[HTML picture tag (w3schools.com)](https://www.w3schools.com/tags/tag_picture.asp) css

**Qué aprenderás sobre HTML y CSS**  1/55

[El camino para convertirte en Desarrollador Web este 2018 | by Martin Mucito | Code Army](https://codearmy.co/el-camino-del-desarrollador-web-en-2018-64efc5b68853)

En resumida cuenta según lo que he podido entender y explicándolo con analogías.  
.  
HTML sería la estructura, como si fuese los esqueletos de un ser humano o los cimientos de un edificio, de las páginas web sin colores, sin diseño.  
.  
CSS sería lo estético como la piel de ser humanos, la pintura y adornos de los edificios o la manera como es posible que la estructura del html se vea mejor para la vista de los usuarios.  
.  
JS seria los músculos lo que da esa interactividad a las personas para moverse y hacer acciones como correr.  
example: [css html js-c5cad547-d054-48d7-83be-bc2caee99577.jpg (472×485) (platzi.com)](https://static.platzi.com/media/user_upload/css%20html%20js-c5cad547-d054-48d7-83be-bc2caee99577.jpg)

<https://commondenominator.email/wp-content/uploads/2017/06/javascript-seo-making-your-bot-experience-as-good-as-your-user-experience.gif>

**¿Qué es Backend?**

3/55

**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** tambien tiene que tener en cuenta la infraestructura, donde va a generar el deploy de su aplicación (esto tambien 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… **¿Cual te llama más la atención?** **¿Cual te gustaría aprender?** ¡En los comentarios podemos seguir esta charla! 😃😃

**HTML: anatomía de una página web**

6/55

[Web Anatomy-296b297a-6db0-4747-b96d-bcf4a7f5e569.jpg (1654×922) (platzi.com)](https://static.platzi.com/media/user_upload/Web%20Anatomy-296b297a-6db0-4747-b96d-bcf4a7f5e569.jpg)

**Index y su estructura básica: body**

8/55

**body** es la parte visible de nuestro website 😃. Es muy importante usar HTML semántico y no llenar todo de **<div>** 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 divido 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>