

Curso Full Stack



Estas son algunas de las tecnologías de Full Stack

Frontend es la parte de un sitio web que interactúa con los usuarios, por eso decimos que se está del lado del cliente.

Un desarrollador frontend debe conocer los siguientes lenguajes y framework (marco de trabajo): HTML, CSS, Bootstrap, JavaScript

Backend es la parte que se conecta con la base de datos, un desarrollador backend

debe tener amplios conocimientos: lenguajes de programación: frameworks y los tipos de base de datos

Un desarrollador Full Stack es un programador con un perfil técnico muy completo, es el encargado de manejar cada uno de los aspectos relacionados con la creación y el mantenimiento de una aplicación web. Para ello es fundamental que el desarrollador Full Stack tenga conocimientos en desarrollo **Front-End y Back-End**

Uno de los sitios de consulta que vamos a utilizar a lo largo de todo el curso es [W3School](https://www.w3schools.com/)

Para trabajar en frontend sólo necesitamos un editor de texto plano y un navegador de internet.

Como editor de texto se pueden usar muchos que son gratuitos: Visual Studio Code, SublimeText, Atom, Notepad++, el que sugerimos que usen es Visual Studio Code, que lo pueden descargar del siguiente link:

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

HTML



QUÉ ES HTML?

HTML: Hyper Text Markup Language

Es un lenguaje de marcado para la elaboración de páginas web, donde se carga el contenido y cómo mostrarlo al navegador.

QUÉ NO ES HTML?

No es un lenguaje de programación.

No cuenta con la capacidad de realizar ningún proceso

Para comenzar un proyecto web, cree una carpeta contenedora, en ella un archivo llamado generalmente index.html, coloque las etiquetas básicas de HTML y comience a agregar en <body> las etiquetas con sus atributos y contenidos. Luego pruébelo en su navegador de internet como se ve. Recuerde que para ver los cambios en el sitio, tiene que grabar en el editor de texto y refrescar en el navegador su página (Ctrl+F5). Más adelante, dentro de la carpeta iremos creando una subcarpetas de Imágenes, de Estilos CSS, otra de Javascript etc.etc.

Ejemplo Básico de una página web:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

  <!-- Acá va el contenido de la página web -->

</body>
</html>
```

Tag o etiquetas

El marcado en HTML incluye "elementos", etiquetas o tag que tiene el siguiente formato:

```
<etiqueta atributo="valor" ..... atributo="valor" >contenido</etiqueta>
```

Casi todas las <etiqueta> tienen un cierre que se indica con </etiqueta>, salvo algunas como
, que hace un salto de línea, <hr> hace una línea horizontal o para mostrar imágenes y no necesitan el cierre de la etiqueta

Etiquetas especiales:

<head> Marca la cabeza de la página, donde se configura, el tipo de archivo, el lenguaje etc.

<!DOCTYPE html> muestra que el tipo de archivo es HTML

<html lang="en"> indica el lenguaje

<title> Se utiliza para colocar el título en la barra del navegador

<style> Es la etiqueta para darle estilo al sitio, ya sea con un archivo externo, o para indicarlos dentro del mismo HTML

`<script>` Es la etiqueta para darle programación con javascript ya sea con un archivo externo, o para hacerlo dentro del mismo HTML

`<body>` Etiqueta que marca el comienzo del cuerpo de la página HTML

Más etiquetas:

Un comentario de bloque: comienza con `<!--` y finaliza con `-->`

Ejemplo:

`<!-- Esta línea no se ejecuta, sirve para colocar comentarios para el programador -->`

`<p>` Etiqueta `p` indica un párrafo

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>` Se utiliza para escribir los títulos o encabezados, siendo `<h1>` el más importante y `<h6>` el menos. Recuerde no utilizar más de una etiqueta `<h1>` por página, sería como un libro con dos títulos

`
` La etiqueta `br` produce un salto de línea

`<hr>` hace una línea horizontal

Etiquetas de texto

`` - **Texto en negrita o bold**

`<mark>` - **Texto marcado**

`<ins>` - Texto insertado

`<small>` - Texto más pequeño

`<i>` - *Texto en Itálica o cursiva*

`` - ~~Texto eliminado~~

`<sup>` - Texto en superíndice

`<sub>` - Texto del subíndice

Direcciones absolutas y relativas:

Los link están formados por una serie de rutas (path), donde indica la dirección a la que tiene que ir el navegador cuando pulsamos sobre ese link o cuando busca una imagen, o un archivo.

Hay dos tipos de direcciones:

Absolutas: Incluye el nombre del dominio. Ejemplo:

`'http://i.imgur.com/SpZyc.png'`

Relativa: Indica el orden de directorios a partir de donde estoy. Ejemplo:

`'/imagenes/flor.gif'`

`` Se utiliza para colocar las imágenes. Tiene varios atributos, src es obligatorio y sirve para indicar la dirección de la misma. Otros atributos son width (ancho), height(alto), alt sirve para cuando no puede encontrar la imagen, indicarle que tendría que haber aparecido..

`<a>` El elemento `<a>` crea un link o enlace a otras páginas. Tiene el atributo obligatorio href donde se indica el link. Se utiliza para ir navegando de página en página y no tiene etiqueta de cierre

<https://Jsbin.com>

Jsbin es una herramienta para experimentar con lenguajes web. En particular HTML, CSS y JavaScript

Para hacer pruebas en Jsbin:

En el navegador colocar jsbin.com, y se abre una página partida en dos, de la izquierda el código HTML, de la derecha como se vería en el navegador, más adelante podremos colocarle estilos de CSS y programas de Javascript.

También se puede compartir lo que se se hizo en el proyecto, haciendo click en [Share]

Ejemplo:

```

</head>
<body>
  <!-- esto es un comentario y no se ejecuta -->
  <h1>Este es el encabezado principal</h1>
  <h2>Este es un encabezado secundario</h2>
  <h3>Este es un encabezado h3</h3>
  <h4>Este es un encabezado h4</h4>

  <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit,
    sed do eiusmod tempor incididunt ut labore
    et dolore magna aliqua.
  </p>

  <a href="https://google.com">ir a google</a>
  <a href="https://www.knorr.com">ir a Knorr</a>
  <a href="https://www.eladerezo.com">ir a eladerezo</a>
  <a href="https://www.jsbin.com">ir a jsbin</a><br><br>

  
    
</body>
</html>

```

Output

Run with JS Auto-run JS

Este es el encabezado principal




Este es un encabezado secundario

Este es un encabezado h3

Este es un encabezado h4

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[ir a google](https://google.com) [ir a Knorr](https://www.knorr.com) [ir a eladerezo](https://www.eladerezo.com) [ir a jsbin](https://www.jsbin.com)

<https://jsbin.com/loripaw/edit?html,output>

Ejemplo para enviar un mail

```

<a href="mailto://ejemplo@gmail.com">envie un mail a
ejemplo@gmail.com</a>

```

Imágenes como enlaces

Para poner una imagen como enlace basta con crear un enlace y añadir en su interior en lugar de texto, una imagen. Veamos un ejemplo:

```

<a href="http://aprenderaprogramar.com">
  
</a>

```

Elementos en bloque o en línea:

Los navegadores suelen mostrar el elemento a nivel de bloque con un salto de línea antes y después del elemento, como las etiquetas <p>, <h>

Los elementos en línea pueden comenzar en cualquier parte de una línea como las etiquetas <a>
