



POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Educación para
vivir mejor

Construcción de elementos de software web 1

Semana 2

Fundamentos de HTML y CSS - Básico

Media Técnica en Programación de Sistemas de Información
Politécnico Jaime Isaza Cadavid - 2020



Politécnico Colombiano Jaime Isaza Cadavid



@PolitecnicoJIC



Fundamentos de HTML

Dentro del desarrollo web, como hemos visto existen varios roles que están involucrados para poder crear páginas y productos web excepcionales, en el transcurso de los conceptos y lenguajes que vamos a aprender y a profundizar van estar orientados a todo el desarrollo que es invertido para ofrecer una excelente experiencia de usuarios a los clientes y/o usuarios que usan nuestra aplicación, es esta la labor de un desarrollador frontend.

Todo desarrollador frontend debe tener una caja de herramientas fundamental para el éxito de sus productos que debe codificar, donde debe tener una gran habilidad haciendo uso de HTML, CSS y Javascript.

En primera instancia vamos a hablar sobre HTML, lenguaje orientado a brindar la estructura o cimientos de lo que va ser nuestro producto web.

¿Qué es HTML?

HTML (Hypertext Market Lenguaje) es un lenguaje de marcado, nos permite codificar un documento en el cual incorporamos etiquetas o marcas que van a contener la información de nuestra página web, es decir, va a empezar a brindarnos una estructura identificando por ejemplo, un título, un subtítulo, una imagen o un párrafo.

Cuando identificamos elementos que van a hacer parte de nuestra estructura haciendo uso de HTML, estos elementos suelen estar formados por dos etiquetas, una de apertura y una de cierre, y así, dentro de estas etiquetas vamos a colocar el contenido que deseamos este encapsulado en dicho elemento, donde el contenido puede ser texto y/o otras etiquetas HTML.

Las etiquetas de apertura pueden incluir **atributos**, los cuales se encargan de modificar el comportamiento que tiene por defecto el elemento o nos permiten aportar información extra a ese elemento, también estos atributos pueden ser denominados modificadores.

Por ejemplo, el siguiente conjunto de etiquetas crean un párrafo con el atributo “lang” donde se indica que este párrafo está en lenguaje español

```
<p lang="es">Párrafo</p>
```



Pero también debemos saber, que existen una serie de elementos HTML que no necesitan una etiqueta de cierre, estos los iremos explorando más adelante.

De acuerdo a lo que hemos visto hasta ahora, para tener un lenguaje unificado a los elementos HTML que vayamos a incorporar y que van a hacer parte de nuestro producto web los vamos a denominar **etiquetas**.

Así podríamos decir que un documento HTML, nos permite tener una estructura definida de nuestra página web donde vamos a alojar todo el contenido ya sean imágenes, textos, títulos videos o secciones que van a definir y materializar nuestra página web, en este punto podemos afirmar que van a existir dos tipos de etiquetas:

- Etiquetas que actúan como contenido, es decir que definen el documento o información que deseamos mostrar y/o visualizar.
- Etiquetas que actúan como contenedoras, las cuales van a encapsular un conjunto de etiquetas ya sean otras etiquetas contenedoras o varias etiquetas de contenido que nos vana ayudar a clasificar y organizar nuestra estructura HTML.

Etiquetas de Página

El estándar actual de HTML se denomina HTML5, el cual permite usar etiquetas que mejoren la semántica de un sitio web, es decir, en el momento en que empezamos a crear nuestras estructuras e identificamos los distintos componentes que va a tener nuestra página web, necesitamos conocer y aplicar cual es la etiqueta más apropiada para alojar este contenido (recordemos que también van a existir etiquetas contenedoras que nos van a ayudar en este proceso), de esta forma al escoger muy bien dichas etiquetas empezamos a darle un significado especial a nuestro sitio web.

Al principio de un documento HTML se incluye `<!DOCTYPE html>` de esta forma nuestro navegador interpreta que estamos usando el estándar de HTML5, generando así una descripción de estructura y sintaxis de nuestro documento.

Contiguo a ello, una página web empieza con una etiqueta que indica que vamos a crear una página HTML, haciendo uso de `<html>`. Dentro de ella va una cabecera o `<head>`, donde se va a definir aspectos relativos al contenido, o metainformación como el título, descripción o palabras claves.



El último elemento pero no menos importante es el cuerpo de nuestro documento, o `<body>` , donde incluiremos el contenido de nuestra página.

Así una página HTML con cabecera y cuerpo se vería de la siguiente manera:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5
6   <body>
7   </body>
8 </html>
```

En este punto es clave que hablemos sobre la **sintaxis** de nuestras páginas web, ya que cuando escribimos código debemos cumplir un orden y relación de acuerdo a los conjuntos o bloques de código que estamos creando, esto con el fin de poder identificar qué conjunto de etiquetas hacen parte de una etiqueta que actuará como contenedora.

Retomando el ejemplo anterior podemos realizar las siguientes descripciones a cada una de las etiquetas que están contenidas en la etiqueta HTML:

- **head:** dentro de esta etiqueta se colocan elementos que el navegador va a usar para pero no se van a mostrar en el cuerpo de la página. Aquí van elementos como el título de nuestra página, la imagen que aparece en la pestaña del navegador, o las palabras clave para indexar en buscadores.
- **body:** Todos los elementos que se escriban dentro de esta etiqueta se van a mostrar en nuestra página web. aquí van las etiquetas de texto, imágenes, botones, entradas de texto, formularios y demás.

Ahora veamos un ejemplo donde tenemos como base el ejemplo anterior, pero vamos a crear una estructura un poco más definida, en este caso en la etiqueta `<html>` se le indica que este documento va a tener su contenido en lenguaje español, donde se le incorpora el atributo `lang`.

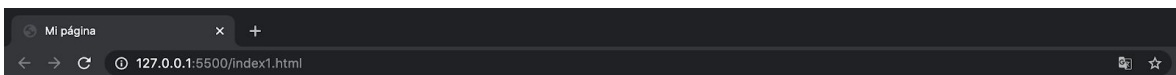


Además hay dos etiquetas adicionales en la cabecera; una que indica la codificación del texto y otra que indica el título del documento.

Es muy importante como buena práctica de codificación acostumbrarnos a usar el atributo “lang” en nuestra etiqueta <html>, de esta forma dejamos claro cuál es el idioma en el que está escrito nuestro contenido.

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

Observe que la etiqueta meta es una de esas etiquetas especiales que podemos notar que para su implementación no necesitan cerrarse y la etiqueta title nos va a mostrar el texto introducido entre las etiquetas de apertura y cierre en la pestaña del navegador, como se evidencia en el siguiente ejemplo:



Cuando vemos nuestro documento en el navegador, debe notar que simplemente se muestra de cara al usuario la información que alojamos en la etiqueta <title>.



Codificación de una página HTML

Cuando hicimos uso de la etiqueta meta, mencionamos que esta etiqueta nos indica la codificación del texto, es importante en este punto hacer una pausa y saber que podemos usar un formato de codificación de caracteres al crear nuestra página, cada formato tiene cierta capacidad de caracteres disponibles, entonces podría suceder que las tildes o caracteres especiales, por ejemplo que usamos en el español, podrían no estar disponibles.

En nuestro caso particular vamos a usar UTF-8, que según la comunidad y de acuerdo a los estándares es el más completo.

La codificación de un documento se indican en dos pasos esenciales:

1. El archivo se guarda usando una codificación específica.
2. En el head de la página se incluye la etiqueta `<meta charset="">` indicando al navegador el formato de codificación de caracteres que hemos usado al guardar el documento.

Introducción a las etiquetas de contenido

El navegador va a interpretar y hacer una lectura de las etiquetas que vayamos alojando en nuestro documento en un orden de escritura, es decir, de arriba hacia abajo, y va a intentar mostrarlas en ese orden.

El buen uso e implementación de estas etiquetas, van a producir que se añada a nuestro contenido una valoración semántica. Recordemos que la semántica al proporcionarnos el significado de las expresiones lingüísticas (en este caso de las etiquetas que incorporamos en HTML) resultan siendo importantes para los siguientes dos aspectos:

1. Hacer sitios web más accesibles: la accesibilidad, ayudará, por ejemplo, cuando se consulta la página usando algún sistema de soporte como lectores de pantalla.
2. SEO (Search Engine Optimization) : Posicionamiento en buscadores o motores de búsqueda, como Google o Bing , facilitando que nuestra página aparezca en las búsquedas donde se usen palabras claves que están relacionadas con nuestros productos web.

Así resulta siendo importante como marcamos un texto, por ejemplo, si marcamos un texto como encabezado le estamos asignando una importancia diferente a que fuera marcado como párrafo o elemento de una lista.



Teniendo claro lo que acabamos de mencionar vamos a indicar y a mostrar algunas etiquetas de contenido que nos van a permitir crear nuestros primeros elementos en HTML:

Títulos o encabezados

Se indican con las etiquetas <h1> hasta <h6>, y van desde un título muy relevante hasta el menos relevante.

1	<code><h1>Encabezado de nivel 1</h1></code>	Encabezado de nivel 1
2	<code><h2>Encabezado de nivel 2</h2></code>	Encabezado de nivel 2
3	<code><h3>Encabezado de nivel 3</h3></code>	Encabezado de nivel 3
4	<code><h4>Encabezado de nivel 4</h4></code>	Encabezado de nivel 4
5	<code><h5>Encabezado de nivel 5</h5></code>	Encabezado de nivel 5
6	<code><h6>Encabezado de nivel 6</h6></code>	Encabezado de nivel 6

Párrafo

Con la etiqueta <p> definiremos una etiqueta del tipo párrafo y vamos a indicar que su contenido va a ser un párrafo de texto

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lista de elementos

Hay momentos donde vamos a necesitar crear o agregar una serie de elementos e indicar que estos se encuentran relacionados: para ello tenemos las listas ordenadas `` y las listas no ordenadas, o desordenadas ``. En ambas usamos `` para marcar un elemento de la lista. Aquí puede evidenciar como la etiqueta `` y la etiqueta `` actúan como etiquetas contenedoras y las etiquetas `` como etiquetas de contenido.



```
1 <ol>
2   <li>Elemento de la lista</li>
3   <li>Elemento de la lista</li>
4   <li>Elemento de la lista</li>
5 </ol>
6
7 <ul>
8   <li>Elemento de la lista</li>
9   <li>Elemento de la lista</li>
10  <li>Elemento de la lista</li>
11 </ul>
```

1. Elemento de la lista
2. Elemento de la lista
3. Elemento de la lista

- Elemento de la lista
- Elemento de la lista
- Elemento de la lista

La lista ordenada nos muestra una lista enumerando cada ítem por orden de escritura y la lista no ordenada añade un símbolo (por defecto es un círculo relleno) delante de cada elemento de la lista, también por orden de escritura.

Elementos en bloque (*block*) y en línea (*inline*)

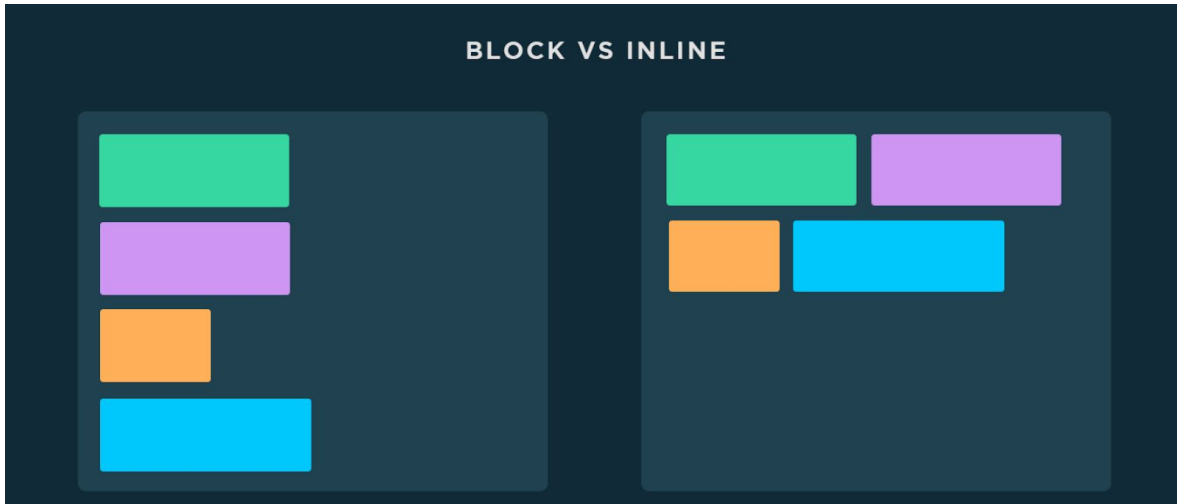
Cada etiqueta y/o elemento en HTML tiene un valor de **display** por defecto, esto depende del tipo de elemento que se vaya a utilizar. El valor por defecto para la mayoría de los elementos es **block** o **inline**.

Un elemento a nivel de block, siempre empieza en una línea nueva y toma el ancho completo disponible, es decir, se estira hacia la izquierda y hacia la derecha tan lejos como pueda.

Un elemento inline no empieza en una línea nueva, y solo toma el ancho que necesite tomar.



En la siguiente imagen podemos evidenciar un ejemplo de block vs inline:



Ahora los elementos de tipo bloque en html son:

`<address>` `<article>` `<aside>` `<blockquote>` `<canvas>` `<dd>` `<div>` `<dl>` `<dt>`
`<fieldset>` `<figcaption>` `<figure>` `<footer>` `<form>` `<h1>-<h6>` `<header>` `<hr>` ``
`<main>` `<nav>` `<noscript>` `` `<p>` `<pre>` `<section>` `<table>` `<tfoot>` `` `<video>`

Y los elementos de tipo inline en html son:

`<a>` `<abbr>` `<acronym>` `` `<bdo>` `<big>` `
` `<button>` `<cite>` `<code>` `<dfn>` ``
`<i>` `` `<input>` `<kbd>` `<label>` `<map>` `<object>` `<output>` `<q>` `<samp>` `<script>`
`<select>` `<small>` `` ``



Fundamentos de CSS

Cuando vamos maquetando una página web podemos empezar a relacionar este proceso con actividades que hemos hecho antes en nuestras vidas y que no tienen relación con el desarrollo de una página web, por ejemplo, cuando realizamos una maquetación en papel o estamos realizando apuntes interesantes en nuestros cuadernos de una clase para que no se nos olviden esos conceptos esenciales.

Partiendo de esta idea, es normal que a estos apuntes les demos un poco de vida usando distintos colores ya sea para los títulos o realizando una representación gráfica del concepto, es decir, otorgamos estilos para definir la apariencia que va a tener este contenido permitiendo así que visualmente sea más llamativo.

¿Qué es CSS?

En el desarrollo de un sitio web para otorgar estilos hacemos uso de CSS (“Cascading Style Sheets”) hojas de estilo en cascada, las cuales ayudan a definir la presentación de un documento escrito en un lenguaje de marcado, como lo es HTML.

CSS tiene una sintaxis simple, y usa un conjunto de palabras claves en inglés para especificar los nombres de varias propiedades de estilo.

Ahora hasta este punto es claro que sean hojas de estilo que nos otorgan una mejor presentación del contenido que hemos implementado en nuestro HTML, pero ¿Qué significa que sean “en cascada”? Esto hace referencia al proceso de combinación y aplicación de estilos en CSS, teniendo en cuenta que hace una lectura de inicio a fin pero que prevalecen ciertas prioridades y la forma en que se van a resolver los conflictos entre ellos.

Entonces, en una hoja de estilos CSS, vamos a tener un conjunto de reglas que dirán como se tienen que mostrar nuestros elementos.

En ese orden de ideas CSS nos va a apoyar para que a nuestro maquetado podamos cambiarle el color, el tamaño de la letra y el color de fondo.



Por ejemplo si quisiéramos realizar estas acciones tendríamos que aplicar las siguientes **propiedades**:

- color : green → ponerle color verde al texto
- font-size : 18px → poner el tamaño de la letra a 18 px
- background-color : yellow → poner el color del fondo de color amarillo

Entonces en el siguiente ejemplo, estaríamos otorgando a nuestro encabezado <h1> que se muestre de color azul y que tenga un tamaño de 20 px.

```
1 h1 {  
2     color: blue;  
3     font-size: 20px;  
4 }
```

Debemos cumplir unas reglas básicas para aplicar estilos a uno o varios elementos de nuestro producto web, la estructura que debemos utilizar es la siguiente:

1. Vamos a escribir el nombre de un **selector**, que es un texto que hace referencia a un elemento HTML, por ejemplo, h1 que hace referencia a la etiqueta h1.
2. Luego vamos a añadir un conjunto de llaves ({ }) para su apertura y cierre.
3. Dentro de estas llaves escribiremos los atributos CSS, estos atributos son un conjunto de reglas formadas **por una propiedad y el valor de esta propiedad**, separador por dos puntos : y terminando en punto y coma ; .

Estás reglas son las que van a definir los distintos estilos que vamos a aplicar al selector que hemos definido previamente, en la siguiente imagen podemos evidenciar de forma visual la especificación de estas tres reglas.





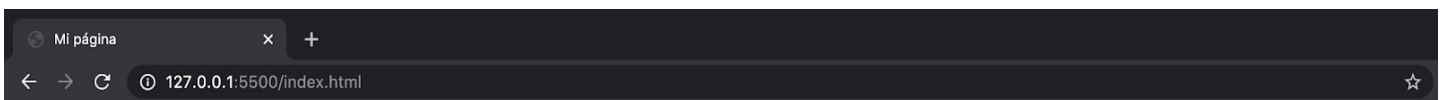
Como buena práctica vamos a escribir un espacio después de los dos puntos :, en cada atributo CSS para que sea más fácil leer nuestro código. Esta es una práctica muy común que usan los desarrolladores de software web y se incorpora dentro de los procesos de codificación en muchas empresas de programación.

Cuando vamos a crear un selector podemos usar:

- La etiqueta del elemento HTML por ejemplo, h1, h2, p, ul, section.
- Un atributo del elemento HTML, hay dos atributos especiales que suelen ser usados para esto: el id y la clase.
- Otros selectores más avanzados.

Dentro de nuestra página web esto se vería así:

```
1 <!doctype html>
2 <html lang="es">
3 <head>
4   <meta charset="utf-8">
5   <title>Mi página</title>
6   <style>
7     h1 {
8       color: blue;
9       font-family: Arial, sans-serif;
10      font-size: 20px;
11    }
12  </style>
13 </head>
14 <body>
15   <h1>Título de mi página</h1>
16   <p>Contenido de prueba de mi página web.</p>
17 </body>
18 </html>
```



Título de mi página

Contenido de prueba de mi página web.



Existen varias formas de añadir estilos a una página web, una de ellas es la que evidenciamos en el ejemplo anterior, y se materializa haciendo uso de la etiqueta `<style>` la cual aplica ajustes visuales sobre la página en este caso en nuestro elemento h1.

Ahora, esta práctica no es la más idónea o adecuada, desde hace unos años se ha buscado separar las responsabilidades de cada uno de los distintos lenguajes que incorporamos en nuestros productos web, así de esta forma cada uno se encarga de realizar su actividad puntual y nos permite tener un código más legible y organizado.

Entonces debemos separar el contenido de la presentación pasando nuestros estilos siempre a un archivo externo que estará enlazado o será llamado desde nuestra página HTML. Para poder invocar o enlazar el archivo CSS haremos uso de la etiqueta `<link>`, que es una de aquellas etiquetas que mencionamos que no necesitan cerrarse.

Esta etiqueta lleva dos atributos, uno que dice el tipo de archivo que va enlazado `rel="stylesheet"` y otra que dice dónde está el archivo `href="styles.css"`.

Reestructurando el ejemplo anterior tendríamos los siguientes dos archivos:

index.html

```
1 <!doctype html>
2 <html lang="es">
3 <head>
4   <meta charset="utf-8">
5   <title>Mi página</title>
6   <link rel="stylesheet" href="styles.css">
7 </head>
8 <body>
9   <h1>Título de mi página</h1>
10  <p>Contenido de prueba de mi página web.</p>
11 </body>
12 </html>
```

styles.css

```
1 h1 {
2   color: blue;
3   font-family: Arial, sans-serif;
4   font-size: 20px;
5 }
```



Taller - Ejercicios Prácticos

1. Vamos a crear una página web que tenga la información correspondiente a una receta de una bandeja paisa.

Bandeja Paisa

La bandeja paisa es, sin duda, uno de los platos más reconocidos de la gastronomía colombiana. Es tradicional de las tierras antioqueñas, pero se consume en todo el país e incluso en el extranjero, donde ha llegado de la mano de migrantes colombianos.

¿Qué ingredientes se necesitan para hacer una bandeja paisa para 6 personas?

- 3 tazas de frijoles de cargamanto o rojos
- 2 libras de carne molida
- 2 libras de chicharrón
- 2 libras de carne pulpa molida
- 6 chorizos
- 6 huevos
- Sal al gusto
- 1 libra de arroz
- 3 Plátanos maduros
- 4 cebollas grandes
- 4 tomates
- 1 zanahoria mediana
- Medio plátano verde
- Aceite

El éxito de la bandeja paisa está en la preparación de los frijoles

Para que tus frijoles sean más fáciles de cocinar, debes ponerlos a remojar en un recipiente con agua. Déjalos toda la noche para que estén más blandos a la hora de ponerlos a cocinar.

Prepara el hogao picando finamente las cebollas y los tomates. En una sartén con un poco de mantequilla y ajo al gusto, ponlos a sofreír durante diez minutos.

Para la cocción de los frijoles utiliza la olla exprés, esto te ahorrará un poco de tiempo. Agregar y cubrir con agua y échales la mitad del hogao, medio plátano verde y la zanahoria mediana partida en trozos. Déjalos pitar aproximadamente por media hora. Después que hayan pitado, sácales la zanahoria, el plátano y una taza de agua y licuar. Agrega esta mezcla a los frijoles nuevamente y ponlos a fuego medio –bajo a espesar.



POLITÉCNICO COLOMBIANO
JAIME ISAZA CADAVID

Educación para
Vivir mejor

2. Añadir hoja de estilos al ejercicio anterior donde:

- El color de fondo de la página tiene que ser **#f3f4f5**
- El título tiene que tener asignado un tipo de letra **Arial a 24px** de tamaño y color de texto **black**.
- Para el texto de los párrafos, tendremos que usar la fuente **Georgia a 18px** de tamaño y color de texto **#757575**.



Politécnico Colombiano Jaime Isaza Cadavid

@PolitecnicoJIC