las páginas en la World Wide Web. Es decir que se utiliza para acceder a páginas de distintas redes sociales para poder acceder a ella necesitas una cuenta.

¿Qué es una aplicación web?

Una aplicación web es una aplicación de software que se ejecuta en un navegador web y utiliza tecnologías web como HTML, CSS y JavaScript para ofrecer funcionalidades y servicios a los usuarios a través de Internet. A diferencia de las aplicaciones de escritorio que se instalan en el equipo del usuario, las aplicaciones web son accesibles desde cualquier dispositivo con conexión a Internet y un navegador web compatible.

Las aplicaciones web se ejecutan en servidores remotos y los usuarios interactúan con ellas a través de la interfaz proporcionada por el navegador web. Esto significa que los usuarios no necesitan instalar nada en su dispositivo, ya que todo el procesamiento y la lógica de la aplicación ocurre en el servidor.

Las aplicaciones web pueden variar en complejidad y funcionalidad. Pueden ser simples, como un sitio web estático que muestra información básica, o pueden ser altamente interactivas y dinámicas, con capacidades de procesamiento de datos, formularios, autenticación de usuarios, integración con bases de datos y más.

Algunos ejemplos comunes de aplicaciones web son las redes sociales, los servicios de correo electrónico en línea, los sistemas de gestión de contenidos (CMS), las tiendas en línea, las aplicaciones de banca en línea y las herramientas de productividad colaborativa.

Tecnologías de frontend

¿Qué es Html?

HTML es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, videos, etc.

El HTML se creó en un principio con objetivos divulgativos de información con texto y algunas imágenes. No se pensó que llegara a ser utilizado para crear área de ocio y consulta

con carácter multimedia (lo que es actualmente la web), de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, si que se han ido incorporando modificaciones con el tiempo, estos son los estándares del HTML.

Numerosos estándares se han presentado ya. El HTML 4.01 es el último estándar a febrero de 2001. Actualización a mayo de 2005, en estos momentos está apunto de presentarse la versión 5 de HTML, de la que ya se tiene un borrador casi definitivo

HTML (Lenguaje de marcado de hipertexto o HyperText Markup Language por sus siglas en inglés)

¿Cuáles son las etiquetas HTML básicas?

<body> para el contenido.

<head> para información sobre el documento.

<div> división dentro del contenido.

<a> para enlaces.

 para poner el texto en negrita.

 para saltos de línea.

<H1>...<H6> para títulos dentro del contenido.

 para añadir imágenes al documento.



SINTAXIS DE HTML

DE APERTURA Y CIERRE



ATRIBUTOS

Los elementos pueden tener **atributos que agregan información adicional**.



HTML es el primer lenguaje que debes aprender para iniciar en el desarrollo web. **SÍ ES UN LENGUAJE (por eso la L en HTML).**

Prof. Álvaro Felipe

¿Qué otros lenguajes de la web ya dominas?

ed.team/cursos/html



¿Qué es CSS?

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilo utilizado para definir la presentación de un documento HTML o XML. CSS se utiliza para describir cómo se debe mostrar la información en la pantalla, en papel o en otros medios. Con CSS, los diseñadores y desarrolladores web pueden controlar la apariencia de un sitio web, incluyendo aspectos como el color, la fuente, el espaciado, la alineación, el diseño y la animación de los elementos de una página web.

En lugar de definir el estilo de cada elemento de una página web en línea dentro del código HTML, CSS permite que los estilos se definen en un archivo separado y luego se apliquen a los elementos HTML según sea necesario. Esto permite una mayor flexibilidad en el diseño y facilita la gestión de los estilos y el mantenimiento del sitio web a largo plazo.

¿CONOCES LA SINTAXIS DE CSS?

```
1 .element {  
2   --color : #FFF;  
3   background : var(--color);  
4   width : 100%;  
5 }  
3
```

1 Regla CSS

2 Selector

3 Propiedades

4 Valores

5 Declaración

Aprende CSS y diseña para la web en:

👉 ed.team/cursos/css



Ejemplo

ejemplo simple de CSS que puede aplicarse a un elemento HTML h1:

HTML

```
<h1>Ejemplo de CSS</h1>
```

CSS

```
h1 {  
  color: blue;  
  font-size: 36px;  
  font-weight: bold;  
  text-align: center;  
}
```

ESPECIALÍZATE EN DISEÑO WEB CON CSS



El diseño web es una de las áreas con mayor crecimiento en los últimos años.
Conviértete en un excelente diseñador web aprendiendo a fondo CSS.

1 CSS DESDE CERO

Haz hermosa la web con el único lenguaje creado para diseño



2 FLEXBOX Y GRID

Diseña interfaces flexibles con el sistema que utilizan todos los frameworks CSS.

3 CSS AVANZADO

Aprende a profundidad los diversos módulos que tiene CSS.



5 RESPONSIVE WEB DESIGN

Diseña sitios web y aplicaciones que se vean perfectas en cualquier tamaño de pantalla.



7 SASS DESDE CERO

Convierte tu estilos CSS en un lenguaje de programación con Sass.



4 VARIABLES CSS

¡CSS tiene variables nativas con soporte total de los navegadores! Te enseñamos a usarlas en tus proyectos.

6 ANIMACIONES CON CSS

Crea animaciones optimizadas para la web en 2D y 3D sin programar.



¿Se pueden crear ciclos y condicionales en CSS? ¿V o F?

ed.team/especialidades/css



En este ejemplo, el CSS se aplica al elemento `h1` y establece las siguientes propiedades:

- `color: blue;` - establece el color del texto en azul.
- `font-size: 36px;` - establece el tamaño de fuente en 36 píxeles.
- `font-weight: bold;` - establece el grosor de la fuente en negrita.
- `text-align: center;` - establece la alineación del texto en el centro del elemento.

Al aplicar este CSS, el texto dentro del elemento `h1` se mostrará en azul, en negrita, con un tamaño de fuente de 36 píxeles y centrado dentro del elemento

¿Qué es Javascript?

JavaScript es un lenguaje de programación que se utiliza principalmente para crear contenido dinámico e interactivo en páginas web. Es un lenguaje de programación de alto nivel, interpretado, orientado a objetos y multiplataforma.

JavaScript se utiliza para crear efectos visuales enriquecedores, como animaciones, así como para crear interacciones complejas entre el usuario y la página web. También se utiliza para interactuar con servicios web y bases de datos, y para desarrollar aplicaciones web en el lado del cliente.

A diferencia de HTML y CSS, que se utilizan para crear la estructura y el diseño de una página web, respectivamente, JavaScript se utiliza para agregar interactividad y funcionalidad a la página web.

JavaScript se ejecuta en el navegador web del usuario y puede interactuar con la página web, el documento y los elementos HTML. También se puede utilizar en el lado del servidor a través de Node.js para crear aplicaciones web escalables y de alto rendimiento.

Ejemplo:

ejemplo simple de JavaScript que muestra una alerta en el navegador cuando se hace clic en un botón:

HTML:

```
<button onclick="mostrarAlerta()">Haz clic aquí</button>
```

JavaScript:

```
function mostrarAlerta() {  
    alert('¡Hola mundo!');  
}
```

En este ejemplo, se define una función `mostrarAlerta()` que muestra una alerta con el mensaje "¡Hola mundo!" cuando se llama. Luego, se agrega un controlador de eventos en el botón que llama a la función `mostrarAlerta()` cuando se hace clic en el botón.

Al hacer clic en el botón, se llama a la función `mostrarAlerta()` y se muestra una alerta en el navegador con el mensaje "¡Hola mundo!". Este es un ejemplo simple de cómo se puede utilizar JavaScript para crear interacciones dinámicas con el usuario en una página web

¿CONOCES LA SINTAXIS DE JAVASCRIPT?

```
1 let language = 'JavaScript'  
2 let company = {  
3   name: 'EDteam',  
4   slogan: 'Nunca te detengas',  
5   founded: 2015  
6 }  
7 console.log(company.name)  
8 // 'EDteam'  
9 const getMajorNumber = (a,b) => {  
10   if (a > b) { return a }  
11   else { return b }  
12 }  
13 getMajorNumber(4,6)  
14 // 6
```



→ Las variables se declaran con let (no hay que indicar el tipo de dato)

→ Los objetos encierran entre llaves parejas con el formato propiedad: valor separadas por comas.

→ console.log() imprime en consola la expresión entre los paréntesis.

→ Para obtener el valor de una propiedad de un objeto se usa objeto.propiedad

→ Condicional (if / else)

Comentarios (Líneas 8 y 14)

→ Ejecución de la función

→ Definición de función (se recomienda usar constantes con const)

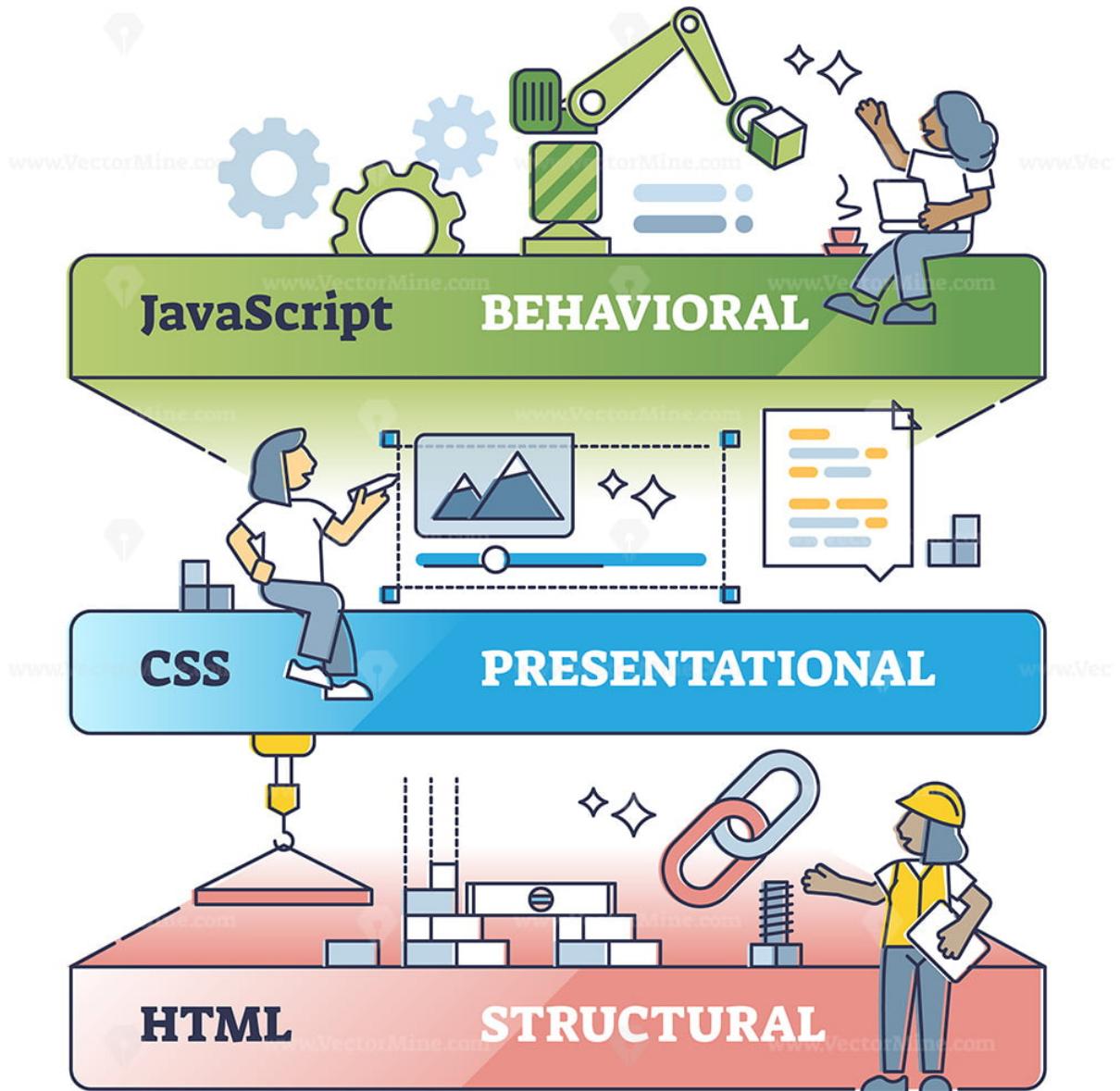


A diferencia de Python, los saltos de línea e indentación no son parte de la sintaxis.

Domina JavaScript desde cero a avanzado (primer curso gratis)

 ed.team/javascript





Tecnologías Backend

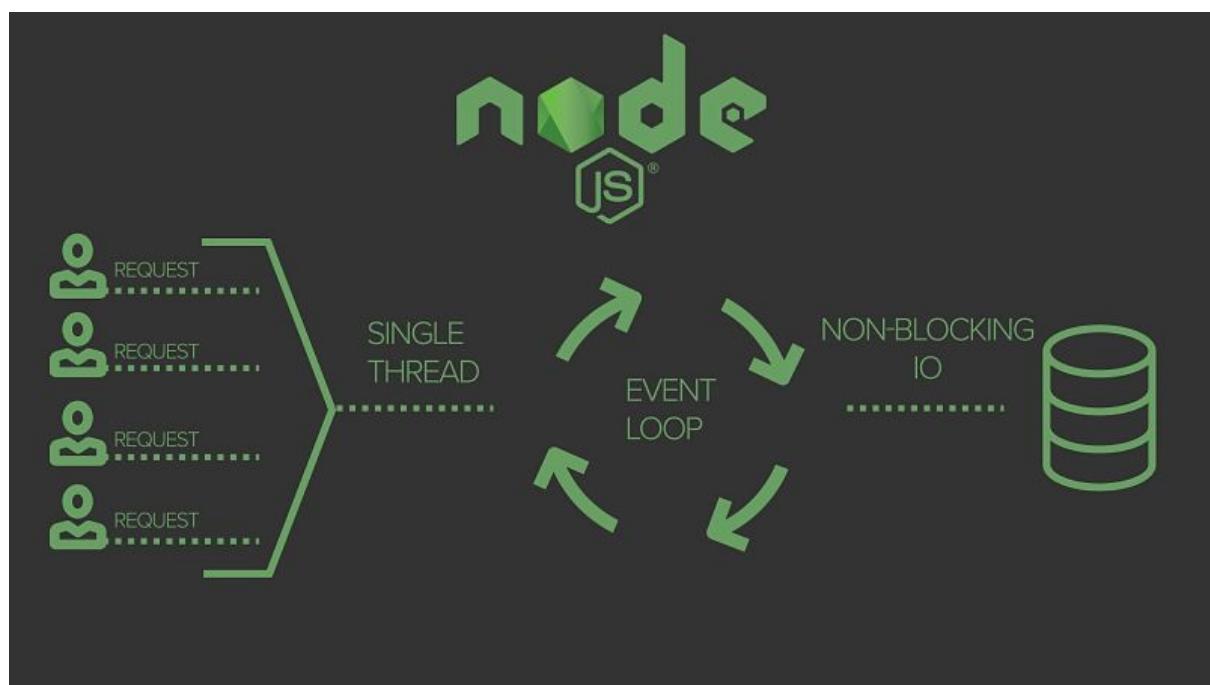
¿Qué es Nodejs?

Node.js es un entorno de ejecución de código abierto basado en el motor de JavaScript V8 de Google. Permite a los desarrolladores ejecutar código JavaScript tanto en el lado del servidor como en el cliente. A diferencia de JavaScript tradicional, que se ejecuta en el navegador, Node.js se ejecuta en el servidor y proporciona un entorno para construir aplicaciones web y de red altamente escalables y de alto rendimiento.

Una de las principales características de Node.js es su enfoque en el modelo de E/S sin bloqueo y basado en eventos. Esto significa que Node.js es capaz de manejar muchas conexiones simultáneas de manera eficiente, sin bloquear la ejecución de otros procesos. Esta capacidad lo hace especialmente adecuado para aplicaciones en tiempo real y de alto tráfico, como aplicaciones de chat, juegos en línea, aplicaciones de streaming, entre otros.

Node.js también cuenta con un gestor de paquetes llamado npm (Node Package Manager), que es uno de los repositorios de paquetes más grandes y activos disponibles para cualquier lenguaje de programación. npm facilita la instalación y gestión de dependencias de proyectos de Node.js, lo que contribuye a la eficiencia y la reutilización de código.

Node.js es utilizado por muchos desarrolladores y empresas para construir aplicaciones web y de red escalables y eficientes. Es especialmente popular en el desarrollo de servidores web, API (Application Programming Interface), aplicaciones de tiempo real y microservicios.



¿Qué es express js?

Express.js es un framework de aplicaciones web para Node.js. Proporciona una capa de abstracción sobre las funcionalidades de bajo nivel de Node.js, lo que simplifica el desarrollo de aplicaciones web y API de manera rápida y eficiente.

Express.js es conocido por su enfoque minimalista y su flexibilidad. No impone una estructura de proyecto rígida y permite a los desarrolladores elegir las herramientas y bibliotecas adicionales que deseen utilizar en su aplicación. Esto permite una mayor libertad y adaptabilidad en el desarrollo.

Con Express.js, los desarrolladores pueden crear fácilmente rutas para manejar solicitudes HTTP, definir middleware para procesar y manipular las solicitudes y respuestas, gestionar sesiones y cookies, y manejar errores de manera centralizada. También ofrece soporte para la creación de APIs RESTful y la gestión de archivos estáticos.

Express.js tiene una gran comunidad de desarrolladores y una amplia gama de complementos y middleware disponibles a través del ecosistema de npm. Esto facilita la integración de funcionalidades adicionales, como autenticación, manejo de bases de datos, renderizado de vistas, entre otros, para construir aplicaciones web más completas y robustas.

En resumen, Express.js es un framework ligero y flexible para Node.js que simplifica el desarrollo de aplicaciones web y API. Es ampliamente utilizado y apreciado por su simplicidad, rendimiento y extensibilidad.

