

**TALENTO
DIGITAL**
INTELIGENCIA
HUMANA

Talento Digital para Chile:

Módulo 3 Fundamentos de Desarrollo Web

UN PROYECTO DE:

DESARROLLADO POR:



MÓDULO 3 – FUNDAMENTOS DE DESARROLLO WEB

3.1.- Contenido 1: Construcción de páginas Web

Objetivo de la jornada

- Conocer y utilizar herramientas que permitan crear sitios web.
 - Construir un documento HTML utilizando las principales etiquetas y atributos para resolver un problema planteado.
 - Detectar inconsistencias de los elementos de una página web usando las herramientas presentes en un navegador de uso masivo.
-

3.1.1.- El Entorno de Desarrollo

3.1.1.1.- ¿Qué es Visual Studio Code?

Visual Studio Code es un editor de código fuente creado por la empresa Microsoft, desde una perspectiva multiplataforma. Actualmente se encuentra disponible para los sistemas operativos Windows, Linux y macOS [1].

Esta aplicación además permite ejecutar operaciones de desarrollo, tales como la depuración de código, ejecución de tareas y control de versiones. Su objetivo es proporcionar solo aquellas herramientas que un desarrollador necesita para un ciclo eficiente de creación de código y depuración. Los flujos más complejos, en tanto, se dejan en manos de entornos de desarrollo integrados (IDE) más complejos.

Una característica interesante de este software, es que su uso es liberado, tanto para uso privado o comercial. Para descargarlo, solo deben acceder al enlace incluido en las referencias.

3.1.1.2.- ¿Por qué usar un editor de texto?

En términos generales, un editor de texto es un tipo de software diseñado para crear, modificar y almacenar archivos de texto. Se diferencia de los procesadores de texto tradicionales por la ausencia de opciones de formato avanzadas y de la posibilidad de mostrar elementos gráficos o multimedia. Casi todos los sistemas operativos masivos cuentan con un editor de texto, además de los editores de texto creados por terceros.

Entre las ventajas que considera una herramienta de esta naturaleza, se destacan las siguientes:

- **Multiplataforma:** cada sistema operativo cuenta con un conjunto de programas de escritorio que permiten editar archivos de texto, algunos incorporados y otros creados por entidades externas. Independiente de esta diversidad, situaciones tales como el manejo de saltos de líneas se encuentra estandarizado, mejorando la visualización y manipulación.
- **Integridad de los datos:** un editor de texto por lo general no agrega caracteres extraños provenientes de formatos o elementos gráficos, permitiendo generar archivos de texto plano. Para codificar sitios web, es fundamental que las etiquetas incorporadas sean incluidas como texto plano, ya que cualquier formato adicional puede alterar la integridad del archivo, generando errores en su visualización en un navegador.
- **Tamaño de archivos:** otra ventaja de los editores de texto, es la posibilidad de manipular archivos de gran tamaño, situación que no se aplica a procesadores de texto habituales. Esto incluye tanto la edición de los documentos como la búsqueda de términos.

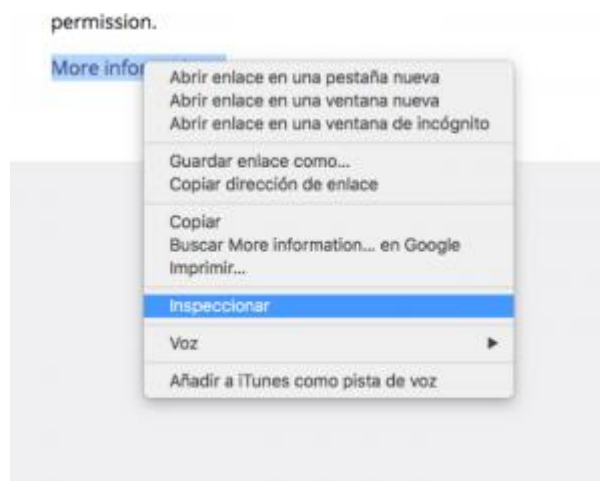
3.1.1.3.- Conociendo el inspector de elementos en un navegador

Un navegador web es una aplicación de software que comunica un dispositivo cliente (PC, notebook, celular, Tablet, etc.) con un servidor web, al cual se le solicita parte de la información que tiene almacenada. Si lo analizamos en términos simples, una página web no es más que un conjunto de archivos de texto plano y documentos multimedia que son interpretados por un navegador, brindándole forma.

Los navegadores más usados en la actualidad son Microsoft Edge, Google Chrome, Mozilla Firefox y Opera [2]. Si bien todos ellos cumplen un objetivo específico y transversal, presentan características que los diferencian de sus pares, y que traen como consecuencia que los usuarios tengan un navegador de preferencia.

Una herramienta que es transversal en los navegadores conocidos es el “inspector de elementos”, la cual es permite analizar los contenidos de cualquier sitio web. A través de esta aplicación se puede ver el código fuente de la página, así como analizar cada uno de los componentes que la conforman. Es de mucha utilidad cuando se quiere ver cómo está hecho algo en un sitio, o bien para afinar estilos y depurar scripts que se ejecuten.

Cada inspector de los navegadores tiene diferencias menores, pero las acciones básicas a desarrollar son prácticamente las mismas. En las imágenes siguientes se usará como referencia el navegador Google Chrome. Para acceder al inspector de elementos debe seleccionar el objeto a analizar, y con el botón derecho del ratón seleccionar la opción “Inspeccionar”.



El inspector se compone de varias pestañas, donde cada una determina distintas funcionalidades y características:



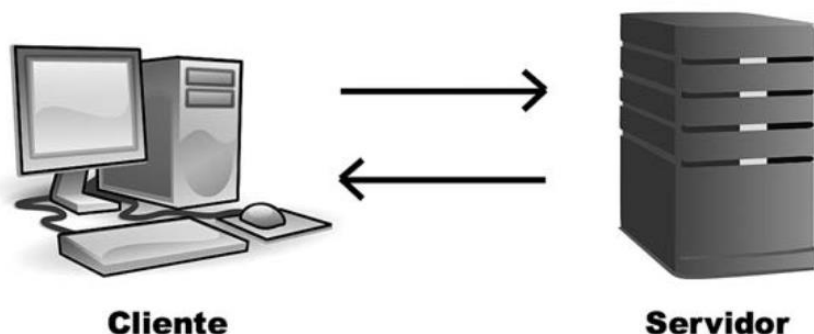
- **Elements:** se puede revisar el código fuente del sitio, junto a los estilos que se aplican sobre ella. Se pueden modificar localmente y añadir nuevas reglas.
- **Console:** en esta sección se muestran los mensajes de error y de advertencias que se producen en la página (imágenes que no cargan correctamente, errores de Javascript, entre otros).
- **Sources:** recursos de la página, expresados en forma de árbol. Se puede analizar su origen y modificarlos de manera local.
- **Network:** esta ventana permite analizar cada una de las peticiones que se hacen a un servidor web, considerando en ello el contenido de la petición, la respuesta que se obtuvo y los tiempos asociados.
- **Performance:** herramientas que permiten medir el desempeño de un sitio.
- **Memory:** mide la memoria que se consume durante el proceso de carga del sitio y durante la ejecución de las tareas respectivas.
- **Application:** datos de utilidad sobre los recursos que emplea una aplicación web.
- **Security:** información sobre los sitios que están enlazados al portal analizado, y el estado de su certificado de seguridad respectivo.
- **Audits:** contiene una serie de controles predeterminados que permiten validar errores típicos.

3.1.2.- El Lenguaje HTML

3.1.2.1.- Introducción al lenguaje de etiquetas de hipertexto

Los sitios web son archivos que los usuarios descargan con sus navegadores desde equipos remotos. Cuando un usuario accede a una página, le indica al navegador la dirección del mismo y descarga los archivos, procesa el contenido y lo despliega. Dado

que los archivos deben estar disponibles todo el tiempo, no pueden ser almacenados en equipos convencionales sino que deben estar dispuestos en equipos especializados. El equipo que hace la consulta se denomina “cliente”, mientras que el que la procesa y envía una respuesta es el “servidor” [3].



Un sitio web es la suma de documentos y recursos multimedia que el navegador descarga cuando el usuario los solicita. Los documentos que forman un sitio web se llaman “páginas” y el proceso de abrir nuevas páginas se conoce como “navegar”.

Si se desea crear un sitio web, es necesario crear un archivo por cada página que se incluirá. Junto con estos archivos, se deben incluir los archivos con las imágenes y cualquier otro recurso que se desplegará en estas páginas; considera que las imágenes y otros medios gráficos se almacenan en archivos aparte.

Los archivos de un sitio web son similares a los documentos que podemos encontrar en un computador común. Cada uno de ellos tiene un nombre asignado por el desarrollador y una extensión que determina el lenguaje utilizado para crear su contenido. Aunque es posible asignar cualquier nombre a estos archivos, el documento que genera la página inicial presenta algunos requisitos, como por ejemplo que el sitio inicial tenga como nombre “index.html”; sin embargo, esta última condición es configurable de acuerdo a las configuraciones aplicadas sobre el software que actúa como servidor web.

Cuando un sitio ha sido creado, todos los archivos que lo componen son almacenados en un servidor web. Cada servidor se identifica con un valor llamado IP (Internet Protocol). Esta IP es única para cada equipo y, por lo tanto, trabaja como una dirección que permite ubicarlo dentro de una red. Cuando el browser web tiene que acceder al 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.

En teoría, se puede acceder a cualquier servidor utilizando su dirección IP, pero estos valores son difíciles de recordar. Por lo mismo, Internet usa 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).

3.1.2.2.- ¿Qué es HTML?

La sigla HTML es una abreviatura que viene del inglés “**HyperText Markup Language**”, y es un lenguaje compuesto por un conjunto de etiquetas definidas con un nombre rodeado de paréntesis “<” y “>”. 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,
) y otras son declaradas en pares, considerando 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.

Es necesario incluir varios elementos para definir un documento. Los elementos son listados en secuencia de descendente, y pueden contener otros elementos en su interior. Por ejemplo, la se dijo que la etiqueta <html></html> indica el inicio y cierre de cualquier sitio; 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, las etiquetas dentro del elemento <html> pueden incluir otras más. El ejemplo indicado a continuación muestra un documento HTML sencillo que incluye todos los elementos necesarios para definir una estructura básica y mostrar el mensaje “HOLA MUNDO!” en un navegador.

```
1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <title>Mi primer documento HTML</title>
5    </head>
6    <body>
7      <p>HOLA MUNDO!</p>
8    </body>
9  </html>
```

En el ejemplo anterior se muestra un código que podría resultar sencillo en principio, pero que posee una estructura compleja. En la primera línea se encuentra una etiqueta que señala el tipo de documento existente en las líneas posteriores (`<!DOCTYPE html>`), seguida por una etiqueta de apertura `<html lang="es">`. 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. Los elementos se listan uno a continuación de otro y también dentro de otros elementos, de modo que se construye una estructura de tipo árbol con el elemento `<html>` como raíz.

3.1.2.3.- Conceptos básicos sobre HTML

Como parte del estudio del HTML, existen conceptos claves que es necesario conocer. Por lo mismo, se irán abordando en detalle en las clases siguientes; por lo pronto, se dispone una lista de conceptos y su definición.

HTML5

Es un lenguaje de etiquetas usado para presentar y estructurar el contenido en un sitio web. Es la quinta revisión del estándar HTML, el cual fue creado en 1990. La W3C (World Wide Web Consortium), entidad que determina las bases del lenguaje, recomendó su utilización para transformarse en el estándar a ser usado en proyectos venideros. Está relacionado también con la entrada en decadencia del anterior estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que se pueden ver en la actualidad. Con HTML5 existen otras posibilidades para explotar sitios usando menos recursos. Con HTML5 entra conjuntamente en desuso el formato XHTML, dado que ya no sería necesaria su implementación.

Se trata de un sistema para formatear el “layout” o disposición de nuestras páginas, así como hacer algunos ajustes a su aspecto. Con HTML5 los navegadores como Firefox, Chrome, Explorer, Safari y más pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto. En este sentido, el HTML5 no se diferencia demasiado de su predecesor. La diferencia principal, sin embargo, es el nivel de sofisticación del código que podremos construir usando HTML5. Las etiquetas que se indicarán más adelante son una combinación del estándar original con HTML5.

Hipervínculos

También llamados “enlaces”, 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 clic 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 (recursos, en nuestro ejemplo).

CSS – Hojas de estilo en cascada

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.

```
1  body {  
2      width: 100%;  
3      margin: 0px;  
4      background-color: #FF0000;  
5  }  
6
```

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. En el ejemplo anterior, el valor “#FF0000” se asigna a la propiedad “background-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). En el mismo ejemplo se declara también una regla con tres propiedades: **width**, **margin** y **background-color**. 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.

JavaScript

A diferencia de HTML y CSS, JavaScript es un lenguaje de programación. Para ser justos, todos estos lenguajes pueden ser considerados lenguajes de programación, pero en la práctica existen algunas diferencias en la forma en la que entregan las instrucciones al navegador. HTML es como un grupo de indicadores que el navegador interpreta para organizar la información, CSS puede ser considerado como una lista de estilos que ayudan al navegador a preparar el documento para ser presentado en pantalla (aunque la última especificación lo convirtió en un lenguaje más dinámico), pero JavaScript es un lenguaje de programación, comparable con cualquier otro lenguaje de programación profesional como C++ o Java. JavaScript difiere de los demás lenguajes en que puede realizar tareas personalizadas, desde almacenar valores hasta calcular algoritmos complejos, incluida la capacidad de interactuar con los elementos del documento y procesar su contenido dinámicamente.

```
1  <script>
2      function cambiarColor() {
3          document.body.style.backgroundColor = "#0000FF";
4      }
5      document.addEventListener("click", cambiarColor);
6  </script>
```

Al igual que HTML y CSS, JavaScript se incluye en los navegadores y, por lo tanto, se encuentra disponible para todos los documentos creados. Para declarar código JavaScript dentro de un documento, HTML ofrece el elemento **<script>**. El siguiente ejemplo es una muestra de un código escrito en JavaScript; el código del ejemplo cambia el color de fondo del elemento **<body>** a azul cuando el usuario hace clic en el documento.

Lenguajes de servidor

Los códigos programados en HTML, CSS, y JavaScript son ejecutados por el navegador en el computador del usuario (el cliente). Esto significa que, después de que los archivos del sitio web se suben al servidor, permanecen inalterables hasta que se descargan en un equipo personal y sus códigos son ejecutados por el navegador o browser. Aunque esto permite la creación de sitios web útiles e interactivos, hay momentos en los cuales necesitamos procesar la información en el servidor antes de enviarla al usuario. El contenido producido por esta información se denomina contenido dinámico, y es generado por códigos ejecutados en el servidor y programados en lenguajes que fueron especialmente diseñados con este propósito (lenguajes de servidor). Cuando el navegador solicita un archivo que contiene este tipo de código, el servidor lo ejecuta y luego envía el resultado como respuesta al usuario. Estos códigos no solo se utilizan para generar contenido y documentos en tiempo real, sino también para procesar la información enviada por el navegador, almacenar datos del usuario en el servidor, controlar cuentas, etc.

Existen varios lenguajes disponibles para crear código ejecutable en los servidores. Los más populares son PHP, Ruby, y Python. En este curso haremos uso del lenguaje Java para programación web, bajo la lógica que promueven las “JavaServer Pages” o JSP.

3.1.2.4.- Estructura de un documento HTML

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.

Los elementos HTML conforman una estructura de tipo árbol con el elemento `<html>` como su raíz. Esta estructura presenta múltiples niveles de organización, con algunos elementos a cargo de definir secciones generales del documento y otros encargados de representar secciones menores o contenido. 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.

<code><html></code>	Este elemento delimita el código HTML. Puede incluir el atributo lang para definir el idioma del contenido del documento.
<code><head></code>	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.
<code><body></code>	Este elemento delimita el contenido del documento (la parte visible de la página).

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.

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. Entre las etiquetas que son parte de la cabecera están descritas en la tabla siguiente.

<title>	Este elemento define el título de la página.
<base>	Este elemento define la URL usada por el navegador para establecer la ubicación real de las URL relativas. El elemento debe incluir el atributo href para declarar la URL base. Cuando se declara este elemento, en lugar de la URL actual, el navegador usa la URL asignada al atributo href para completar las URL relativas.
<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.
<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).
<style>	Este elemento se usa para declarar estilos CSS dentro del documento
<script>	Este elemento se usa para cargar o declarar código JavaScript

Como parte de la sección visible de una página (etiqueta <body>), se incluyen todas las estructuras que dan vida al contenido del sitio, y que permiten desplegar la información. Debido a que esta información está compuesta por diferentes elementos visuales, como títulos, textos, imágenes y videos, entre otros, HTML define varios elementos para representarla. A continuación se indica un listado con etiquetas presentes en esta sección, y su función.

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

	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.
	Este elemento se usa para indicar énfasis. El texto se muestra por defecto con letra cursiva.
	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.
	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.

<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 computador (por ejemplo, <data value="32">Treinta y Dos</data>).
<a>	Este elemento crea un enlace. El texto o la imagen que representa el enlace se incluye entre las etiquetas de apertura y cierre. El elemento incluye el atributo href para especificar la URL del enlace.
<download>	Este es un atributo booleano que, cuando se incluye, indica que en lugar de leer el archivo el navegador debería descargarlo.
<ping>	Este atributo declara la ruta del archivo que se debe abrir en el servidor cuando el usuario hace clic en el enlace. El valor puede ser una o más URL separadas por un espacio.

Además de las etiquetas mencionadas en las tablas anteriores, dentro del estándar se consideran categorías que determinan diferentes comportamientos y usos del lenguaje.

Imágenes

Las imágenes pueden ser consideradas el segundo medio más importante en la Web. HTML incluye los siguientes elementos para introducir imágenes en los documentos que conforman un portal.

	Este elemento inserta una imagen en el documento. El elemento requiere del atributo src para especificar la URL del archivo con la imagen que queremos incorporar.
<picture>	Este elemento inserta una imagen en el documento. Trabaja junto con el elemento <source> para ofrecer múltiples imágenes en diferentes resoluciones. Es útil para crear sitios web adaptables.
<figure>	Este elemento representa contenido asociado con el contenido principal, pero que se puede eliminar sin que se vea afectado, como fotos, vídeos, etc.
<figcaption>	Este elemento introduce un título para el elemento <figure> .

Listados

A menudo la información se debe representar como una lista de ítems. Por ejemplo, muchos sitios web incluyen listados de libros, películas, o términos y descripciones. Para crear estos listados, HTML ofrece los siguientes elementos.

	Este elemento crea una lista de ítems sin orden. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems (y) y trabaja junto con el elemento para definir cada uno de los ítems de la lista.
	Este elemento crea una lista ordenada de ítems. Está compuesto por etiquetas de apertura y cierre para agrupar los ítems (y) y trabaja junto con el elemento para definir los ítems de la lista. Este elemento puede incluir los atributos reversed para invertir el orden de los indicadores, start para determinar el valor desde el cual los indicadores tienen que comenzar a contar y type para determinar el tipo de indicador que queremos usar. Los valores disponibles para el atributo type son 1 (números), a (letras minúsculas), A (letras mayúsculas), i (números romanos en minúsculas) e I (números romanos en mayúsculas).
<dl>	Este elemento crea una lista de términos y descripciones. El elemento trabaja junto con los elementos <dt> y <dd> para definir los ítems de la lista. El elemento <dl> define la lista, el elemento <dt> define los términos y el elemento <dd> define las descripciones.

Los siguientes elementos se han diseñado con propósitos diferentes, pero también se utilizan frecuentemente para construir listas de ítems.

<blockquote>	Este elemento representa un bloque de texto que incluye una cita tomada de otro texto en el documento.
<details>	Este elemento crea una herramienta que se expande cuando se hace clic en ella para mostrar información adicional. La parte visible se define con el elemento <summary> , y se pueden usar elementos comunes como <p> para definir el contenido.

Tablas

Las tablas organizan información en filas y columnas. Debido a sus características, se usaron durante mucho tiempo para estructurar documentos HTML, pero con la introducción de CSS, los desarrolladores pudieron lograr el mismo efecto implementando otros elementos. Aunque ya no se recomienda usar tablas para definir la estructura de un documento, todavía se utilizan para presentar información tabular, como estadísticas o especificaciones técnicas, por ejemplo. HTML incluye varios elementos para crear una tabla. Los siguientes son los más utilizados.

<table>	Este elemento define una tabla. Incluye etiquetas de apertura y cierre para agrupar el resto de los elementos que definen la tabla.
<tr>	Este elemento define una fila de celdas. Incluye etiquetas de apertura y cierre para agrupar las celdas.
<td>	Este elemento define una celda. Incluye etiquetas de apertura y cierre para delimitar el contenido de la celda y puede incluir los atributos colspan y rowspan para indicar cuántas columnas y filas ocupa la celda.
<th>	Este elemento define una celda para la cabecera de la tabla. Incluye etiquetas de apertura y cierre para delimitar el contenido de la celda y puede incluir los atributos colspan y rowspan para indicar cuántas columnas y filas ocupa la celda.

Formularios

Los formularios son herramientas que podemos incluir en un documento para permitir a los usuarios insertar información, tomar decisiones, comunicar datos y cambiar el comportamiento de una aplicación. El propósito principal de los formularios es permitir al usuario seleccionar o insertar información y enviarla al servidor para ser procesada.

Los formularios pueden presentar varias herramientas que permiten al usuario interactuar con el documento, incluidos campos de texto, casillas de control, menús desplegables y botones. Cada una de estas herramientas se representa por un elemento y el formulario queda definido por el elemento `<form>`, que incluye etiquetas de apertura y cierre para agrupar al resto de los elementos y requiere de algunos atributos para determinar cómo se envía la información al servidor. Los atributos posibles se indican en la tabla a continuación.

name	Este atributo especifica el nombre del formulario. También se encuentra disponible para otros elementos, pero es particularmente útil para elementos de formulario, como veremos más adelante.
method	Este atributo determina el método a utilizar para enviar la información al servidor. Existen dos valores disponibles: GET y POST. El método GET se usa para enviar una cantidad limitada de información de forma pública (los datos son incluidos en la URL, la cual no puede contener más de 255 caracteres). Por otro lado, el método POST se utiliza para enviar una cantidad ilimitada de información de forma privada (los datos no son visibles al usuario y pueden tener la longitud que necesitemos).
action	Este atributo declara la URL del archivo en el servidor que va a procesar la información enviada por el formulario.
target	Este atributo determina dónde se mostrará la respuesta recibida desde el servidor. Los valores disponibles son <code>_blank</code> (nueva ventana), <code>_self</code> (mismo recuadro), <code>_parent</code> (recuadro padre), y <code>_top</code> (la ventana que contiene el recuadro). El valor <code>_self</code> se declara por defecto, lo que significa que la respuesta recibida desde el servidor se mostrará en la misma ventana.
enctype	Este atributo declara la codificación aplicada a los datos que envía el formulario. Puede tomar tres valores: <code>application/x-www-form-urlencoded</code> (los caracteres son codificados), <code>multipart/form-data</code> (los caracteres no son codificados), <code>text/plain</code> (solo los espacios son codificados). El primer valor se asigna por defecto.
accept-charset	Este atributo declara el tipo de codificación aplicada al texto del formulario. Los valores más comunes son UTF-8 e ISO-8859-1. El valor por defecto se asigna al documento con el elemento <code><meta></code> .

Un formulario puede incluir diferentes herramientas para permitir al usuario seleccionar o insertar información. HTML incluye múltiples elementos para crear estas herramientas. Los siguientes son los más utilizados.

<input>	Este elemento crea un campo de entrada. Puede recibir diferentes tipos de entradas, dependiendo del valor del atributo type.
<textarea>	Este elemento crea un campo de entrada para insertar múltiples líneas de texto. El tamaño se puede declarar en números enteros usando los atributos rows y cols, o en píxeles con estilos CSS.
<select>	Este elemento crea una lista de opciones que el usuario puede elegir. Trabaja junto con el elemento <option> para definir cada opción y el elemento <optgroup> para organizar las opciones en grupos.
<button>	Este elemento crea un botón. Incluye el atributo type para definir el propósito del botón. Los valores disponibles son submit para enviar el formulario (por defecto), reset para reiniciar el formulario, y button para realizar tareas personalizadas.
<output>	Este elemento representa un resultado producido por el formulario. Se implementa por medio de código JavaScript para mostrar el resultado de una operación al usuario.
<meter>	Este elemento representa una medida o el valor actual de un rango.
<progress>	Este elemento representa el progreso de una operación.
<datalist>	Este elemento crea un listado de valores disponibles para otros controles. Trabaja junto con el elemento <option> para definir cada valor.
<label>	Este elemento crea una etiqueta para identificar un elemento de formulario.
<fieldset>	Este elemento agrupa otros elementos de formulario. Se usa para crear secciones dentro de formularios extensos. El elemento puede contener un elemento <legend> para definir el título de la sección.

El elemento `<input>` es el más versátil de todos. Este elemento genera un campo de entrada en el que el usuario puede seleccionar o insertar información, pero puede adoptar diferentes características y aceptar varios tipos de valores dependiendo del valor de su atributo `type`. Los siguientes son los valores disponibles para este atributo.

- **text:** Este valor genera un campo de entrada para insertar texto genérico.
- **email:** Este valor genera un campo de entrada para insertar cuentas de correo.
- **search:** Este valor genera un campo de entrada para insertar términos de búsqueda.
- **url:** Este valor genera un campo de entrada para insertar URL.
- **tel:** Este valor genera un campo de entrada para insertar números de teléfono.
- **number:** Este valor genera un campo de entrada para insertar números.
- **range:** Este valor genera un campo de entrada para insertar un rango de números.
- **date:** Este valor genera un campo de entrada para insertar una fecha.
- **datetime-local:** Este valor genera un campo de entrada para insertar fecha y hora.
- **week:** Este valor genera un campo de entrada para insertar el número de la semana (dentro del año).
- **month:** Este valor genera un campo de entrada para insertar el número del mes.
- **time:** Este valor genera un campo de entrada para insertar una hora (horas y minutos).
- **hidden:** Este valor oculta el campo de entrada. Se usa para enviar información complementaria al servidor.
- **password:** Este valor genera un campo de entrada para insertar una clave. Reemplaza los caracteres insertados con estrellas o puntos para ocultar información sensible.
- **color:** Este valor genera un campo de entrada para insertar un color.
- **checkbox:** Este valor genera una casilla de control que permite al usuario activar o desactivar una opción.
- **radio:** Este valor genera un botón de opción para seleccionar una opción de varias posibles.
- **file:** Este valor genera un campo de entrada para seleccionar un archivo en el ordenador del usuario.
- **button:** Este valor genera un botón. El botón trabaja como el elemento `<button>` de tipo `button`. No realiza ninguna acción por defecto; la acción debe ser definida desde JavaScript, como veremos en próximos capítulos.

- **submit:** Este valor genera un botón para enviar el formulario.
- **reset:** Este valor genera un botón para reiniciar el formulario.
- **image:** Este valor carga una imagen que se usa como botón para enviar el formulario. Un elemento **<input>** de este tipo debe incluir el atributo **src** para especificar la URL de la imagen.

3.1.2.5.- Etiquetas semánticas

HTML siempre ha ofrecido diferentes maneras de construir y organizar la información en el cuerpo del documento. Uno de los primeros elementos utilizados con este propósito fue **<table>** (tabla). Este elemento permitía a los desarrolladores organizar datos, textos, imágenes, así como herramientas en filas y columnas de celdas. Con la introducción de CSS, la estructura generada por estas tablas ya no resultaba práctica, por lo que los desarrolladores comenzaron a implementar un elemento más flexible llamado **<div>** (división). Pero **<div>**, así como **<table>**, no facilita demasiada información acerca de las partes del cuerpo que representa. Cualquier cosa, desde imágenes hasta menús, texto, enlaces, códigos o formularios, se puede insertar entre las etiquetas de apertura y cierre de un elemento **<div>**. En otras palabras, el nombre **div** solo especifica una división en el cuerpo, como una celda en una tabla, pero no ofrece ninguna pista acerca del tipo de división que está creando, cuál es su propósito o qué contiene. Esta es la razón por la que HTML5 introdujo nuevos elementos con nombres más descriptivos que permiten a los desarrolladores identificar cada parte del documento. Estos elementos no solo ayudan a los desarrolladores a crear el documento, sino que además informan al navegador sobre el propósito de cada sección. La siguiente lista incluye todos los elementos disponibles 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.

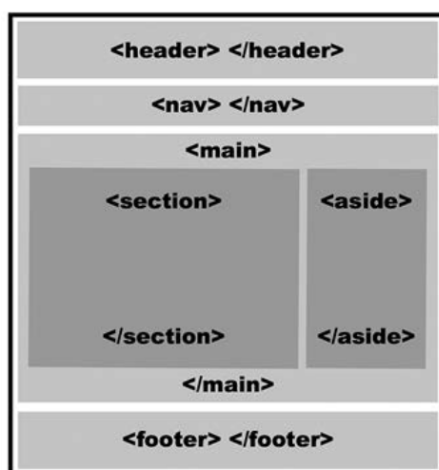
<section>	Este elemento define una sección genérica. Se usa frecuentemente para separar contenido temático, o para generar columnas o bloques que ayudan a organizar el contenido principal.
<aside>	Este elemento define una división que contiene información relacionada con el contenido principal pero que 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 que se muestra a continuación ilustra este tipo de diseño.



A pesar de que cada desarrollador crea sus propios diseños, en general podremos describir todo sitio web considerando estas secciones. En la barra superior, descrita como cabecera en la figura ubicamos el logo, el nombre del sitio, los subtítulos y una descripción breve del sitio o página web en cuestión. 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. 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 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 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.

Como se mencionó anteriormente, los elementos de HTML5 se han diseñado siguiendo este patrón. En la figura siguiente se aplican los elementos introducidos anteriormente para definir el diseño de la figura anterior.



3.1.2.6.- Accesibilidad web

La accesibilidad Web significa que personas con algún tipo de discapacidad van a poder hacer uso de la Web. Al hablar de accesibilidad Web se está haciendo referencia a un diseño Web que permitirá que estas personas puedan percibir, entender, navegar e interactuar con la Web, aportando a su vez contenidos. La accesibilidad Web también beneficia a otras personas, incluyendo personas de edad avanzada que han visto

mergadas sus habilidad a consecuencia de la edad.

La accesibilidad Web considera muchos tipos de discapacidades, incluyendo problemas visuales, auditivos, físicos, cognitivos, neurológicos y del habla. El documento "Cómo utilizan la Web personas con discapacidad" muestra la forma en la que diferentes discapacidades pueden dificultar la utilización de la Web e incluye algunos escenarios de personas con discapacidad utilizando la Web.

La Web es un recurso muy importante para diferentes aspectos de la vida: educación, empleo, gobierno, comercio, sanidad, entretenimiento y muchos otros. Es muy importante que la Web sea accesible para así proporcionar un acceso equitativo e igualdad de oportunidades a las personas con discapacidad. Una página Web accesible puede ayudar a personas con discapacidad a que participen más activamente en la sociedad.

La accesibilidad Web se ha entendido siempre como responsabilidad de los desarrolladores Web. Sin embargo, el software Web tiene también un papel importante en la accesibilidad Web. Es importante que el software ayude a los desarrolladores a generar y evaluar sitios Web accesibles para que las personas con discapacidad puedan utilizarlos. Una de las funciones de la Iniciativa de Accesibilidad Web (WAI) es desarrollar pautas y técnicas que proporcionen soluciones accesibles para el software Web y para los desarrolladores Web. Las pautas de WAI son consideradas como estándares internacionales de accesibilidad Web.

Hacer un sitio Web accesible puede ser algo sencillo o complejo, depende de muchos factores como por ejemplo, el tipo de contenido, el tamaño y la complejidad del sitio, así como de las herramientas de desarrollo y el entorno. Muchas de las características accesibles de un sitio se implementan de forma sencilla si se planean desde el principio del desarrollo del sitio Web o al comienzo de su rediseño. La modificación de sitios Web inaccesible puede requerir un gran esfuerzo, sobre todo aquellos que no se "etiquetaron" correctamente con etiquetas estándares de XHTML, y sitios con cierto tipo de contenido, como multimedia. Existen, por cierto, sitios que validan que la estructura del sitio cumpla con los estándares solicitados.

3.1.2.7.- Optimización en buscadores

SEO son las siglas en inglés de **"Search Engine Optimization"** (optimización en buscadores) o **Search Engine Optimizer** (optimizador de buscadores), y es el proceso de mejorar la visibilidad de un sitio web en los diferentes buscadores, como Google, Bing o Yahoo de manera orgánica, es decir sin pagarle dinero al buscador para tener acceso a una posición destacada en los resultados. Este posicionamiento se logra de manera

natural realizando tareas de optimización en las páginas web. Con el objetivo de aparecer en las primeras posiciones de los buscadores y aumentar el tráfico de visitas en una página web, es conveniente que en los sitios se apliquen tareas de optimización. La tarea de optimizar la estructura de una web y el contenido de la misma, así como la utilización de diversas técnicas de linkbuilding, linkbaiting o contenidos virales con el objetivo de aparecer en las primeras posiciones de los resultados de los buscadores (cuando un usuario busca por una determinada palabra clave o keyword), es conocida como SEO.

Contratar a un especialista en SEO es una decisión muy importante que puede mejorar cualquier sitio web, generando un ahorro de tiempo, aunque también puede traer consigo daño a su funcionamiento y reputación. El desarrollador debe estar seguro de investigar las posibles ventajas y perjuicios para su sitio web si la empresa solicitante contrata a un especialista en SEO adecuado. Muchos de ellos, así como otras agencias y consultorías, ofrecen servicios útiles a los propietarios de sitios web, como los siguientes:

- Revisión del contenido y la estructura del sitio web
- Asistencia técnica al desarrollo de sitios web; por ejemplo, alojamiento, redirecciones, páginas de error, uso de JavaScript y otros
- Desarrollo de contenidos
- Gestión de campañas de desarrollo empresarial online
- Investigación sobre palabras clave
- Formación en SEO
- Experiencia en zonas geográficas y mercados concretos

Si una empresa está pensando en contratar a un especialista en SEO, cuanto antes lo haga es mejor. Un buen momento para contratar estos servicios es cuando necesite cambiar el diseño de su sitio web o cuando vaya a publicar uno nuevo. De esta forma, podrá asegurarse de que el sitio se ha diseñado conforme a los requisitos del buscador. No obstante, un buen especialista en SEO también puede ayudar a mejorar un sitio web que ya esté publicado.

Se recomienda a cualquier empresa, entonces, considerar los siguientes puntos al momento de contratar a un experto SEO:

- Compromiso con la implementación de los cambios recomendados
- Entrevistar al posible especialista en SEO
- Comprobar las referencias laborales del especialista en SEO
- Solicitar una auditoría técnica y de búsqueda
- Decidir si lo contratará

Anexo 1: Referencias

[1] Sitio oficial de Visual Studio Code

Referencia: <https://code.visualstudio.com/>

[2] Mejores navegadores web

Referencia: <https://es.digitaltrends.com/computadoras/mejores-navegadores-web/>

**[3] J.D Gauchat “El Gran libro de HTML5, CCS3 y JavaScript”
3ª edición**