

Curso Definitivo de HTML y CSS

Qué aprenderás sobre HTML y CSS

HTML

Estructura de los documentos que se visualizan en los navegadores (Chrome, Edge, Firefox, Safari, Opera, etc) Es el lenguaje con el que clasificamos o etiquetamos cada parte del documento, dejándoles saber al navegador, que información va a ver

De esa manera, presentamos: Barras de Navegación, Secciones, artículos, listas, títulos de diferentes tamaños, audios, videos, imágenes, entre otros.

CSS

Es el lenguaje con el que le damos estilos a todo el contenido previamente etiquetado con HTML. Es el encargado o usado para darle todo el aspecto visual de manera profesional, usando estándares de diseño de documentos y logrando así la correcta visualización de textos y recursos audiovisuales en formatos digitales, en nuestro caso, las páginas web.

JAVASCRIPT

Es lenguaje que se usa para crear o brindar a las páginas web la capacidad de interactuar con el usuario y los recursos asociados a una página web, como lo son: bases de datos, servidores de diversa índole, información derivada de la interacción de cada página en sí.

¿Qué es el Frontend?

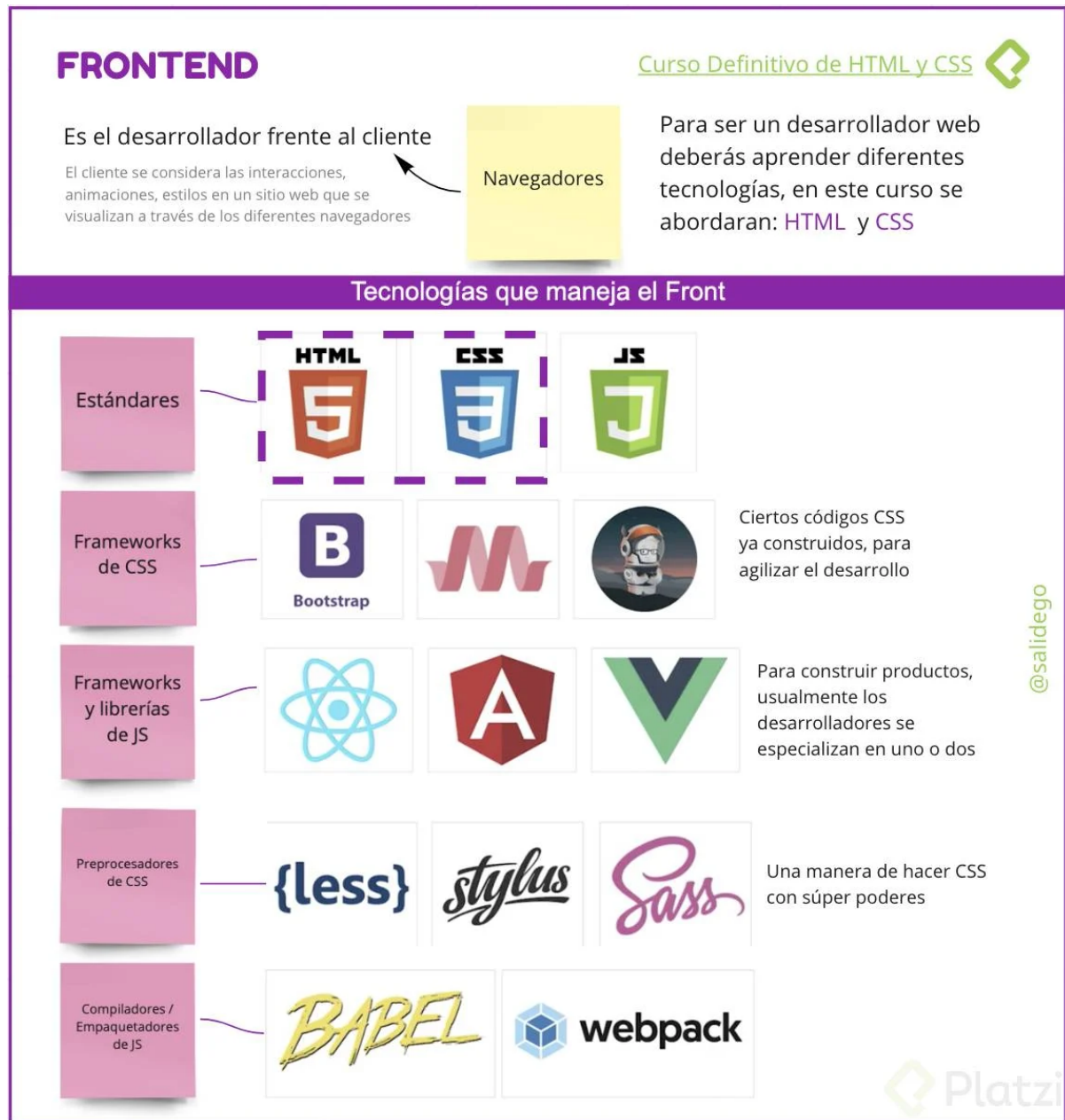
Front-end: Es alguien que se dedica al desarrollo en el lado del cliente, estas pueden ser las interacciones que tienes en una página web, las animaciones y los estilos son cosas que suceden en el cliente.

Los Front-end utilizan las tecnologías de HTML, CSS y JS, que son estándares que entiende el navegador.

También utilizan Frameworks de CSS, lo cual nos puede facilitar fragmentos de CSS ya contruidos para que podamos agregarlos a nuestros sitios web! Es una ayuda para acortar tiempos de desarrollo!

Y espera! También tenemos los Frameworks y librerías de JS, los cuales nos permitirán construir productos que puedan escalar más rápido y que tengan una interacción mayor. Y por si fuera poco. Tenemos los Preprocesadores de CSS, podríamos decir que es CSS con super poderes!

Para terminar tenemos compiladores / empaquetadores de JS como Babel y Webpack



Back-end

Es lo opuesto a un Front-end, ya que el Back-end trabaja en el lado del servidor mientras el Front-end en el lado del cliente.

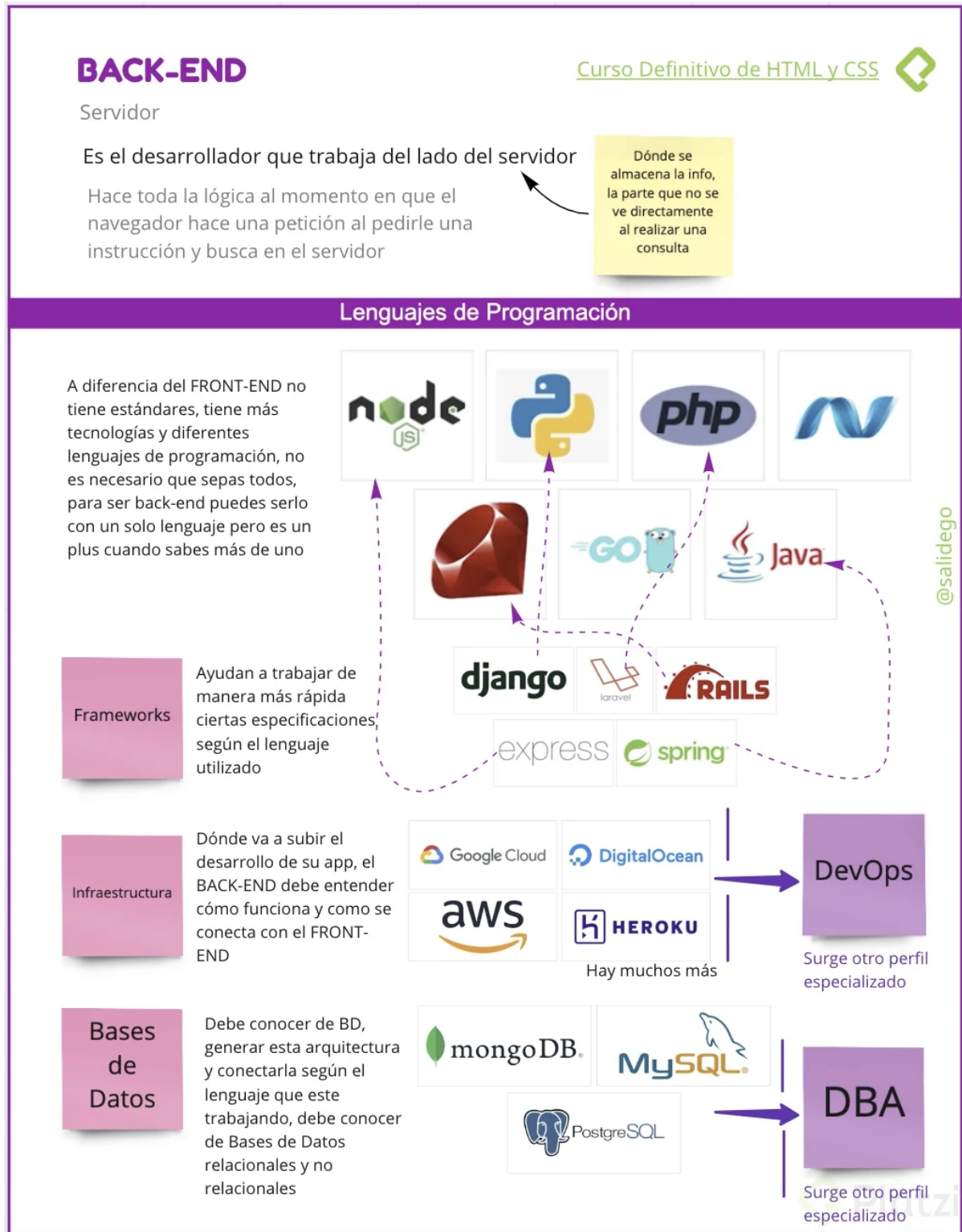
Es el responsable de manejar toda la lógica que hay detrás de una petición dada por el navegador.

Una característica que lo diferencia del Front-end es que no tiene estándares, ya que tiene varios lenguajes de programación (Node.js, Python, PHP, Ruby, GO, Java, .NET entre otros) . Los cuales estos lenguajes tienen sus propios frameworks como Django (Python), Lavarel (PHP), Rails (Ruby), Express (JS), Spring (Java).

El Back-end también tiene que tener en cuenta la infraestructura, donde va a generar el deploy de su aplicación (esto también puede ser tarea de un DevOps, un perfil dedicado a la infraestructura). Con tecnologías como Google Cloud, DigitalOcean, AWS y Heroku.

Y por último entramos en Bases de datos, teniendo las relacionales (mongoDB) y las no-relacionales. (MySQL). Ahora que tienes más noción de que trata cada uno... ¿Cuál te llama más la atención? ¿Cuál te gustaría aprender? ¡En los comentarios podemos seguir esta charla! 😊😊

Utiliza lenguajes de programacion/ frameworks que corran del lado del servidor, infraestructura en la nube y base de datos.

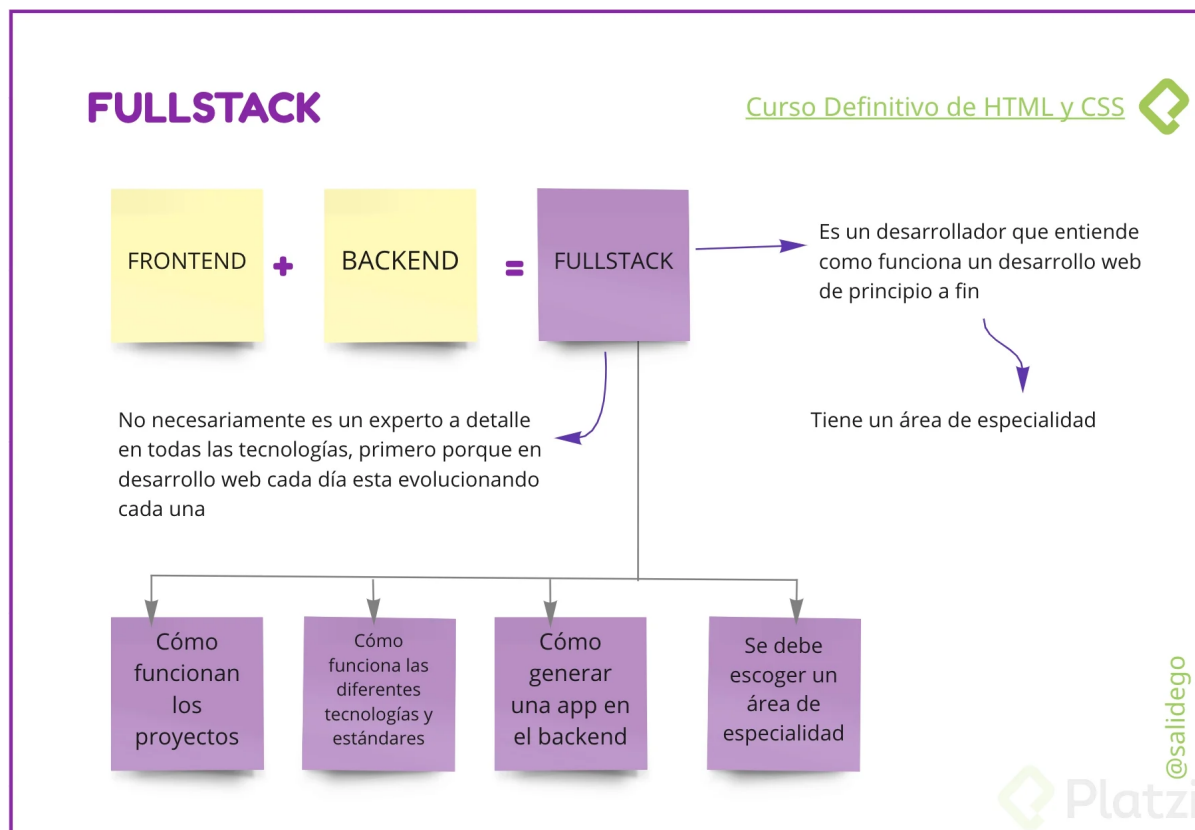


Desarrollador Fullstack

es alguien que tiene conocimientos tanto del Front-end como del Back-end, es alguien que entiende como funciona un producto web desde principio a fin. Desde la idea hasta la producción!

Sin dudas tiene un alto conocimiento acerca de una aplicación web.

Pero Hey! Ellos tienen sus áreas de especialización, ya sea Front-end o Back-end Espero ver sus aportes relacionados a un Desarrollador Fullstack, es a mi parecer, un mundo bastante amplio!



Páginas Estáticas vs. Dinámicas

- **Sitos Web Estáticos (Páginas informativas):** La información que contiene, se mantiene constante y estática. No se actualiza con la interacción del usuario. Conveniente para landing pages (Páginas informativas) o Blogs. Serán siempre iguales para todos los usuarios.
- **Sitos Web Dinámicos (Aplicaciones Web - Web Apps):** Actualizan su información con respecto a la interacción del usuario. Dependen de una base de datos, de donde extrae e ingresa información. Serán diferentes, dependiendo del usuario que la use.

PÁGINAS ESTÁTICAS VS DINÁMICAS

Curso Definitivo de HTML y CSS

Estática

El contenido es fijo una vez se sube

- Blog post
- Páginas informativas - Landing pages
- No están conectadas a un servidor o base de datos
- Pueden tener navegación entre páginas

Se pueden actualizar esporádicamente, pero siguen siendo estáticas

Ejemplos:

- Un Blog
- Una página empresarial que sea de carácter informativo

Dinámica

- Web Apps
- Aplicaciones

- Si están conectadas a una base de datos
- Se debe contar con un login de acceso a cuenta

Ejemplos:

- Twitter: al ingresar cada vez, el contenido es diferente y yo puedo interactuar y la info se almacena en una base de datos
- Platzi

El desarrollo del tipo de página dependerá de las necesidades del cliente

@salidego

Platzi

HTML: anatomía de una página web

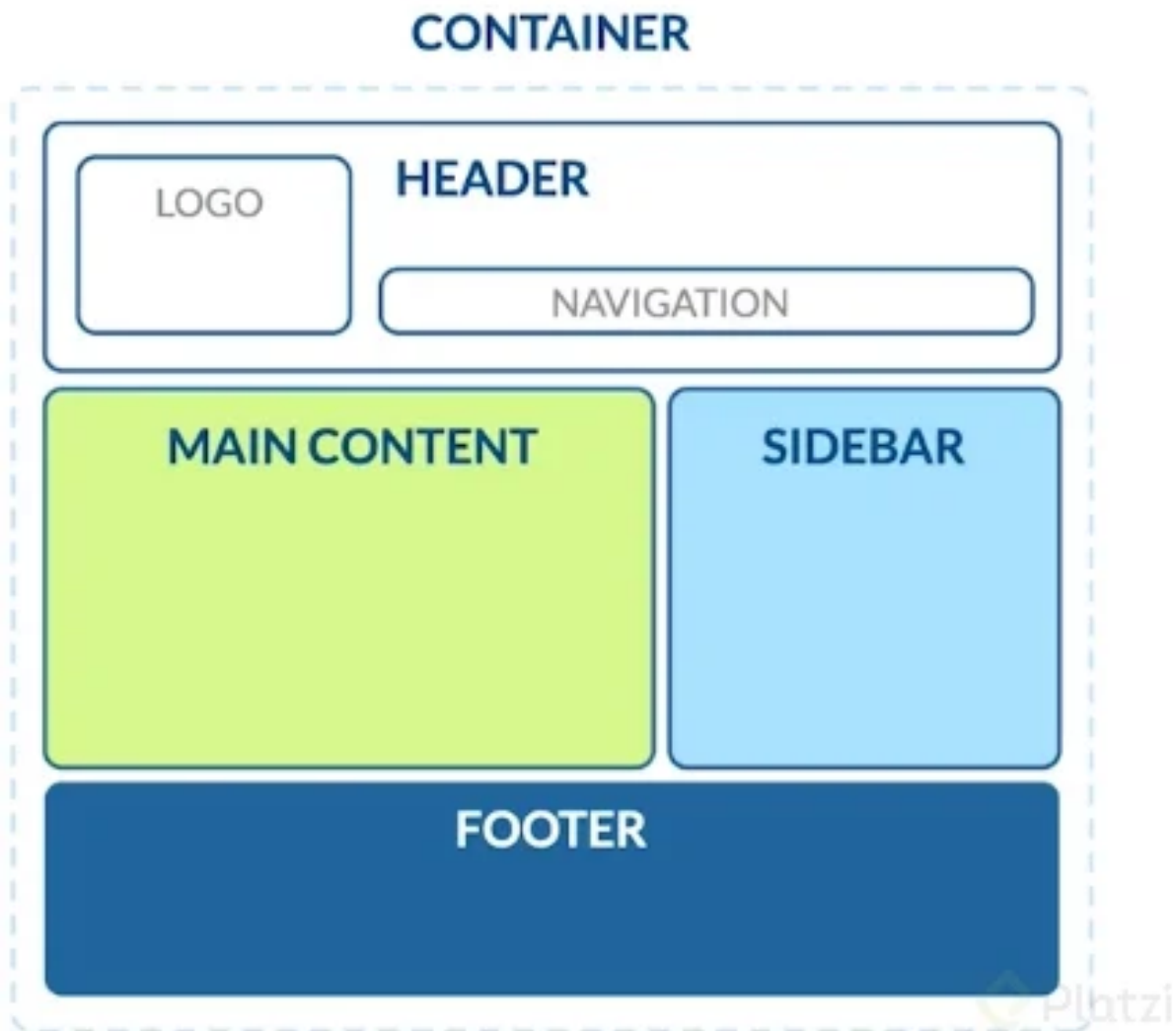
La web se conforma de tres conceptos:

URL => Uniform Resource Locator. HTTP => Protocolo de transferencia de hipertexto. HTML => código que se utiliza para estructurar el contenido de tu web, y darle sentido y propósito.

- **HTML** son siglas, Hyper Text Markup Language (**Lenguaje de Marcado de Hipertexto**). Hyper Text significa que el texto tiene interactividad, conexión con otros documentos. _Markup _significa que le pone etiquetas a los elementos. Por eso también se le conoce como un lenguaje de etiquetas. HTML es un lenguaje interpretado. HTML es un estándar, así que no importa desde que navegador, que dispositivo se ejecute, es lo mismo en cualquier sitio. Sigue siendo el mismo código.

Anatomía de una página Web:

Container.- Contenedor principal Header.- Cabecera de la página Main content.- Estructura principal, por ejemplo el post de un red social. Sidebar.- Contenido secundario de una página. Footer.- Pie de página.



Hay etiquetas en HTML que nos ayuda a diferenciar en donde estamos.

Index y su estructura básica: head

En el head van todos los archivos importantes para que nuestro proyecto funcione correctamente, algunos como: Estilos, fuentes, descripciones, librerías... Es la parte no visible de nuestra página.

```
<!DOCTYPE html>
<!--Le decimos al navegador que este
archivo es del tipo html:5-->

<html lang="es">
  <!--Es la etiqueta "padre" donde vivirá nuestro proyecto.
  El atributo lang establece el idioma del sitio web.
  Debemos usarlo para que el navegador pueda traducir
  nuestra página-->

  <head>

    <meta charset="UTF-8" />
    <!--Este atributo nos ayuda a la hora de incluir
    caracteres especiales y emojis en nuestro proyecto-->

    <meta name="description"
      content="Esta página te mostrará fotos de gatos" />
    <!--Muestra una descripción de nuestro sitio
    en los buscadores-->

    <meta name="robots" content="index,follow" />
    <!--Le dice a los robots de los navegadores que
    rastreen nuestra página y la muestran en las búsquedas-->

    <title>Mi página</title>
    <!--Título de nuestra página, no confundir con
    los H1-H6. Este titulo es el que ves en la pestaña
    del navegador-->

    <meta name="viewport" content="width=device-width,
    initial-scale=1.0" />
    <!--Nos ayuda a trabajar en proyectos responsive-->

    <link rel="stylesheet" href="./css/style.css">
    <!--Linkea/Enlaza archivos de estilos u otros
    archivos que necesitemos en nuestro proyecto-->

  </head>
```

Index y su estructura básica: body

body es la parte visible de nuestro website 😊. Es muy importante usar HTML semántico y no llenar todo de para que nuestro sitio sea mejor interpretado por el navegador y por lo tanto más accesible.

```
<body>

<header> <!--Sección superior de nuestro website-->

    <nav></nav> <!--Sección de navegación de nuestro
    website, siempre dentro del header-->

</header>

<main> <!--Main es el contenido central de
nuestro website, "la parte del medio"-->

    <section>
        <!--Nuestro website puede estar dividido por
        secciones, por ejemplo platzi tiene 3:
        El navegador de cursos y rutas, el feed
        y nuestras rutas de aprendizaje-->

        <article>
            <!--Contenido independiente de la página.
            Es reutilizable-->
        </article>

    </section>

    <ul> <!--Lista desordenada: Sin numerar-->

        <li><!--Item List. Elementos de la lista--></li>

    </ul>

    <ol></ol> <!--Lista ordenada: Numerada-->

</main>

<footer> <!--Sección final de nuestro website-->

</footer>

<p>Soy un texto</p> <!--Párrafo, texto-->

<h1>Soy un titulo</h1>
<!--Títulos, muestran el texto más grande y
con negrilla. Existen desde el h1 al h6-->

<a href="#">Soy un link</a>
```



```
<!--Enlaces/links que nos permitirán movernos  
entre páginas.-->
```

```
</body>
```

Anatomía de una etiqueta de HTML

¿Qué son los atributos?

Podemos pensarlos como instrucciones que se incluyen dentro de la etiqueta de apertura de una etiqueta HTML y determinan el comportamiento de ésta última. Generalmente se asocian a un valor, el cual suele escribirse entre comillas dobles.

Existen tres tipos de atributos, los cuales se distinguen mediante los valores asociados a ellos. Un atributo puede tener como valor:

1. Conjunto finito de valores.
2. Valores libres (como una URL o un texto).
3. Valores booleanos.

En HTML5, los atributos de valor booleano se escriben sin valor asociado sólo en caso de que dicho valor sea verdadero, y se omite su mención cuando su valor es falso.

