

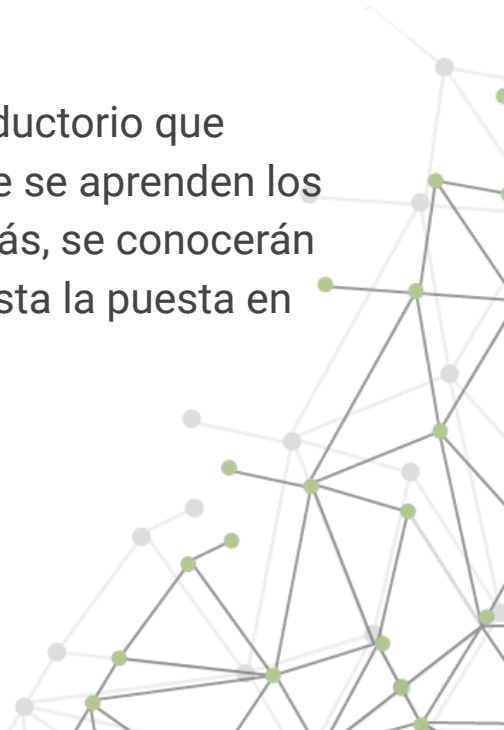


# HTML y CSS

Introducción a HTML

# ¿Qué aprenderemos en este módulo?

- Fundamentos de Desarrollo Web o Front End, es un módulo introductorio que entrega las herramientas básicas para el desarrollo web en donde se aprenden los conceptos base de desarrollo, entornos y flujos de trabajo. Además, se conocerán las buenas prácticas de desarrollo web desde la construcción hasta la puesta en marcha.



***Estructurar el contenido de un documento con HTML y utilizar componentes CSS para definir el aspecto visual de la página web.***

- Unidad 1: HTML y CSS
- Unidad 2: Bootstrap
- Unidad 3: JavaScript
- Unidad 4: Terminal, Git, GitHub y GitHub Pages



Te encuentras aquí



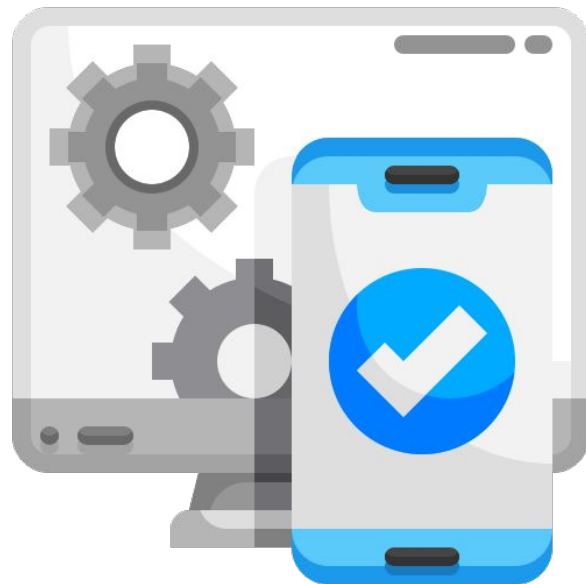
## ¿Qué aprenderás en esta sesión?

- *Implementar la estructura assets del proyecto (imágenes y archivos CSS), utilizando rutas locales, absolutas y/o CDN, de acuerdo a las buenas prácticas de la industria.*

***/\* ¿Qué es Desarrollo Web? \*/***

# ¿Qué es desarrollo web?

- Cuando hablamos de desarrollo, nos referimos al proceso de construcción de aplicaciones o sitios web, y dentro de este superfluo concepto podemos encontrarnos con distintos roles que conforman el espectro moderno de desarrollo web.



# Diferencias entre Front-End, Back-End y Fullstack

## *Front-end y back-end*



- **Front-end developer** es quien se encarga de desarrollar la interfaz gráfica que interactúa con los usuarios de un sitio web.

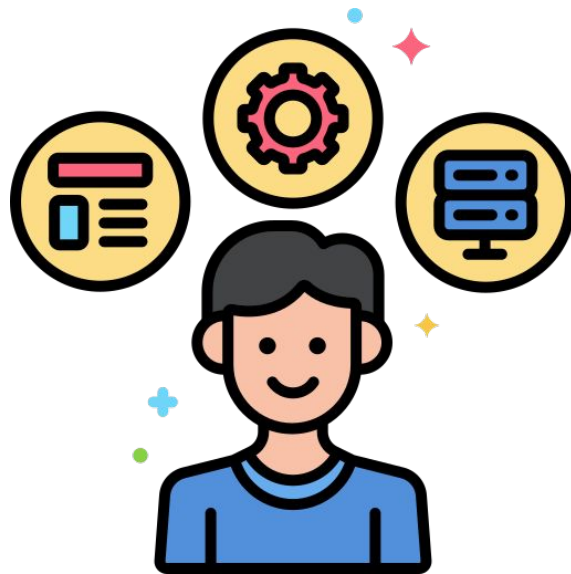


- **El Back-end** es responsable de implementar un sitio web o aplicación web y todos sus componentes, desarrollar la lógica y las soluciones para implementar correctamente las operaciones necesarias del sitio web o aplicación web.

# Diferencias entre Front-End, Back-End y Fullstack

## *Full Stack*

- El **Full Stack** es el rol que tiene conocimientos en todas las áreas y coopera con la comunicación efectiva y técnica de las piezas de rompecabezas.





# ¿Qué rol tiene el navegador en HTML? Investiguemos



# ¿Qué es la W3C? Investiguemos



***/\* ¿Qué es HTML? \*/***

# ¿Qué es HTML?

**HTML**

Sigla que proviene del inglés **HyperText Markup Language**, podemos traducir HTML como un lenguaje de marcas de hipertexto, que sirve para la elaboración de páginas web.

- **Estructura**

```
<p> Hola </p> <!-- esto es una etiqueta! -->
```



**/\* Estructura base de un documento  
HTML \*/**

# Estructura base de un documento HTML

La estructura básica de un archivo HTML consiste principalmente en:

1. **Cabeza** (`<head>`).
2. **Cuerpo** (`<body>`).

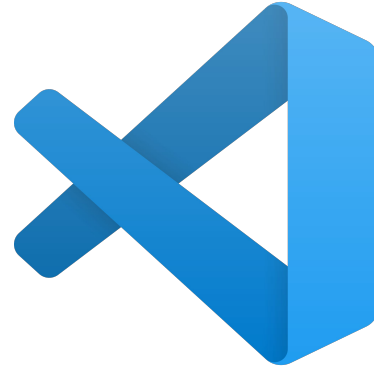
Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Aquí va la información para el navegador -->
  </head>
  <body>
    <!-- Aquí va el contenido para el usuario -->
  </body>
</html>
```

***/\* Herramientas \*/***

# Herramientas necesarias

## 1. Navegador Web



Fuente : [Visual Studio Code](#).



**/\* Ayuda para generar la estructura  
HTML \*/**

# Ayuda para generar la estructura HTML

## Autocompletado

- Uso:

1. Nos aseguramos de que estamos escribiendo en HTML.



Para profundizar y conocer más acerca de las funciones de VSC, revisa la [documentación oficial](#).

**{desafío}**  
latam\_

2. Escribimos html (o html:5) y apretamos la tecla **Tab**.

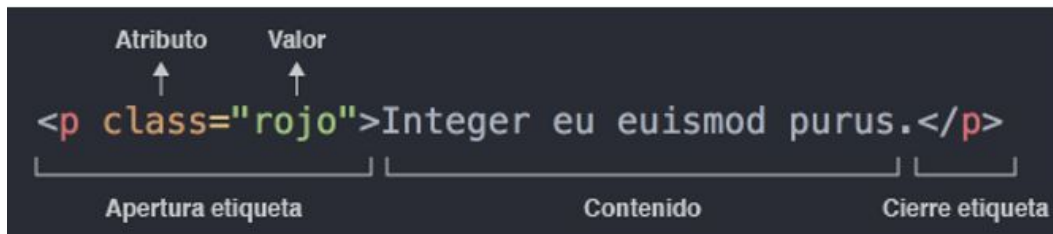
```
1  <!DOCTYPE html>
2  <html lang="en" dir="ltr">
3  <head>
4    <meta charset="utf-8">
5    <title></title>
6  </head>
7  <body>
8
9  </body>
10 </html>
```

**`/* Etiquetas */`**

# Etiquetas

## Estructura

- HTML se organiza en base a etiquetas, que son los elementos con los que puedes dar formato y estructura a un archivo HTML.
- Por lo general, la estructura, o sintaxis de una etiqueta es la siguiente:



Fuente: Desafío Latam.



Para conocer más en detalle la sintaxis de una etiqueta, revisa el siguiente [enlace](#).

# Etiquetas

## Ejemplos

```
<div>Hola soy una etiqueta Div </div>  
<button> Hola soy una etiqueta button </button>
```

- En este caso definiremos un nombre “div-contenedor” para así reconocer este <div> por el nombre de su clase (class):

```
<div class="div-contenedor">Hola soy una etiqueta Div </div>
```

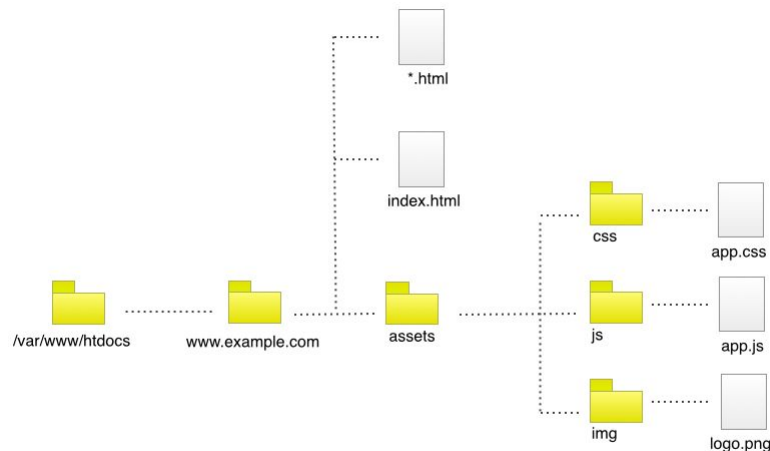
- Podemos agregar más atributos como id, type, alt, src, entre otros.

***/\* Estructura de Assets \*/***

# Estructura de carpetas

## Modelo de organización por tipo de archivo

Los archivos HTML se guardarán en el **directorio principal**, mientras que los recursos adicionales se almacenarán en subcarpetas dentro de una carpeta común llamada “**assets**”.

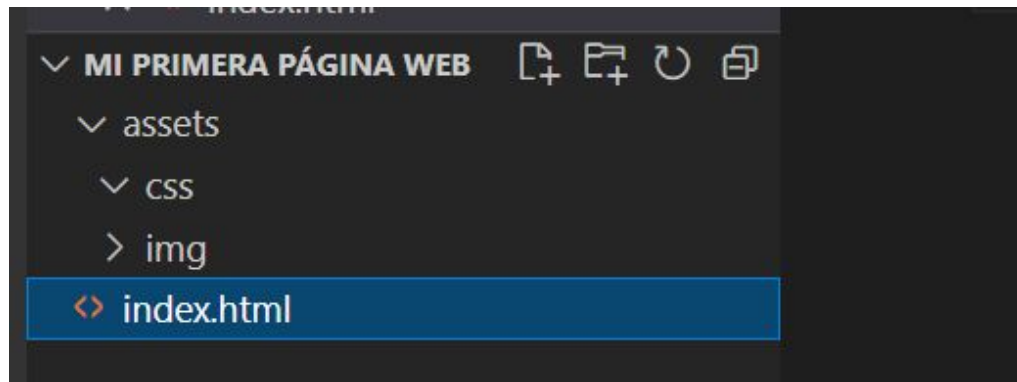


Fuente: [eamexicano](https://www.eamexicano.com)

# Estructura de carpetas

## *Modelo de organización por tipo de archivo*

“**Assets**” corresponde entonces a una convención que indica dónde se deben almacenar los archivos adicionales de nuestro proyecto, como **hojas de estilo (css)**, **Javascript (js)** o **imágenes (img)**.



Fuente: Desafío Latam.



¿Tienes instalado Google  
Chrome o algún otro  
navegador de internet en  
tu computador?



# Instalemos Visual Studio Code



# Resumen

- El **editor de texto**, que es un programa que se instala en el computador y que ocuparemos para escribir el lenguaje que luego interpretará el navegador.
- **HTML** es un lenguaje de marcado que se utiliza para estructurar y escribir contenido, el cual luego es interpretado por un navegador web.
- La **estructura básica de un archivo HTML** consiste en una cabeza (<head>) y un cuerpo (<body>). El <head> contiene toda la información que es para el navegador y el <body> contiene todo el contenido que es para el usuario.
- La **carpeta Assets** indica dónde se deben almacenar los archivos adicionales de nuestro proyecto, como hojas de estilo (css), Javascript (js) o imágenes (img).



## Próxima sesión...

- *Emplear adecuadamente la estructura y sintaxis de las etiquetas de un documento HTML, para dar solución a una problemática.*

**{desafío}**  
**latam\_**

*Academia de  
talentos digitales*

