

## **Tema 1: Desarrollo o programación web con HTML**

En este primer módulo empezaremos por conocer los conceptos básicos de HTML5 para que te animes a realizar tu primera página Web. Luego iremos aprendiendo cómo agregar estilos para hacer una página más atractiva para el usuario.

### **¡Arrancamos!!!**

A través de varios capítulos vamos a descubrir el principal lenguaje utilizado para la creación de páginas Web: Hyper Text Markup Language (lenguaje de marcas de Hipertexto), más conocido mediante sus siglas HTML

Puede que, en un principio, el hecho de hablar de un lenguaje informático te ponga un poco nervioso, sin embargo, el HTML no deja de ser más que una forma sencilla de especificar el contenido de las páginas, indicando el texto y otros elementos como imágenes, tablas, listas, etc.

Al final es de suma importancia el lenguaje porque es el medio con el cual se suministra el contenido a los navegadores y, por tanto, si queremos comenzar a aprender a crear páginas web, necesariamente debemos comenzar por aquí.

Este recorrido será teórico-práctico. Iremos conociendo el lenguaje a través de numerosos ejemplos que te sugerimos realizar por ti mismo para asimilar mejor los conocimientos. Verás también como HTML dentro del contexto de la creación de una página web se apoya en editores, programas para escribir las mismas.

### **HTML: historia, objetivos y otros lenguajes.**

HTML es el lenguaje con el que se escribe el contenido de las páginas web. Las páginas web pueden ser vistas por el usuario

mediante un tipo de aplicación llamada cliente Web o más comúnmente "navegador". Podemos decir por lo tanto que el HTML es el lenguaje usado para especificar el contenido que los navegadores deben representar a la hora de mostrar una página Web. Este lenguaje nos permite agrupar textos, imágenes, enlaces y combinarlos a nuestro gusto.

La ventaja del HTML a la hora de representar el contenido en un navegador, con respecto a otros formatos físicos como libros o revistas, es justamente la posibilidad de colocar referencias a otras páginas, por medio de los enlaces hipertexto.

## Historia.

El lenguaje HTML se creó en 1991. Tiene una historia realmente corta, pero para su poca vida ha sufrido importantes cambios. Su padre es Tim Berners-Lee que lo diseñó con objetivos divulgativos. Inicialmente no se pensó que la web llegaría a ser un área de ocio con carácter multimedia, de modo que, nació sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro.

Sin embargo, pese a esta deficiente planificación, sí que se han ido incorporando modificaciones con el tiempo, agregando nuevas características para cubrir las nuevas necesidades. Como hemos dicho, los programas que leen y presentan las páginas web a los usuarios se llaman navegadores. Éstos son los responsables de interpretar el HTML y "pintar" una página tal como ellos entiendan que deben hacer.

No obstante, existen reglas de representación que no son propias de cada fabricante del navegador, sino que existe una organización llamada W3C que se encarga de definir el estándar que todos deben seguir a la hora de escribir e interpretar el HTML.

Estos estándares del HTML se conocen como "Especificaciones", las cuales han ido apareciendo en el tiempo. El HTML5 es el último estándar en la actualidad. Históricamente los navegadores, además de la propia comunidad de usuarios, han sido los mayores impulsores de los cambios ocurridos en el lenguaje. Una vez detectada la necesidad es el W3C el que crea el estándar y marca una dirección que todos deben seguir.

## Los lenguajes de la web

HTML no está solo como único lenguaje para crear las páginas Web, aunque en un principio era así. Su evolución tan anárquica ha supuesto toda una serie de inconvenientes y deficiencias que han debido ser superados con la introducción de otras tecnologías accesorias capaces de organizar, optimizar y automatizar el funcionamiento de las webs. Ejemplos que pueden sonar como lo son las CSS, o JavaScript. Veremos más adelante en qué consisten estas tecnologías.

## Los editores de HTML

Además del navegador necesario para ver los resultados de nuestro trabajo, necesitamos evidentemente otra herramienta capaz de crear la página Web. Un archivo HTML (una página) no es más que un texto plano (sin forma estética) al que le colocamos extensión ".html". Es por ello que para programar en HTML necesitamos un editor de texto. Es recomendable usar un editor de textos sencillo, de texto plano.

En la actualidad existen plataformas online que son una opción para quienes tienen limitaciones de hardware, usan computadoras fuera de su entorno personal e incluso para personas que programan a través de dispositivos móviles.

# >Talentos Digitales\_

Entre esos IDE online tenemos CodePen, que ofrece un entorno de desarrollo y ejecución de código FrontEnd (HTML, CSS y Javascript), facilitando la creación de pequeños proyectos y pruebas de fragmentos de código, que se almacenan en el entorno de las herramientas de desarrollo denominadas como pen. Es ideal para principiantes en programación. Ofrece una versión paga y gratuita, acceso con o sin inicio de sesión, pero para que el código se almacene, se requiere el login.

Otro que funciona muy bien es CodeSandbox, un entorno de desarrollo para aplicaciones web al que se puede acceder con un navegador. No se requiere configuración, por lo que se puede comenzar a usar de inmediato. Los espacios aislados se pueden compartir fácilmente con equipos o colegas y se pueden crear todos los que se necesiten. Con la funcionalidad de actualizaciones en vivo, los códigos se ejecutan de inmediato, lo que permite ver los resultados directamente en el espacio aislado. Los espacios aislados se pueden vincular a GitHub.

Existen infinidad de editores de código interesantes, que nos aportan más o menos facilidades y que nos permiten aumentar nuestra productividad. No obstante, es aconsejable en un principio utilizar una herramienta lo más sencilla posible para poder prestar la máxima atención a nuestro código y familiarizarnos lo antes posible con él.

Igualmente, una persona que comience puede pensar que tienen muchas opciones, pero realmente podemos comenzar con lo básico, crear nuevos archivos, editar el código, y guardarlos en una carpeta de nuestro disco duro. Recomendamos estos editores porque están disponibles para todas las plataformas, Windows, Mac y Linux y porque son gratuitos para cualquier uso.

Aunque comenzaremos a usar HTML para definir cómo se puede ver un contenido, por ejemplo: si el texto debe tener color rojo, o el tamaño de la letra, o si se debe alinear a la derecha, es de destacar que actualmente para especificar el aspecto que debe tener una web se usa un lenguaje complementario, llamado CSS.

## HTML Iniciando con la sintaxis.

HTML es un "lenguaje de marcado". Basa su sintaxis en un elemento base al que llamamos marca, tag o etiqueta. A través de las etiquetas vamos definiendo los elementos del documento, como enlaces, párrafos, imágenes, etc. Así pues, un documento HTML estará constituido por texto y un conjunto de etiquetas para definir la función que juega cada contenido dentro de la página. Todo eso servirá al navegador para saber cómo se tendrá que presentar el texto y otros elementos en la página.

Existen etiquetas para crear negritas, párrafos, imágenes, tablas, listas, enlaces, etc. Así pues, aprender HTML es básicamente aprenderse una serie de etiquetas, sus funciones, sus usos y saber un poco sobre cómo debe de construirse un documento básico.

## Anatomía de una etiqueta HTML

La etiqueta presenta frecuentemente dos partes, su apertura y cierre, y se encierran ambas partes entre símbolos "menor que <" y "mayor que >".

### ESTRUCTURA DE UNA ETIQUETA HTML



## ¿Qué es una etiqueta HTML?

# >Talentos Digitales\_

En los navegadores, hay varias formas de acceder al código HTML de la página:

Pulsando la combinación de teclas CTRL+U. Te aparecerá el código fuente tal cual lo recibe el navegador.

Pulsando CTRL+SHIFT+i accediendo a la pestaña Elements, o simplemente haciendo click con botón derecho y seleccionando la opción Inspeccionar. Allí se muestra un panel más avanzado e interactivo.

Dicho documento está formado por etiquetas, que son la base del lenguaje HTML. Existen muchas etiquetas y cada una se utiliza para contener información y darle un cierto significado a dicha información, dependiendo de la etiqueta que se trate. Las etiquetas HTML tienen la siguiente estructura:

```
<etiqueta>contenido</etiqueta>
```

En HTML no se puede utilizar cualquier palabra como etiqueta (en el ejemplo anterior, es incorrecto utilizar la etiqueta «etiqueta»). En su lugar, existen una serie de etiquetas concretas, cada una de ellas con su finalidad y características propias. Por norma general, las etiquetas deben cerrarse para indicar donde finaliza su contenido. Si en nuestro documento HTML escribimos una frase con el siguiente código:

```
<b> Esto está en negrita </b>
```

Veremos que las palabras "Esto está en negrita" aparecen en negrita. Ya que el texto que encerramos entre una <b> que abre y otra que cierra </b> genera que la página muestre ese texto en negrita:

Otro ejemplo rápido. La etiqueta <p> define un párrafo. Si en nuestro documento HTML escribimos:

```
<p>Hola, estamos en el párrafo 1</p>
<p>Ahora hemos cambiado de párrafo</p>
```

Pág. 6



Como resultado obtendríamos dos párrafos con esos textos. En HTML los párrafos están separados por un doble salto de línea.

## Etiquetas:

A continuación, utilicemos la etiqueta `<strong>` (etiqueta utilizada para darle importancia a una palabra sobre el resto) en el siguiente ejemplo:

```
<p>Dentro de este texto, esta <strong>palabra</strong> es más importante que el resto. </p>
```

Obsérvese que en el ejemplo estamos utilizando dos etiquetas diferentes. La etiqueta `<p>`, que contiene toda la frase, y la etiqueta `<strong>`, que contiene sólo la palabra palabra. La primera etiqueta, `<p>`, se utiliza para contener párrafos, mientras que la segunda etiqueta, `<strong>`, indica la palabra que tiene mayor importancia del texto.

También se habrá observado que se pueden anidar etiquetas, esto es, incluir etiquetas dentro de otras. Es algo que se hace continuamente en HTML y que a medida que avancemos se tornará en algo habitual.

Gran parte de los elementos tiene etiqueta de apertura y de cierre, pero hay excepciones, como los elementos vacíos. Estos no requieren etiqueta de cierre porque carecen de contenido.

Por ejemplo:

```
<img src= " /" alt= "Imagen">
```

Esta etiqueta contiene un atributo `src` y uno `alt`, que es el texto descriptivo, pero no tiene contenido ni etiqueta final.

Todo atributo debe contener:

- Un espacio entre el atributo y el nombre del elemento

- El nombre del atributo y seguidamente el signo de igual (=).
- Comillas de apertura y de cierre

Lo anterior se debe a que una imagen no contiene información a la que afectar. Su objetivo es simplemente colocar o desplegar una imagen en la página HTML en el lugar en que se coloque.

## ¿Qué es la semántica?

Uno de los principales objetivos de HTML5 es introducir información en un documento HTML5 de forma que sea semántico y no visual. Con esto queremos decir que todos los aspectos visuales deben dejarse para el apartado de presentación, que se gestiona desde el lenguaje CSS.

En el documento HTML debe aparecer información correctamente individualizada, de modo que al leer una página HTML comprendamos su significado, y si queremos cambiar la apariencia, lo hagamos en el documento CSS. Esto es lo que comúnmente se conoce como separación de la presentación del contenido.

## Ejemplo de semántica HTML

Un ejemplo donde se ve claramente esto es con la etiqueta `<b>` de HTML4 y anteriores. Dicha etiqueta se utilizaba para poner en negrita (*bold*) un texto específico:

Hola, quiero resaltar esta `<b>palabra</b>`.

En este caso, se está utilizando una propiedad de presentación (visual) en el HTML, algo que no se debe hacer en HTML5. La misión de HTML5 es mantener sólo contenido e información semántica en HTML5. Por ello, la forma de hacerlo en HTML5 es la siguiente:

Hola, quiero resaltar esta `<strong>palabra</strong>`



En este nuevo ejemplo, se reemplaza la etiqueta `<b>` (*negrita, característica de presentación*) por `<strong>`, una etiqueta que indica información semántica. De esta forma, en el HTML5 sólo se está añadiendo información particular sobre fragmentos de texto, y si queremos dotar de presentación visual, lo haremos desde CSS:

```
strong {  
  font-weight: bold;  
  color: black;  
}
```

De esta forma podríamos cambiar el estilo (*negrita, colores, tipo de letra, etc...*) independientemente de la información y contenido que existe en el HTML. Incluso podríamos establecer el mismo estilo que en el texto, para que visualmente se vea todo de forma uniforme, pero, sin embargo, al leer la página HTML, el navegador sabrá siempre que fragmentos de texto son más importantes, aunque tenga el mismo estilo visual.

El objetivo de crear documentos HTML semánticos es que, aunque estamos acostumbrados a crear páginas para usuarios (o *más concretamente, para navegadores*), cada vez tendemos más a una Internet capaz de procesar información de forma autónoma. Muestra de ello son, por ejemplo, los robots o crawlers de buscadores como Googlebot (*el robot de búsqueda de Google*) que es capaz de acceder a páginas web para.

## Atributos HTML

En algunas etiquetas HTML, existen algunos atributos específicos (*que pueden ser opcionales u obligatorios*). Los atributos determinan cierta información sobre la etiqueta (*o su modo de actuar*) y generalmente van asociados a un valor determinado. Este par atributo-valor se escribe después del nombre de la

etiqueta, separándola por espacio y antes del carácter > de la etiqueta de apertura:

```
<strong id="dato">Contenido</strong>
```

En este caso, la etiqueta sería strong, el atributo id, el valor de id sería dato, mientras que el contenido de la etiqueta es Contenido. Por otro lado, y al igual que las etiquetas, cada atributo tiene una misión y comportamiento concreto. Aunque los valores pueden ir rodeados por comillas simples, se recomienda escribir el valor siempre entre comillas dobles.

Existen 3 tipos de atributos dependiendo de sus valores:

1. **Conjunto de valores:** Son aquellos atributos en los que sólo se permiten unos valores concretos. Cualquier otro valor diferente, no será válido.
2. **Valores libres:** Son los atributos en los que puedes especificar un valor libremente, como una dirección URL o un texto, y no existe una serie de valores específicos para escribir.
3. **Valores booleanos:** Son los atributos que deben tener un valor verdadero (*true*) o un valor falso (*false*). En HTML5 un atributo sin valor (*solo el atributo*) significa verdadero y si se omite el atributo se considera falso.

## Contenido de la etiqueta

En el interior de la etiqueta HTML (*después de la etiqueta de apertura y antes de la etiqueta de cierre*) se debe colocar la información que queremos que sea afectada por dicha etiqueta. En el siguiente ejemplo se ve como contiene un fragmento de texto:

```
<strong id="dato" class="clase1" lang="es" >Contenido texto</strong>
```

Ten en cuenta que una etiqueta puede tener varios pares atributo-valor, como se ve en el ejemplo anterior, pero nunca se debe repetir el mismo atributo en una misma etiqueta varias veces, ya que sobrescribiría al anterior. El orden de los atributos no importa. Más adelante explicaremos que tipos de atributos existen y para qué sirven.

Sin embargo, una etiqueta puede contener desde un fragmento de texto hasta un grupo de etiquetas. Esto depende mucho de la etiqueta a utilizar, y se verá más adelante en el apartado de etiquetas. Un posible ejemplo sería el siguiente:

```
<div id="pagina">  
<strong>Contenido importante</strong>  
</div>
```

Nótese que dentro de la etiqueta `<div>` vemos que no sólo hay un fragmento de texto, sino que además incluye otra etiqueta, `<strong>` en este caso. Esto ocurre en las etiquetas de agrupación, que veremos más adelante.

## Comentario HTML

Los comentarios son una práctica muy común y habitual en los desarrolladores o programadores. Se basa en introducir breves fragmentos de texto que el navegador ignora y no tendrá en cuenta a la hora de crear la página visualmente, pero que a nosotros nos sirven de ayuda para documentar algún detalle, explicar algo importante o simplemente introducir algún texto que consideramos relevante:

```
<!--Esto es un comentario de ejemplo que el navegador ignorará -->
```

Como se puede ver, para introducir estos comentarios en el código HTML, basta con escribir los fragmentos de texto `<!--` y `-->` entre el comentario en cuestión que queramos incluir.

## Atributos comunes en HTML

Como hemos visto, el HTML está formado por etiquetas. Muchas etiquetas. Cada etiqueta tiene una misión y tarea y contendrá cierto contenido relacionado con su misión. Y además tenemos los atributos, que son palabras clave de texto que modifican ligeramente el comportamiento de la etiqueta que lo contiene.

## Atributos de CSS

En este apartado vamos a ver una lista de atributos comunes que tienen cierta relación con CSS y que pueden ser utilizados en prácticamente cualquier etiqueta HTML:

Atributo	Valor	Descripción
<b>id</b>	<i>nombre</i>	Establece un <b>identificador único</b> a la etiqueta HTML. Sólo el mismo nombre una vez por documento.
<b>class</b>	<i>nombre</i>	Establece una <b>clase</b> (género) a una etiqueta HTML. Puede repetirse por documento.
<b>style</b>	<i>estilos CSS</i>	<i>Aplica propiedades CSS directamente al elemento HTML en cuestión.</i>

Veamos cada uno de ellos por separado para entenderlo bien.

## El atributo id (identificador)

En HTML, podemos darle un identificador a una etiqueta HTML y de esta forma darle un nombre. Simplemente, añadimos el atributo id y colocamos el nombre como valor de ese atributo.

Ese identificador único tiene unas normas:

- No debe empezar nunca por un número, pero puede contenerlos más adelante.
- El texto debe estar en kebab-case, es decir, minúsculas y separados por un guión.
- Es preferible que no contenga caracteres raros, acentuados o emojis.
- En un documento HTML no pueden existir dos elementos con el mismo id.

Veamos un ejemplo de código:

```
< div id="pagina">  
  <div>Aquí irá un anuncio</div>  
  <div id="articulo">Aquí irá el contenido de texto del  
  artículo</div>  
  <div>Aquí irá un anuncio</div>  
</div>
```

El detalle principal más importante de los id, es que no pueden existir dos con el mismo nombre en un mismo documento.

Ejemplos correctos de id serían:

- ✓ «pagina» (contiene el fragmento de código HTML que incluye todo el contenido de la página)
- ✓ «comentarios» (contiene toda la zona de comentarios de la página)

- ✓ «header» (contiene la parte con el logo y la cabecera de la página)

Lo importante que hay que saber de los id es que se suelen utilizar cuando queremos localizar zonas específicas que sabemos que no se van a repetir en esa misma página.

Sin embargo, mi consejo es que se utilicen clases salvo que se necesite expresamente un id, cosa que sólo suele ocurrir cuando necesitamos enlaces de tipo anclas. Más adelante, en el apartado de enlaces o hipervínculos, veremos las anclas, que sirven para acceder rápidamente a esa sección concreta de la página.

## El atributo class (clases)

Las clases funcionan de una forma muy similar a los id, pero son mucho más flexibles. Utilizaremos el atributo class, donde le asignaremos el nombre de clase. En primer lugar, no tienen la limitación de los id, por lo que pueden existir elementos con la misma clase en un documento. De hecho, la idea de las clases es establecer géneros o tipos de etiquetas, a los que les asociemos características comunes.

Sigamos con el ejemplo anterior:

```
<div id="pagina">
  <div class="anuncio">Aquí irá un anuncio</div>
  <div id="articulo">Aquí irá el contenido de texto del
artículo</div>
  <div class="anuncio">Aquí irá un anuncio</div>
</div>
```

Obsérvese que, al tener la misma clase «anuncio», podemos realizar acciones para todas las etiquetas de ese tipo y no tener que hacerlo para cada una de ellas por separado. Un ejemplo clásico donde se ve bien su utilidad, es respecto a utilizar id y



clases para dar estilo CSS, donde aplicaremos unos estilos concretos a todas las etiquetas HTML con clase «anuncio».

Además, a diferencia de los id, una etiqueta puede tener múltiples clases diferentes, no necesariamente una sola. Esto nos da más flexibilidad a la hora de crear clases específicas, donde cada una se ocupa de un detalle diferente:

```
<div id="pagina">  
  <div class="anuncio primero">Aquí irá un  
  anuncio</div>  
  <div id="artículo">Aquí irá el contenido de texto  
  del artículo</div>  
  <div class="anuncio ultimo">Aquí irá un
```

Nótese que en la primera etiqueta del anuncio hemos aplicado las clases anuncio y primero, mientras que en el último anuncio hemos aplicado las clases anuncio y último. Esto nos permitiría asignar atributos comunes al anuncio en la clase anuncio, y atributos que sólo dependan de la posición donde está colocado en primero y/o ultimo.

Recuerda que para indicar múltiples clases se deben separar las clases por espacio, dentro del mismo atributo class. Si se indican múltiples atributos class en la misma etiqueta, el navegador ignorará los primeros y sólo utilizará el valor del último class.

## El atributo style (Estilos en línea)

El atributo style es un atributo que se utiliza en las etiquetas HTML para incrustar código CSS directamente en la propia etiqueta. Este tipo de CSS se denomina Estilos CSS en línea.

Generalmente, la mejor forma de aplicar estilos CSS es colocar el código CSS en un documento .css independiente de nuestro documento HTML. Esto garantiza que nuestro código esté bien separado y modularizado. Sin embargo, hay situaciones donde

puede ser conveniente (o incluso necesario) añadir el código CSS en la propia etiqueta. En ese caso, se haría de la siguiente forma:

```
<p style="background: red; color: white;">Esto es un  
mensaje con estilo CSS</p>
```

Sin embargo, en la mayoría de los casos, no se recomienda añadir los estilos de esta forma, y suele ser considerado una mala práctica, ya que estás acoplando el código CSS al documento HTML, cuando deberían estar separados.

## Referencias Bibliográficas:

- <https://desarrolloweb.com/articulos/que-es-html.html>
- Qué es HTML y para qué sirve:  
<https://www.vadavo.com/blog/html-que-es-y-para-que-sirve/>
- ¿Qué es HTML? Explicación de los fundamentos del lenguaje de marcado: [https://www.hostinger.com.ar/tutoriales/que-es-html#Las\\_etiquetas\\_y\\_elementos\\_HTML\\_mas\\_utilizados](https://www.hostinger.com.ar/tutoriales/que-es-html#Las_etiquetas_y_elementos_HTML_mas_utilizados)
- Código HTML – todo lo que debes saber:  
<https://www.servnet.mx/blog/codigo-html-que-es-y-todo-lo-que-debes-saber>

¡¡Seguimos leyendo el material sobre  
el Tema 1: segunda parte!!