

## Objetivo General

El estudiantado aplicará los conocimientos básicos en lenguaje HTML y lenguaje CSS con la finalidad de realizar el desarrollo de su primera página web estática.

## Temario Desarrollo de paginas Web

### 1. Introducción

- A. Fundamentos técnicos de una página web.
- B. Tecnologías de software para el desarrollo de páginas web.

### 2. Herramientas para la creación de páginas web

- A. Editor de texto.
- B. Editores de texto en línea.

## 3. Estructura básica de un documento HTML

- A. Experiencia de Usuario.
- B. Interfaz de Usuario.

## 4. Lenguaje HTML

- A. Introducción a HTML.
- B. Atributos.
- C. Listas.
- D. Tablas.
- E. Enlaces.
- F. Imágenes.

G. Vídeos.

H. Formularios.

I. Atributos de Input.

J. Contenedores.

K. Etiqueta Meta.

## 5. Lenguaje CSS

A. Selectores.

B. Seudo Clases y Seudo Elementos.

C. Modelo de Cajas.

D. Uso de Float.



E. Estilos de Tabla.

F. Estilos de Lista.

Proyecto Tu primer página Web

## Introducción

Todo comenzó con una versión simplificada de un lenguaje de programación llamado *HTML (Hyper Text Markup Language)*. El lenguaje, junto con identificadores y protocolos de comunicación, se concibió con el propósito de ofrecer la base necesaria para la creación de la Web.

El propósito inicial del HTML era estructurar texto para poder compartir documentos entre ordenadores remotos. Con el transcurso del tiempo, la introducción de mejores sistemas y pantallas de color obligaron al lenguaje a evolucionar y poder así trabajar con otros medios además de texto, como imágenes y tipos de letras personalizados.



Esta expansión complicó el trabajo de los desarrolladores, a quienes les resultaba cada vez más difícil crear y mantener sitios web extensos usando solo HTML. El problema se resolvió con la incorporación de un nuevo lenguaje llamado CSS, el cual permite a los desarrolladores preparar el documento que se va a presentar en pantalla.

La asociación entre HTML y CSS simplificó el trabajo de los desarrolladores, pero la capacidad de estos lenguajes para responder al usuario o realizar tareas como la reproducción de vídeo o audio era aún muy limitada.



*La aparición de JavaScript*, permitía a los desarrolladores acceder al contenido del documento y modificarlo de forma dinámica, solicitar datos adicionales desde el servidor, procesar información y mostrar los resultados en la pantalla, convirtiendo los sitios web en pequeñas aplicaciones.

Con la combinación de HTML, CSS y JavaScript, las tecnologías requeridas para construir la Web de las que disfrutamos hoy en día estaban listas, pero todavía existía un problema que resolver.





Estos lenguajes habían sido desarrollados de forma independiente y, por lo tanto, seguían sus propios caminos, ajenos a los cambios presentados por los demás. La solución surgió con la definición de una nueva especificación llamada *HTML5*.

HTML5 unifica todas las tecnologías involucradas en el desarrollo web. A partir de ahora, HTML se encarga de definir la estructura del documento, CSS prepara esa estructura y su contenido para ser mostrado en pantalla, y JavaScript introduce la capacidad de procesamiento necesaria para construir aplicaciones web completamente funcionales.

La integración entre HTML, CSS y JavaScript bajo el amparo de HTML5 cambió la Web para siempre. De la noche a la mañana se crearon nuevas compañías basadas en aplicaciones web y mercados completos, lo que originó una era de oro para el desarrollo web.



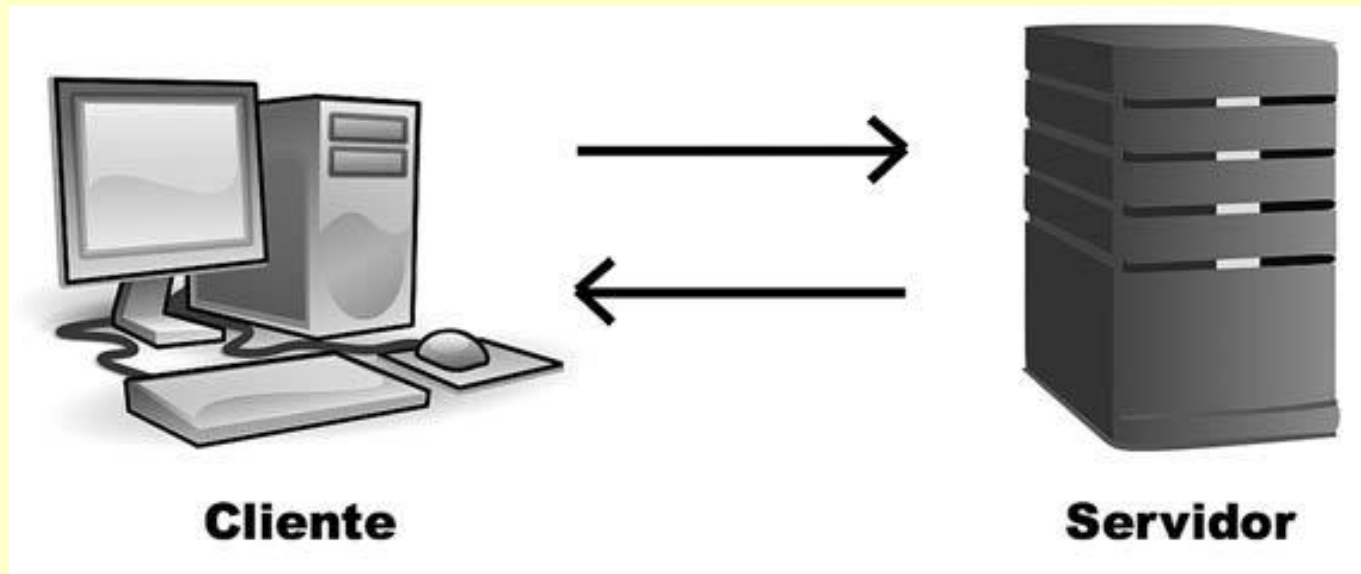
Implementando estas tecnologías, las oportunidades son infinitas. La Web está aquí para quedarse y tú puedes ser parte de ella.

Sitios Web (se basa en 3 conceptos. http, html y url)

Los sitios web son archivos que los usuarios descargan con sus navegadores desde ordenadores remotos. Cuando un usuario decide acceder a un sitio web, le comunica al navegador la dirección del sitio y el programa descarga los archivos, procesa su contenido y lo muestra en pantalla.

Debido a que los sitios webs son de acceso público e Internet es una red global, estos archivos deben estar siempre disponibles. Por este motivo, los sitios web no se almacenan en ordenadores personales, sino en ordenadores especializados diseñados para despachar estos archivos a los usuarios que los solicitan.

El ordenador que almacena los archivos y datos de un sitio web se llama *servidor* y el ordenador que accede a esta información se llama *cliente*, tal como lo ilustra la Figura 1-1.



***Figura 1-1: Clientes y servidores***



Los servidores son muy similares a los ordenadores personales, con la diferencia de que están continuamente conectados a la red y ejecutando programas que les permiten responder a las solicitudes de los usuarios, sin importar cuándo se reciben o de donde proceden. Los programas más populares para servidores son Apache, para sistemas Linux, e IIS (Internet Information Server), creado por Microsoft para sistemas Windows. Entre otras funciones, estos programas son responsables de establecer la conexión entre el cliente y el servidor, controlar el acceso de los usuarios, administrar los archivos, y despachar los documentos y recursos requeridos por los clientes.

Para desarrollar un sitio web, tenemos que crear un archivo por cada página que queremos incluir. Junto con estos archivos, también debemos incluir los archivos con las imágenes y cualquier otro recurso que queremos mostrar dentro de estas páginas (las imágenes y otros medios gráficos se almacenan en archivos aparte). En la Figura 1-2 se representa cómo se muestran los directorios y archivos de un sitio web una vez que se suben al servidor.



*Figura 1-2: Archivos de un sitio web*

El archivo index.html contiene el código y la información correspondiente a la página principal (la página que el usuario ve cuando entra a nuestro sitio web por primera vez), el archivo contacto.html contiene el código necesario para presentar un formulario que el usuario puede rellenar para enviarnos un mensaje y el archivo noticias.html contiene el código necesario para mostrar las noticias que queremos compartir con nuestros usuarios.

Los servidores son flexibles en cuanto a los nombres que podemos asignar a nuestros archivos, pero existen reglas que debería seguir para asegurarse de que sus archivos son accesibles. Evite usar espacios. Si necesita separar palabras use el guion bajo en su lugar (\_).

## Dominios y URL

Los servidores se identifican con un valor llamado *IP* (Internet Protocol). Esta IP es única para cada ordenador y, por lo tanto, trabaja como una dirección que permite ubicar a un ordenador dentro de una red. Cuando el navegador tiene que acceder a un servidor para descargar el documento solicitado por el usuario, primero busca el servidor a través de esta dirección IP y luego le pide que le envíe el documento.



Las direcciones IP están compuestas por números enteros separados por un punto, o números y letras separadas por dos puntos, dependiendo de la versión (IPv4 o IPv6).

Por ejemplo, la dirección **142.250.113.138** corresponde al servidor donde se encuentra alojado el sitio web de Google. Si escribimos esta dirección IP en la barra de navegación de nuestro navegador, la página inicial de Google se descarga y muestra en pantalla.





En teoría, podríamos acceder a cualquier servidor utilizando su dirección IP (Es el identificador numérico de una página web, es único y representa la dirección dónde está el ordenador que contiene esa página web), pero estos valores son difíciles de recordar. Por esta razón, Internet utiliza un sistema que identifica a cada servidor con un nombre específico. Estos nombres personalizados, llamados *dominios*, son identificadores sencillos que cualquier persona puede recordar, como *google* o *yahoo*, con una extensión que determina el propósito del sitio web al que hacen referencia, como *.com* (comercial) o *.org* (organización).

Los sitios web están compuestos por múltiples archivos, por lo que debemos agregar el nombre del archivo al dominio para indicar cuál queremos descargar. Esta construcción se llama *URL (Uniform Resource Locator)* e incluye tres partes, tal como se describe en la Figura 1-3.



*Figura 1-3: URL*



La primera parte de la URL es una cadena de caracteres que representa el protocolo de comunicación que se utilizará para acceder al recurso (el protocolo creado para la Web se llama *HTTP* (HyperText Transfer Protocol)), el siguiente componente es el dominio del sitio web y el último componente es el nombre del recurso que queremos descargar.

Existe otra versión disponible de este protocolo llamado HTTPS. La S indica que la conexión es encriptada por protocolos de encriptación como TLS o SSL. Los sitios web pequeños no necesitan encriptación, pero se recomienda utilizarla en sitios web que manejan información sensible.

La URL en nuestro ejemplo le pide al navegador que utilice el protocolo HTTP para acceder al archivo contacto.html, ubicado en el servidor identificado con el domino [www.ejemplo.com](http://www.ejemplo.com)

**Hipervínculos**, también llamados *enlaces en inglés links*, son referencias a documentos dentro de las páginas de un sitio web. Incorporando estos enlaces, una página puede contener referencias a otras páginas. Si el usuario hace click con el ratón en un enlace, el navegador sigue esa referencia y el documento indicado por la URL de la referencia se descarga y muestra en pantalla.



Debido a estas conexiones entre páginas, los usuarios pueden navegar en el sitio web y acceder a todos sus documentos simplemente haciendo clic en sus enlaces.

Los enlaces son lo que transforma a un grupo de archivos en un sitio web. Para crear un sitio web, debe programar los documentos correspondientes a cada página e incluir dentro de las mismas los enlaces que establecen una ruta que el usuario puede seguir para acceder a cada una de ellas.

## URL absolutas y relativas

Las URL absolutas son aquellas que incluyen toda la información necesaria para acceder al recurso (ver Figura 1-3), mientras que las relativas son aquellas que solo declaran la parte de la ruta que el navegador tiene que agregar a la URL actual para acceder al recurso. Por ejemplo, si tenemos un hipervínculo dentro de un documento que referencia una imagen dentro del directorio imagenes, podemos crear el enlace con la URL `http://www.ejemplo.com/imagenes/miimagen.png`, pero también tenemos la opción de declararla como `"imagenes/miimagen.png"` y el navegador se encargará de agregar a esta ruta la URL actual y descargar la imagen.



Las URL relativas no solo pueden determinar una ruta hacia abajo, sino también hacia arriba de la jerarquía. Por ejemplo, si tenemos un documento dentro del directorio recursos en el ejemplo de la Figura 1-2 y queremos acceder a un documento en el directorio raíz, podemos crear una URL relativa usando los caracteres ../ al comienzo de la ruta. Si el documento que queremos acceder es noticias.html, la URL relativa sería ../noticias.html. Los dos puntos .. le indican al navegador que el documento al que queremos acceder se encuentra dentro del directorio padre del actual directorio.



## Historia de HTML

- 1989 – Inicio de su desarrollo
- 1991 – Lanzamiento de la web
- 1992 – Lanzamiento de HTML (creado por Tim Bernes Lee)
- 1994 – Creación de la W3C (Consortio que define los estándares de la web)
- 1998 - HTML 4
- 1999 - HTML 4.1
- 2004 - Creación de la WHATWG
- 2008 – HTML 5
- 2014 – Estándar HTML 5 (Lanzado por W3C de forma oficial)



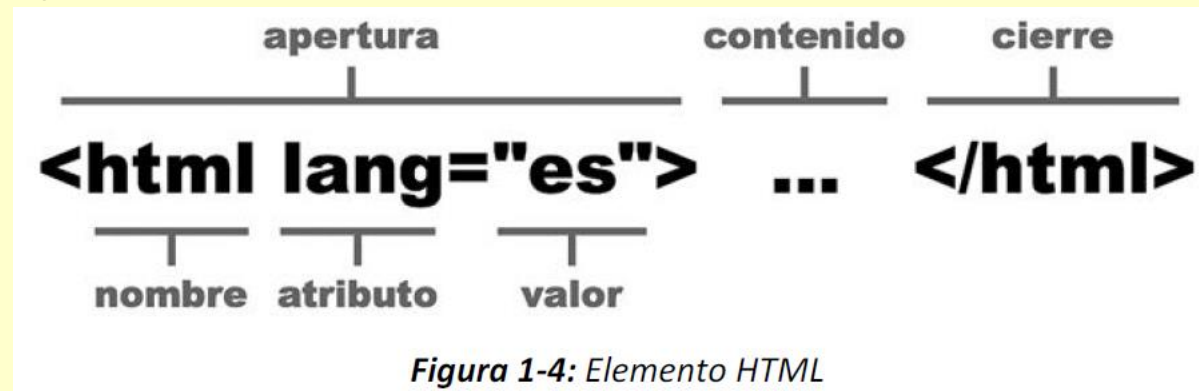
## Lenguaje

HTML5 incorpora tres características (estructura, estilo, y funcionalidad), con lo que integra tres lenguajes de programación independientes: HTML, CSS, y JavaScript. Estos lenguajes están compuestos por grupos de instrucciones que los navegadores pueden interpretar para procesar y mostrar los documentos al usuario. Para crear nuestros documentos, tenemos que aprender todas las instrucciones incluidas en estos lenguajes y saber cómo organizarlas.



**HTML** (*HyperText Markup Language*) Lenguaje de Marcado de Hipertexto, es un lenguaje compuesto por un grupo de etiquetas definidas con un nombre rodeado de paréntesis angulares. Los paréntesis angulares delimitan la etiqueta y el nombre define el tipo de contenido que representa. Por ejemplo, la etiqueta **<html>** indica que el contenido es código HTML. Algunas de estas etiquetas son declaradas individualmente (por ejemplo, **<br>**) y otras son declaradas en pares, que incluyen una de apertura y otra de cierre, como **<html></html>** (en la etiqueta de cierre el nombre va precedido por una barra invertida).

Las etiquetas individuales y las de apertura pueden incluir atributos para ofrecer información adicional acerca de sus contenidos (por ejemplo, `<html lang="es">`). Las etiquetas individuales y la combinación de etiquetas de apertura y cierre se llaman *elementos*. Los elementos compuestos por una sola etiqueta se usan para modificar el contenido que los rodea o incluir recursos externos, mientras que los elementos que incluyen etiquetas de apertura y cierre se utilizan para delimitar el contenido del documento, tal como ilustra la Figura 1-4.





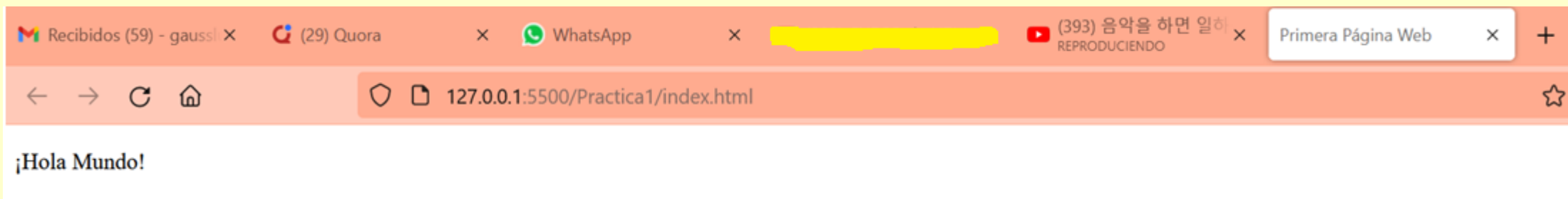
El elemento `<html>` que se muestra en la Figura 1-4 declara que su contenido debe ser interpretado como código HTML. Por lo tanto, el resto de los elementos que describen el contenido de ese documento se deben declarar entre las etiquetas `<html>` y `</html>`. A su vez, los elementos dentro del elemento `<html>` pueden incluir otros elementos. El siguiente ejemplo muestra un documento HTML sencillo que incluye todos los elementos necesarios para definir una estructura básica y mostrar el mensaje HOLA MUNDO! en la pantalla.

# Desarrollo Web



GOBIERNO DE LA  
CIUDAD DE MÉXICO

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Primera Página Web</title>
8 </head>
9 <body>
10   <p>¡Hola Mundo!</p>
11 </body>
12 </html>
```



En la primera línea, se encuentra una etiqueta individual que declara el tipo de documento (**<!DOCTYPE html>**) seguida por una etiqueta de apertura **<html lang="en">**. Entre las etiquetas **<html>** y **</html>** se incluyen otros elementos que representan la cabecera y el cuerpo del documento (**<head>** y **<body>**), los cuales a su vez encierran más elementos con sus respectivos contenidos (**<title>** y **<p>**), demostrando cómo se compone un documento HTML.

En general, todo elemento puede ser anidado, convertirse en un contenedor o ser contenido por otros elementos. Los elementos exclusivamente estructurales como **<html>**, **<head>** y **<body>** tienen un lugar específico en un documento HTML, pero el resto son flexibles



Las etiquetas individuales y de apertura pueden incluir atributos. Por ejemplo, la etiqueta de apertura **<html>** del ejemplo no está solo compuesta por el nombre **html** y los paréntesis angulares, sino también por el texto **lang="en"**. Este es un atributo con un valor. El nombre del atributo es **lang** y el valor es se asigna al atributo usando el carácter **=**. Los atributos ofrecen información adicional acerca del elemento y su contenido. En este caso, el atributo **lang** declara el idioma del contenido del documento (es por English).



CSS (*Cascading Style Sheets*) . Es el lenguaje que se utiliza para definir los estilos de los elementos HTML, como el tamaño, el color, el fondo, el borde, etc. Aunque todos los navegadores asignan estilos por defecto a la mayoría de los elementos, estos estilos generalmente están lejos de lo que queremos para nuestros sitios web. Para declarar estilos personalizados, CSS utiliza propiedades y valores. Esta construcción se llama *declaración* y su sintaxis incluye dos puntos después del nombre de la propiedad, y un punto y coma al final para cerrar la línea.



**color: #FF0000;**

propiedad                      valor

*Figura 1-5: Propiedad CSS*

En el ejemplo de la Figura 1-5, el valor #FF0000 se asigna a la propiedad color. Si esta propiedad se aplica luego a un elemento HTML, el contenido de ese elemento se mostrará en color rojo (el valor #FF0000 representa el color rojo). Las propiedades CSS se pueden agrupar usando llaves. Un grupo de una o más propiedades se llama *regla* y se identifica por un nombre llamado *selector*.

```
body {  
  width: 100%;  
  margin: 0px;  
  background-color: #FF0000;  
}
```

**Listado 1-2:** Declarando reglas CSS



El Listado 1-2 declara una regla con tres propiedades: width, margin y backgroundcolor. Esta regla se identifica con el nombre body, lo que significa que las propiedades serán aplicadas al elemento <body>. Si incluimos esta regla en un documento, el contenido del documento se extenderán hacia los límites de la ventana del navegador y tendrán un fondo rojo.

**Editores.** Los documentos HTML, así como los archivos CSS y JavaScript, son archivos de texto, por lo que podemos usar cualquier editor incluido en nuestro ordenador para crearlos, como el bloc de notas de Windows o la aplicación editor de texto de los ordenadores de Apple, pero también existen editores de texto especialmente diseñados para programadores y desarrolladores web que pueden simplificar nuestro trabajo.

Estos editores resaltan texto con diferentes colores para ayudarnos a identificar cada parte del código, o listan los archivos de un proyecto en un panel lateral para ayudarnos a trabajar con múltiples archivos al mismo tiempo.



La siguiente es una lista de los editores y de IDE (Integrated Development Environments) más populares disponibles para ordenadores personales y ordenadores de Apple.

- **Atom** ([www.atom.io](http://www.atom.io)) es un editor gratuito, simple de usar y altamente personalizable (recomendado).
- **Brackets** ([www.brackets.io](http://www.brackets.io)) es un editor gratuito creado por Adobe.
- **KompoZer** ([www.kompozer.net](http://www.kompozer.net)) es un editor gratuito con un panel de vista previa que facilita la búsqueda de partes específicas del documento a modificar.

- **Aptana** ([www.aptana.com](http://www.aptana.com)) es una IDE gratuita con herramientas que simplifican la administración de archivos y proyectos.
- **NetBeans** ([www.netbeans.org](http://www.netbeans.org)) es una IDE gratuita con herramientas para administrar archivos y proyectos.
- **Sublime** ([www.sublimetext.com](http://www.sublimetext.com)) es un editor de pago con una versión gratuita de evaluación.
- **Komodo** ([www.komodoide.com](http://www.komodoide.com)) es una IDE de pago que puede trabajar con una cantidad extensa de lenguajes de programación.
- **Dreamweaver** ([www.adobe.com/products/dreamweaver.html](http://www.adobe.com/products/dreamweaver.html)) es una IDE de pago con tecnología WYSIWYG incorporada (Lo Que Ves Es Lo Que Obtienes) que nos permite ver los resultados de la ejecución del código en tiempo real.



**Registro de dominios.** Una vez que nuestro sitio web está listo para ser presentado en público, tenemos que registrar el dominio que los usuarios van a escribir en la barra de navegación para acceder a él. Como ya mencionamos, un domino es simplemente un nombre personalizado con una extensión que determina el propósito del sitio web. El nombre puede ser cualquiera que deseemos, y contamos con varias opciones para definir la extensión, desde extensiones con propósitos comerciales, como .com, a aquellas sin ánimo de lucro o personales, como .org, .net o .info, por no mencionar las extensiones regionales que incluyen un valor adicional para determinar la ubicación del sitio web, como .co.uk para sitios web en el Reino Unido o .mx para sitios web relacionados con México.



Para obtener un dominio para nuestro sitio web, tenemos que abrir una cuenta con un registrante y adquirirlo. La mayoría de los dominios requieren del pago de un arancel anual, pero el proceso es relativamente sencillo y hay muchas compañías disponibles que pueden hacerse cargo del trámite por nosotros. La más popular es GoDaddy ([www.godaddy.com](http://www.godaddy.com)), pero la mayoría de las compañías que ofrecen servicios para desarrolladores también incluyen la posibilidad de registrar un dominio.

La compañía que registra su dominio asigna nombres de servidor por defecto que ellos usan como destino provisional (también conocido como *aparcamiento* o *parking*).



## Estructura

A pesar de las innovaciones introducidas por CSS y JavaScript en estos últimos años, la estructura creada por el código HTML sigue siendo la parte fundamental del documento. Esta estructura define el espacio dentro del documento donde el contenido estático y dinámico es posicionado y es la plataforma básica para toda aplicación.

Para crear un sitio o una aplicación web, lo primero que debemos hacer es programar el código HTML que define la estructura de cada una de las páginas que lo componen.

## Tipo de documento

Debido a que los navegadores son capaces de procesar diferentes tipos de archivos, lo primero que debemos hacer en la construcción de un documento HTML es indicar su tipo. Para asegurarnos de que el contenido de nuestros documentos sea interpretado correctamente como código HTML, debemos agregar la declaración **<!DOCTYPE>** al comienzo del archivo. Esta declaración, similar en formato a las etiquetas HTML, se requiere al comienzo de cada documento para ayudar al navegador a decidir cómo debe generar la página web. Para documentos programados con HTML5, la declaración debe incluir el atributo html, según la definimos en el siguiente ejemplo.

```
<!DOCTYPE html>
```

La línea con la declaración `<!DOCTYPE>` debe ser la primera línea de su documento, sin ningún espacio o código previo. Esto activa el modo estándar y obliga a los navegadores a interpretar HTML5 cuando es posible o ignorarlo en caso contrario.

## Elementos estructurales

Los elementos HTML conforman una estructura de tipo árbol con el elemento `<html>` como su raíz. Los siguientes son los elementos disponibles para definir la columna vertebral de la estructura y facilitar la información que el navegador necesita para mostrar la página en la pantalla.

**<html>**—Este elemento delimita el código HTML. Puede incluir el atributo lang para definir el idioma del contenido del documento.

**<head>**—Este elemento se usa para definir la información necesaria para configurar la página web, como el título, el tipo de codificación de caracteres y los archivos externos requeridos por el documento.

**<body>**—Este elemento delimita el contenido del documento (la parte visible de la página).

Existen varios valores disponibles para el atributo lang, incluidos en para inglés, es para español, fr para francés, entre otros.

```
<!DOCTYPE html>  
<html lang="es">  
  
</html>
```

*Listado 2-2: Incluyendo el elemento <html>*

El código HTML insertado entre las etiquetas <html> se tiene que dividir en dos secciones principales: la cabecera y el cuerpo. Por supuesto, la cabecera va primero y, al igual que el resto de los elementos estructurales, está compuesta por etiquetas de apertura y cierre.

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
  
</head>  
</html>
```

---

*Listado 2-3: Incluyendo el elemento <head>*

Entre las etiquetas **<head>** debemos definir el título de la página web, declarar el tipo de codificación de caracteres, facilitar información general acerca del documento, e incorporar los archivos externos con estilos y códigos necesarios para generar la página. Excepto por el título e iconos, el resto de la información insertada en medio de estas etiquetas no es visible para el usuario. La otra sección que forma parte de la organización principal de un documento HTML es el cuerpo, la parte visible del documento que se especifica con el elemento **<body>**.

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
  
</head>  
<body>  
  
</body>  
</html>
```

*Listado 2-4: Incluyendo el elemento <body>*

La estructura HTML puede describirse como un árbol, con el elemento **<html>** como su raíz, pero otra forma de definir la relación entre los elementos es describirlos como padres, hijos o hermanos, de acuerdo a sus posiciones en la estructura. Por ejemplo, en un documento HTML típico, el elemento **<body>** es hijo del elemento **<html>** y hermano del elemento **<head>**. Ambos, **<body>** y **<head>**, tienen al elemento **<html>** como su padre.

La estructura básica ya está lista. Ahora tenemos que construir la página, comenzando por la definición de la cabecera. La cabecera incluye toda la información y los recursos necesarios para generar la página. Los siguientes son los elementos disponibles para este propósito.



**<title>**—Este elemento define el título de la página.

**<meta>**—Este elemento representa metadatos asociados con el documento, como la descripción del documento, palabras claves, el tipo de codificación de caracteres, etc. El elemento puede incluir los atributos name para describir el tipo de metadata, content para especificar el valor, y charset para declarar el tipo de codificación de caracteres a utilizar para procesar el contenido.

**<style>**—Este elemento se usa para declarar estilos CSS dentro del documento.

**<script>**—Este elemento se usa para cargar o declarar código JavaScript.

**<link>**—Este elemento especifica la relación entre el documento y un recurso externo (generalmente usado para cargar archivos CSS). El elemento puede incluir los atributos **href** para declarar la ubicación del recurso, **rel** para definir el tipo de relación, **media** para especificar el medio al que el recurso está asociado (pantalla, impresora, etc.), y **type** y **sizes** para declarar el tipo de recurso y su tamaño (usado a menudo para cargar iconos).

Lo primero que tenemos que hacer cuando declaramos la cabecera del documento es especificar el título de la página con el elemento **<title>**.

Este texto es el que muestran los navegadores en la parte superior de la ventana, y es lo que los usuarios ven cuando buscan información en nuestro sitio web por medio de motores de búsqueda como Google.

```
Practica1 > index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Primera Página Web</title>
8  </head>
9  <body>
10     <p>¡Hola Mundo!</p>
11 </body>
12 </html>
```

Además del título, también tenemos que declarar los metadatos del documento. Los metadatos incluyen información acerca de la página que los navegadores y los motores de búsqueda utilizan para generar y clasificar la página web. Los valores se declaran con el elemento **<meta>**. Este elemento incluye varios atributos. El valor más importante es el que define la tabla de caracteres a utilizar para presentar el texto en pantalla, el cual se declara con el atributo **charset**.

Se define el grupo de caracteres como **utf-8**, que es el que se recomienda debido a que incluye todos los caracteres utilizados en la mayoría de los idiomas, pero existen otros disponibles.

Se pueden incluir múltiples elementos **<meta>** para declarar información adicional. Por ejemplo, dos datos que los navegadores pueden considerar a la hora de procesar nuestros documentos son la descripción de la página y las palabras claves que identifican su contenido.

Estos elementos **<meta>** requieren el atributo **name** con los valores "description" y "keywords", y el atributo **content** con el texto que queremos asignar como descripción y palabras clave (las palabras clave se deben separar por comas).

Otro elemento importante de la cabecera del documento es **<link>**. Este elemento se usa para incorporar al documento estilos, códigos, imágenes o iconos desde archivos externos. Para cargar este icono, tenemos que incluir un elemento **<link>** con el atributo **rel** definido como icon, el atributo **href** con la ubicación del archivo que contiene el icono, el atributo **type** para especificar el formato con el que se ha creado el icono, y el atributo **sizes** con el ancho y la altura del icono separados por la letra x.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="estilos.css">
9 </head>
10 <body>
11   <h1 style="color: palevioletred;">Lorem ipsum dolor sit.</h1>
12 </body>
13 </html>
```

```
<link rel="icon" href="imagenes/favicon.png" type="image/png"
sizes="16x16">
```

El valor asignado al atributo **type** del elemento **<link>** debe ser especificado como un tipo MIME. Todo archivo tiene un tipo MIME asociado para indicar al sistema el formato de su contenido. Por ejemplo, el tipo MIME de un archivo HTML es **text/html**. Existe un tipo MIME para cada tipo de archivo disponible, que incluye **image/jpeg** e **image/png** para imágenes JPEG y PNG.

El elemento **<link>** se usa comúnmente para cargar archivos CSS con los estilos necesarios para generar la página web. En este caso, solo necesitamos incluir el atributo **rel** para declarar el tipo de recurso (para hojas de estilo CSS debemos asignar el valor "**stylesheet**") y el atributo **href** con la URL que determina la ubicación del archivo.

Con la cabecera lista, es hora de construir el cuerpo. Esta estructura (el código entre las etiquetas **<body>**) es la encargada de generar la parte visible de nuestro documento (la página web). La siguiente lista incluye todos los elementos disponibles en HTML5 para definir la estructura del cuerpo.

**<div>**—Este elemento define una división genérica. Se usa cuando no se puede aplicar ningún otro elemento.

**<main>**—Este elemento define una división que contiene el contenido principal del documento (el contenido que representa el tema central de la página).



**<nav>**—Este elemento define una división que contiene ayuda para la navegación, como el menú principal de la página o bloques de enlaces necesarios para navegar en el sitio web.

**<section>**—Este elemento define una sección genérica. Se usa frecuentemente para separar contenido temático, o para generar columnas o bloques que ayudan a organizar el contenido principal.

**<aside>**—Este elemento define una división que contiene información relacionada con el contenido principal pero que no es parte del mismo, como referencias a artículos o enlaces que apuntan a publicaciones anteriores.

**<article>**—Este elemento representa un artículo independiente, como un mensaje de foro, el artículo de una revista, una entrada de un blog, un comentario, etc.

**<header>**—Este elemento define la cabecera del cuerpo o de secciones dentro del cuerpo.

**<footer>**—Este elemento define el pie del cuerpo o de secciones dentro del cuerpo.

Estos elementos han sido definidos con el propósito de representar secciones específicas de una página web. Aunque son flexibles y se pueden implementar en diferentes partes del diseño, todos siguen un patrón que se encuentra comúnmente en la mayoría de los sitios web. La Figura 2-2, a continuación, ilustra este tipo de diseño.

*Figura 2-2: Representación visual de un diseño web tradicional*





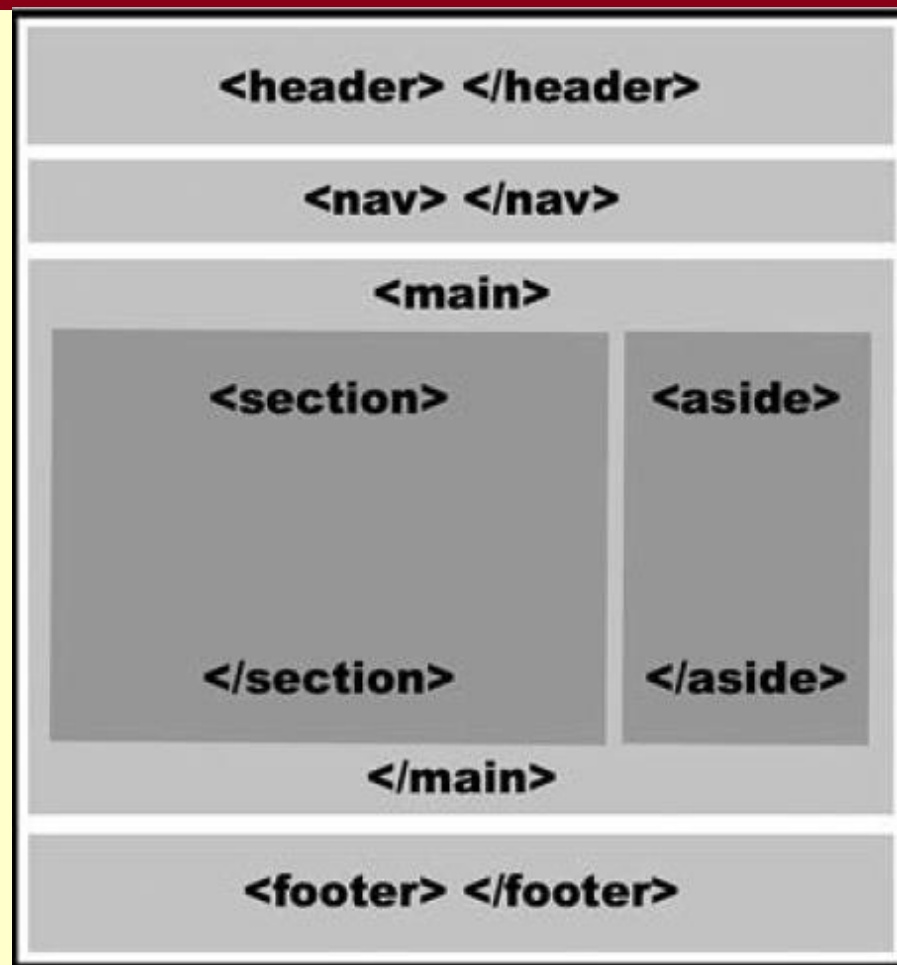
En la barra superior, descrita como **cabecera** en la Figura 2-2, ubicamos el logo, el nombre del sitio, los subtítulos y una descripción breve de nuestro sitio o página web.

En la **barra de navegación** situada debajo es donde la mayoría de los desarrolladores ofrecen un menú o una lista de enlaces para navegar en el sitio.

En la parte inferior de un diseño tradicional, nos encontramos con otra barra llamada **barra institucional**. La llamamos de este modo porque en este área es donde mostramos información general acerca del sitio web, el autor, la compañía, los enlaces relacionados con reglas de uso, términos y condiciones, el mapa del sitio, etc.

El contenido relevante de la página se ubica en el medio del diseño, donde generalmente encontramos artículos o noticias, y también enlaces a documentos relacionados o recursos. En el ejemplo de la Figura 2-2, esta sección se ha dividido en dos columnas, **información principal** y **barra lateral**, pero los diseñadores la adaptan a sus necesidades insertando columnas adicionales o dividiendo las columnas en bloques más pequeños.

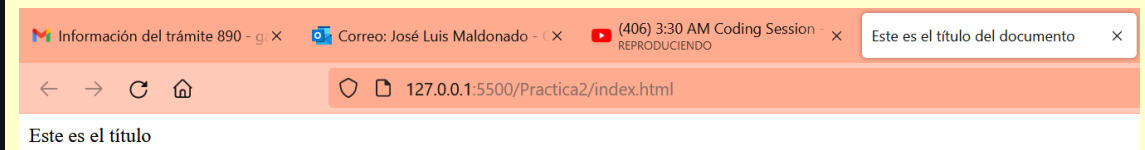
En la Figura 2-3 aplicamos los elementos introducidos anteriormente para definir el diseño de la Figura 2-2.



*Figura 2-3: Representación de la estructura de un documento usando elementos de HTML5*

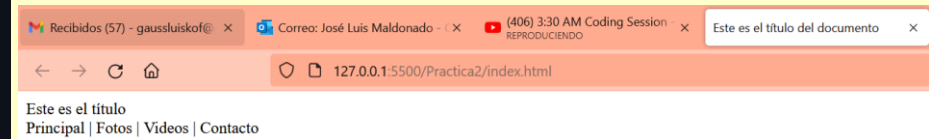
El primer elemento de un diseño tradicional es **<header>**. No debemos confundir este elemento con el elemento **<head>**. Al igual que **<head>**, el elemento **<header>** se ha definido para facilitar información introductoria, como títulos o subtítulos, pero no para el documento, sino para el cuerpo o secciones dentro del cuerpo del documento. En el siguiente ejemplo, este elemento se usa para definir el título de la página web.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Este es el título del documento</title>
8 </head>
9 <body>
10 <header>
11   Este es el título
12 </header>
13 </body>
14 </html>
```



La siguiente sección de nuestro ejemplo es la **barra de navegación**. Esta barra define una sección con ayuda para la navegación y se representa con el elemento `<nav>`.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Este es el título del documento</title>
8  </head>
9  <body>
10 <header>
11   Este es el título
12 </header>
13
14 <nav>
15   Principal | Fotos | Videos | Contacto
16 </nav>
17 </body>
18 </html>
```



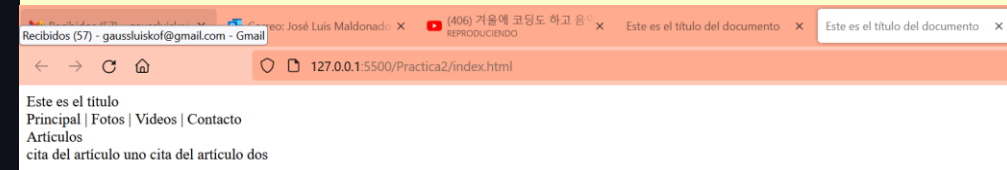


La estructura y el orden que decidimos implementar depende de lo que nuestro sitio web o aplicación requieran. Los elementos HTML son bastante flexibles y solo nos dan ciertos parámetros con los que trabajar, pero el modo en que los usemos depende de nosotros. Sin embargo, siempre debemos considerar que estos elementos se han creado para ofrecer información adicional al navegador, y ayudar a cada nuevo programa y dispositivo a identificar las partes relevantes del documento. Si queremos mantener nuestro código HTML portable y legible, es mejor seguir los estándares establecidos por estos elementos. El elemento `<nav>` se ha creado con la intención de contener ayuda para la navegación, como el menú principal o bloques de enlaces importantes, y deberíamos usarlo con este propósito.

Otro ejemplo de especificidad es el que ofrecen los elementos **<main>**, **<section>**, y **<aside>**, que se han diseñado para organizar el contenido principal del documento. En nuestro diseño, estos elementos representan las secciones que llamamos **Información principal** y **Barra lateral**. Debido a que la sección **Información principal** abarca más, su contenido generalmente se representa por elementos **<section>** (uno o varios, dependiendo del diseño), y debido al tipo de información que contiene, el elemento **<aside>** se ubica en los laterales de la página. La mayoría del tiempo, estos dos elementos son suficientes para representar el contenido principal, pero como se pueden usar en otras áreas del documento, se implementa el elemento **<main>** para agruparlos, como lo muestra el siguiente ejemplo.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Este es el título del documento</title>
8 </head>
9 <body>
10  <header>
11    Este es el título
12  </header>
13
14  <nav>
15    Principal | Fotos | Videos | Contacto
16  </nav>
17
18  <main>
19    <section>
20      Artículos
21    </section>
22    <aside>
23      cita del artículo uno
24      cita del artículo dos
25    </aside>
26  </main>
27 </body>
28 </html>
```

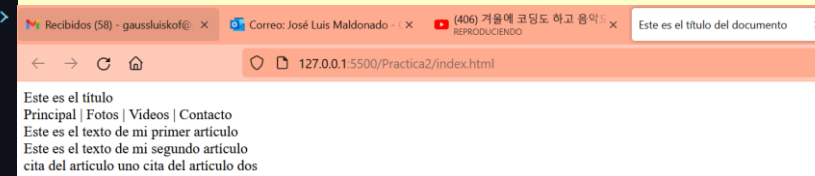


El elemento **<aside>** describe la información que contiene, no un lugar en la estructura, por lo que se podría ubicar en cualquier parte del diseño, y se puede usar mientras su contenido no se considere el contenido principal del documento. Para mostrar las secciones en el lugar correcto, debemos asignar estilos CSS a cada elemento del documento.

Como los artículos de un diario, las páginas web generalmente presentan la información dividida en secciones que comparten características similares. El elemento **<article>** nos permite identificar cada una de estas partes.

En el siguiente ejemplo, implementamos este elemento para representar las publicaciones que queremos mostrar en la sección principal de nuestra página web.

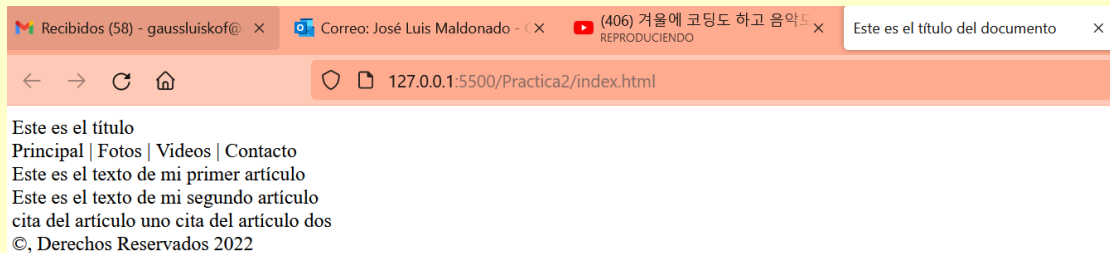
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Este es el título del documento</title>
8 </head>
9 <body>
10  <header>
11    Este es el título
12  </header>
13
14  <nav>
15    Principal | Fotos | Videos | Contacto
16  </nav>
17
18  <main>
19    <section>
20      <article>
21        Este es el texto de mi primer artículo
22      </article>
23      <article>
24        Este es el texto de mi segundo artículo
25      </article>
26    </section>
27    <aside>
28      cita del artículo uno
29      cita del artículo dos
30    </aside>
31  </main>
32 </body>
33 </html>
```



# Desarrollo Web



En este punto, ya contamos con la cabecera y el cuerpo del documento, secciones con ayuda para la navegación y el contenido, e información adicional a un lado de la página. Lo único que nos queda por hacer es cerrar el diseño y finalizar el cuerpo del documento. Con este fin, HTML ofrece el elemento **<footer>**.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Este es el título del documento</title>
8 </head>
9 <body>
10  <header>
11    Este es el título
12  </header>
13
14  <nav>
15    Principal | Fotos | Videos | Contacto
16  </nav>
17
18  <main>
19    <section>
20      <article>
21        Este es el texto de mi primer artículo
22      </article>
23      <article>
24        Este es el texto de mi segundo artículo
25      </article>
26    </section>
27    <aside>
28      cita del artículo uno
29      cita del artículo dos
30    </aside>
31  </main>
32  <footer>
33    &copy, Derechos Reservados 2022
34  </footer>
35 </body>
36 </html>
```

El ejemplo incluye la cadena de caracteres **&copy;**; Sin embargo, cuando se carga el documento, el navegador reemplaza estos caracteres por el carácter de derechos de autor (©). Estas cadenas de caracteres se denominan *entidades* (**carácter entities**) y representan caracteres especiales que no se encuentran en el teclado o tienen un significado especial en HTML, como el carácter de derechos de autor (©), el de marca registrada (®) o los paréntesis angulares usados por HTML para definir los elementos (< y >). Cuando necesite incluir estos caracteres, debe escribir la entidad en su lugar. Por ejemplo, si quiere incluir los caracteres < y > dentro de un texto, debe representarlos con las cadenas de caracteres **&lt;** y **&gt;**. Otras entidades de uso común son **&amp;** (&), **&quot;** ("), **&apos;** ('), **&pound;** (£).



## Atributos globales

Aunque la mayoría de los elementos estructurales tienen un propósito implícito que se refleja en sus nombres, esto no significa que se deban usar solo una vez en el mismo documento. Por ejemplo, algunos elementos como **<section>** y **<aside>** se pueden utilizar muchas veces para representar diferentes partes de la estructura, y otros como **<div>** aún son implementados de forma repetida para separar contenido dentro de secciones. Por esta razón, HTML define atributos globales que podemos usar para asignar identificadores personalizados a cada elemento.

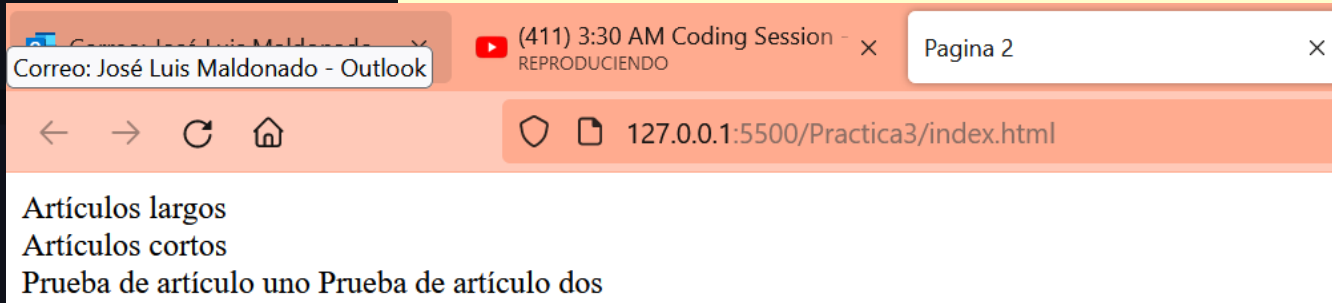


**id**—Este atributo nos permite asignar un identificador único a un elemento.

**class**—Este atributo asigna el mismo identificador a un grupo de elementos.

El atributo **id** identifica elementos independientes con un valor único, mientras que el valor del atributo **class** se puede duplicar para asociar elementos con características similares.

```
9 <body>
10 <main>
11 <section id="noticias">
12   Artículos largos
13 </section>
14 <section id="noticias locales">
15   Artículos cortos
16 </section>
17 <aside>
18   Prueba de artículo uno
19   Prueba de artículo dos
20 </aside>
21 </main>
22 </body>
23 </html>
```



Por ejemplo, si tenemos dos o más elementos **<section>** que necesitamos diferenciar entre sí, podemos asignar el atributo **id** a cada uno con valores que declaren sus propósitos.

Por otro lado, si lo que necesitamos es identificar un grupo de elementos con características similares, podemos usar el atributo **class**. El siguiente ejemplo divide el contenido de una sección con elementos **<div>**. Debido a que todos tienen un contenido similar, compartirán los mismos estilos y, por lo tanto, deberíamos identificarlos con el mismo valor (todo son de la misma clase).

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Pagina 2</title>
8  </head>
9  <body>
10     <main>
11         <section>
12             <div class="libros">Libro: IT, Stephen King</div>
13             <div class="libros">Libro: Carrie, Stephen King</div>
14             <div class="libros">Libro: El resplandor, Stephen King</div>
15             <div class="libros">Libro: Misery, Stephen King</div>
16         </section>
17         <aside>
18             Cita del artículo uno
19             Cita del artículo dos
20         </aside>
21     </main>
22 </body>
23 </html>
```

En el ejemplo, tenemos un **<section>** con el que representamos el contenido principal del documento, pero hemos creado varias divisiones con elementos **<div>** para organizar el contenido. Estos elementos se han identificado con el atributo **class** y el valor "libros", cada vez que accedemos o modificamos elementos referenciando la clase **libros**, todos estos elementos se ven afectados.

**Contenido** Los elementos HTML estudiados hasta el momento ayudan a identificar cada sección del diseño y asignarles un propósito, pero lo que importa en una página web es lo que hay dentro de esas secciones. Estas se componen por diferentes elementos visuales, como títulos, textos, imágenes y vídeos, entre otros.

## Texto

El medio más importante que puede incluir un documento es texto. HTML define varios elementos para determinar el propósito de cada palabra, frase, o párrafo en el documento. El siguiente elemento se usa para representar títulos.

**<h1>**—Este elemento representa un título. El título se declara entre las etiquetas de apertura y cierre. HTML también incluye elementos adicionales para representar subtítulos, hasta seis niveles (**<h2>**, **<h3>**, **<h4>**, **<h5>**, y **<h6>**).



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Este es el título del documento</title>
8 </head>
9 <body>
10   <header>
11     <h1>Este es el título</h1>
12   </header>
13
14   <nav>
15     Principal | Fotos | Videos | Contacto
16   </nav>
17
18   <main>
19     <section>
20       <article>
21         Este es el texto de mi primer artículo
22       </article>
23       <article>
24         Este es el texto de mi segundo artículo
25       </article>
26     </section>
27     <aside>
28       cita del artículo uno
29       cita del artículo dos
30     </aside>
31   </main>
32   <footer>
33     &copy, Derechos Reservados 2022
34   </footer>
35 </body>
36 </html>
```



(411) 3:30 AM Coding Session - x  
REPRODUCIENDO

Este es el título del documento x



127.0.0.1:5500/Practic

## Este es el título

Principal | Fotos | Videos | Contacto

Este es el texto de mi primer artículo

Este es el texto de mi segundo artículo

cita del artículo uno cita del artículo dos

©, Derechos Reservados 2022

Los siguientes son los elementos que ofrece HTML para representar el cuerpo del texto.

**<p>**—Este elemento representa un párrafo. Por defecto, los navegadores le asignan un margen en la parte superior para separar un párrafo de otro.

**<pre>**—Este elemento representa un texto con formato predefinido, como código de programación o un poema que requiere que los espacios asignados a cada carácter y los saltos de línea se muestren como se han declarado originalmente.

**<span>**—Este elemento puede contener un párrafo, una frase o una palabra. No aplica ningún estilo al texto pero se usa para asignar estilos personalizados.

El elemento **<p>** se utiliza ampliamente para representar el cuerpo del texto. Por defecto, los navegadores les asignan estilos que incluyen márgenes y un salto de línea para diferenciar un párrafo de otro. Debido a estas características, también podemos utilizar los elementos **<p>** para dar formato a líneas de texto, como las citas de nuestro ejemplo.

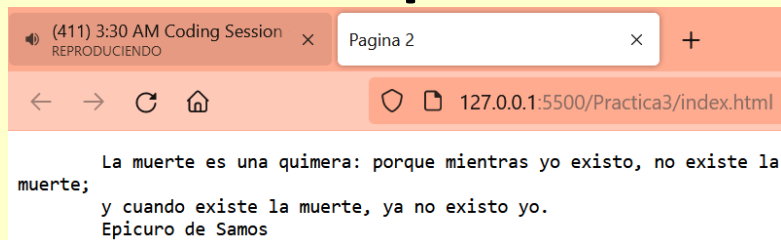




```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Este es el título del documento</title>
8 </head>
9 <body>
10 <header>
11   <h1> Este es el título</h1>
12 </header>
13
14 <nav>
15   Principal | Fotos | Videos | Contacto
16 </nav>
17
18 <main>
19   <section>
20     <article>
21       Este es el texto de mi primer artículo
22     </article>
23     <article>
24       Este es el texto de mi segundo artículo
25     </article>
26   </section>
27   <aside>
28     <p>cita del artículo uno</p>
29     <p>cita del artículo dos</p>
30   </aside>
31 </main>
32 <footer>
33   &copy, Derechos Reservados 2022
34 </footer>
35 </body>
36 </html>
```



Cuando un párrafo incluye múltiples espacios, el elemento `<p>` automáticamente reduce ese espacio a solo un carácter e ignora el resto. El elemento también hace algo similar con los saltos de línea. Todo salto de línea introducido en el documento no se considera cuando el texto se muestra en la pantalla. Si queremos que estos espacios y saltos de línea se muestren al usuario, en lugar de usar el elemento `<p>` tenemos que usar el elemento `<pre>`.



```
<body>
  <article>
    <pre>
      La muerte es una quimera: porque mientras yo existo, no existe la
muerte;
      y cuando existe la muerte, ya no existo yo.
      Epicuro de Samos
    </pre>
  </article>
</body>
```

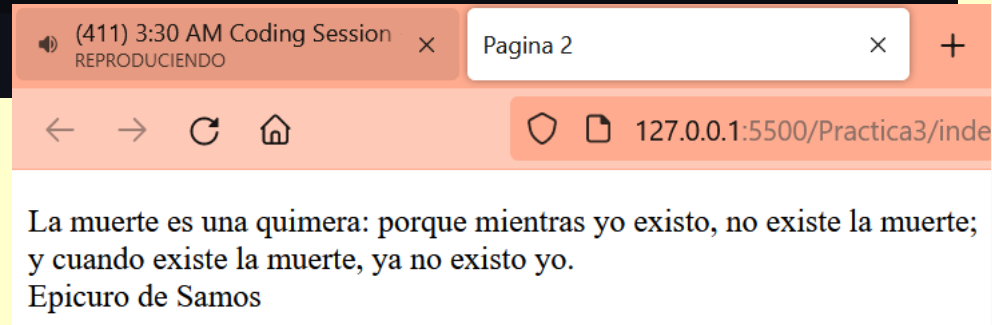
El elemento **<pre>** se configura por defecto con márgenes y un tipo de letra que respeta el formato asignado al texto original, lo que lo hace apropiado para presentar código de programación y cualquier clase de texto con formato predefinido.

**<br>**—Este elemento se usa para insertar saltos de línea.

**<wbr>**—Este elemento sugiere la posibilidad de un salto de línea para ayudar al navegador a decidir dónde cortar el texto cuando no hay suficiente espacio para mostrarlo entero.



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Pagina 2</title>
8  </head>
9  <body>
10 <article>
11 <p>
12     La muerte es una quimera: porque mientras yo existo, no existe la
13     muerte;<br>y cuando existe la muerte, ya no existo yo.<br>Epicuro de
14     Samos</p>
15 </p>
16 </article>
17 </body>
18 </html>
```



Debido a que no todas las palabras en un texto tienen el mismo énfasis, HTML incluye los siguientes elementos para declarar un significado especial a palabras individuales o frases completas.

**<em>**—Este elemento se usa para indicar énfasis. El texto se muestra por defecto con letra cursiva.

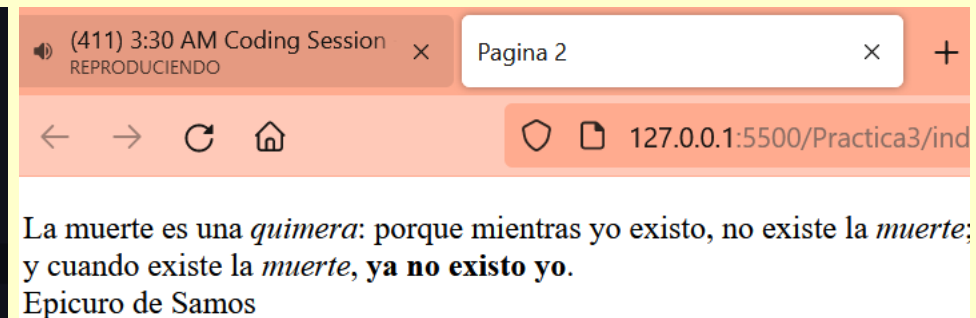
**<strong>**—Este elemento se utiliza para indicar importancia. El texto se muestra por defecto en negrita.

**<i>**—Este elemento representa una voz alternativa o un estado de humor, como un pensamiento, un término técnico, etc. El texto se muestra por defecto con letra cursiva.

**<u>**—Este elemento representa texto no articulado. Por defecto se muestra subrayado.

**<b>**—Este elemento se usa para indicar importancia. Debería ser implementado solo cuando ningún otro elemento es apropiado para la situación. El texto se muestra por defecto en negrita.

```
0 <article>
1 <p>
2   La muerte es una <em>quimera</em>: porque mientras yo existo, no
3   existe la <i>muerte</i>;<br>y cuando existe la <i>muerte</i>,
4   <strong>ya no existo yo</strong>.<br>Epicuro de Samos
5 </p>
6
7 </article>
```



La especificidad de elementos estructurales también se manifiesta en algunos de los elementos utilizados para definir el contenido. Por ejemplo, HTML incluye los siguientes elementos para insertar textos que tienen un propósito claramente definido.

**<mark>**—Este elemento resalta texto que es relevante en las circunstancias actuales (por ejemplo, términos que busca el usuario).

**<small>**—Este elemento representa letra pequeña, como declaraciones legales, descargos, etc.

**<cite>**—Este elemento representa el autor o título de una obra, como un libro, una película, etc.

**<address>**—Este elemento representa información de contacto. Se implementa con frecuencia dentro de los pies de página para definir la dirección de la empresa o el sitio web.

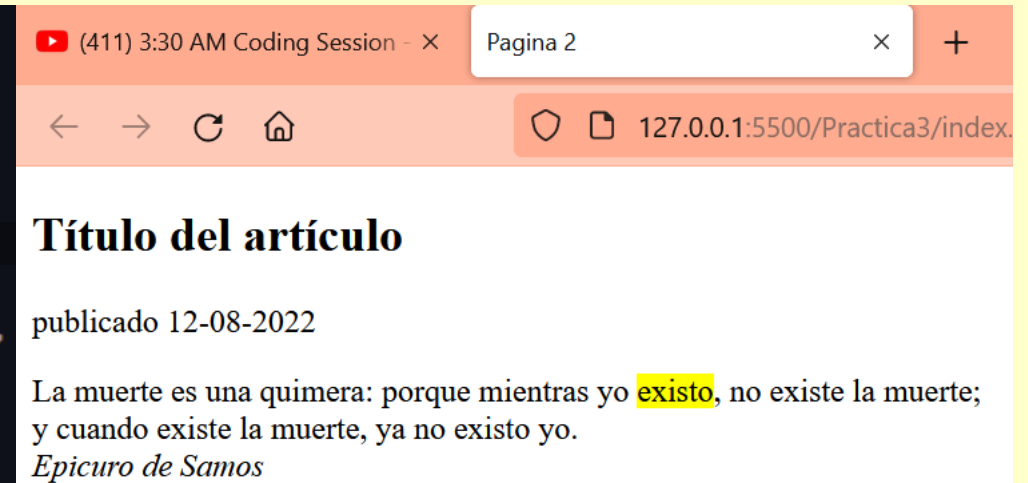
**<time>**—Este elemento representa una fecha en formato legible para el usuario. Incluye el atributo **datetime** para especificar un valor en formato de ordenador y el atributo **pubdate**, el cual indica que el valor asignado al atributo **datetime** representa la fecha de publicación.

**<code>**—Este elemento representa código de programación. Se usa en conjunto con el elemento **<pre>** para presentar código de programación en el formato original.



**<data>**—Este elemento representa datos genéricos. Puede incluir el atributo **value** para especificar el valor en formato de ordenador (por ejemplo, **<data value="32">Treinta y Dos</data>**).

```
9 <body>
10   <article>
11     <header>
12       <h1>Título del artículo</h1>
13       <time datetime="202-08-12" pubdate>publicado 12-08-2022</time>
14     </header>
15     <p>
16       La muerte es una quimera: porque mientras yo <mark>existo</mark>,
17       no existe la muerte;<br>y cuando existe la muerte, ya no existo
18       yo.<br><cite>Epicuro de Samos</cite>
19     </p>
20   </article>
21 </body>
```



Por defecto, los navegadores asignan estilos al texto dentro del elemento **<mark>** que incluyen un fondo amarillo y muestran el contenido del elemento **<cite>** en cursiva, tal como ilustra en el ejemplo.

El atributo **pubdate** es un atributo booleano. Este tipo de atributos no requieren un valor; representan el valor **true** (verdadero) cuando están presentes o **false** (falso) en caso contrario.

Los elementos **<address>** y **<small>** se insertan normalmente dentro de un elemento **<footer>** para resaltar información acerca de la página o una sección.



```
9  <body>
10  <article>
11    <header>
12      <h1>Título del artículo</h1>
13      <time datetime="202-08-12" pubdate>publicado 12-08-2022</time>
14    </header>
15    <p>
16      La muerte es una quimera: porque mientras yo <mark>existo</mark>,
17      no existe la muerte;<br>y cuando existe la muerte, ya no existo
18      yo.<br><cite>Epicuro de Samos</cite>
19    </p>
20  </article>
21  <footer>
22    <address>Ciudad de México, México</address>
23    <small>&copy; Derechos Reservados 2022</small>
24  </footer>
25 </body>
```

