



El futuro digital  
es de todos

MinTIC

»

# INTRODUCCIÓN AL DISEÑO WEB USANDO EL LENGUAJE HTML

xxx



Misión  
TIC2022



# INTRODUCCIÓN

El desarrollo de aplicaciones para la web es una tendencia importante en la actualidad y es fundamental que los analistas y desarrolladores se apropien de las tecnologías que hacen posible este tipo de aplicaciones.

En primer lugar se deben apropiar los tres conceptos fundamentales o trilogía del diseño web: HTML, CSS y JavaScript.

El lenguaje de marcación de hipertextos (HyperText Markup Language) es la primera pieza de esta trilogía y permite definir la estructura de los contenidos que aparecen en una página web. Su concepción se deriva de una tecnología llamada SGML (Standard Generalized Markup Language) el cual es un lenguaje de marcación que aplicaba a cualquier tipo de documento.

Para describir la apariencia que tendrán las páginas web se propusieron las hojas de estilos en cascada o (CSS). Este mecanismo, la segunda pieza de la trilogía, permite dar una personalización o estilo a los elementos de las páginas web como son textos, imágenes, tablas, encabezados, pies de página entre otros.



- El último elemento es el lenguaje JavaScript que añade interactividad a las páginas web. Tanto HTML como CSS son lenguajes declarativos, es decir, solamente indican qué debe hacerse sin indicar los detalles. Por otra parte está JavaScript que es un lenguaje imperativo que indica cómo debe hacerse una tarea paso a paso.

Es con la ayuda de JavaScript que las páginas web se pueden realizar tareas más complejas como validar los datos de entrada, conectarse a bases de datos, emitir mensajes de error, consultar la versión del navegador, entre otras. Es importante anotar que JavaScript no tiene ninguna relación con el lenguaje de programación Java, son dos cosas distintas que tienen un nombre similar por mera casualidad.

Por último han surgido componentes de software estructurados en los llamados “Frameworks” que facilitan el desarrollo de aplicaciones mediante la integración de las tecnologías descritas anteriormente. Ejemplos de frameworks JavaScript son: AngularJS desarrollado por Google, ReactJS desarrollado por Facebook. Un ejemplo de un framework CSS es Bootstrap desarrollado por Twitter.

También están disponibles componentes estructurados en librerías. Uno de los más famosos y utilizados es JQuery que es una librería JavaScript que permite modificar los elementos HTML de manera dinámica. En este recurso se hará una introducción a las tecnologías HTML, CSS y JavaScript.

También se hará una introducción a la librería JQuery y se mostrarán las novedades introducidas en la versión 5 de HTML. Es importante anotar que el conocimiento completo de las tres tecnologías requiere un espacio y dedicación más amplio. Por tanto una vez apropiado este recurso el aprendiz podrá profundizar estas temáticas mediante la consulta de la bibliografía anexa o mediante investigación en otros recursos o por Internet. Las tecnologías que hacen posible las aplicaciones web están en constante evolución. Por lo anterior es importante que el desarrollador se esté constantemente actualizando en las distintas tendencias y conceptos.



El futuro digital  
es de todos

MinTIC



x  
x  
x



MAPA DE CONTENIDO



Misión  
TIC2022





# DESARROLLO DE CONTENIDOS



## 1. El lenguaje HTML

El HTML es el lenguaje de marcación principal para la creación de documentos en la WWW o Red Mundial (W3C, 2016).

El lenguaje de marcación para hipertextos o HTML utiliza un conjunto de etiquetas o “tags” para describir la estructura que tendrán las páginas web. Además del HTML existen otros lenguajes de marcación entre ellos el XML o eXtensible Markup Language. Los archivos .html son planos, es decir, pueden ser leídos por un editor de texto.

Una vez creado un archivo html válido este puede ser interpretado por un analizador sintáctico o “parser” que está incluido en los browsers o navegadores.

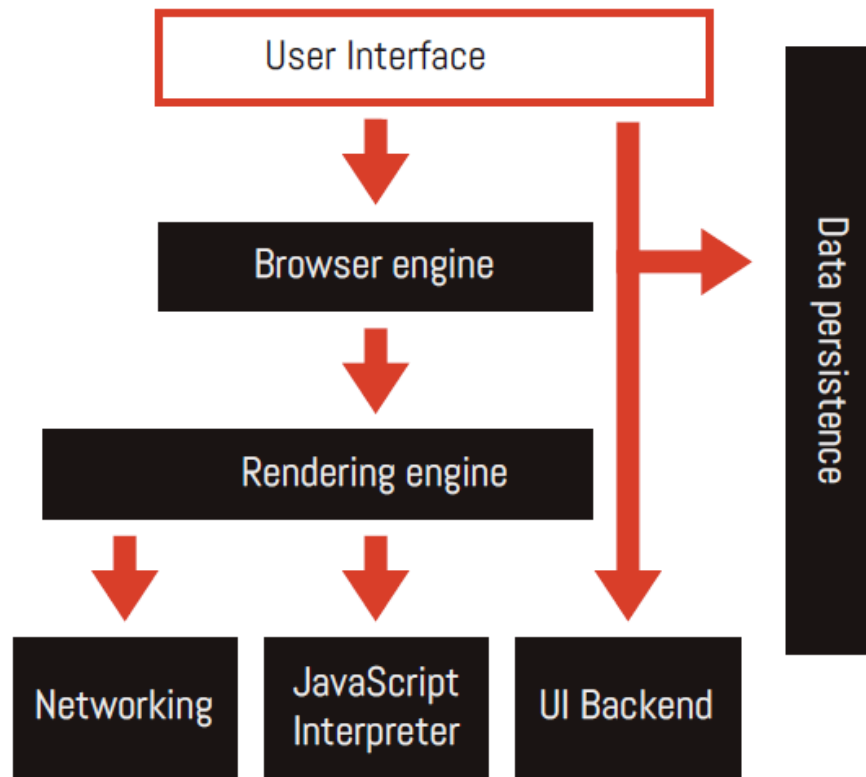
### 1.1 Estructura de un browser o navegador.

Los navegadores modernos interpretan el código escrito en HTML y CSS y lo transforman en órdenes que son transmitidas a los distintos recursos que maneja un computador como son su interfaz gráfica, su componente de redes o su capa de almacenamiento permanente (NIEDERST, 2012).



x  
x  
x

En la figura 1.2 se aprecia una arquitectura de un navegador moderno .



**Figura 1.2. Arquitectura de un navegador.**

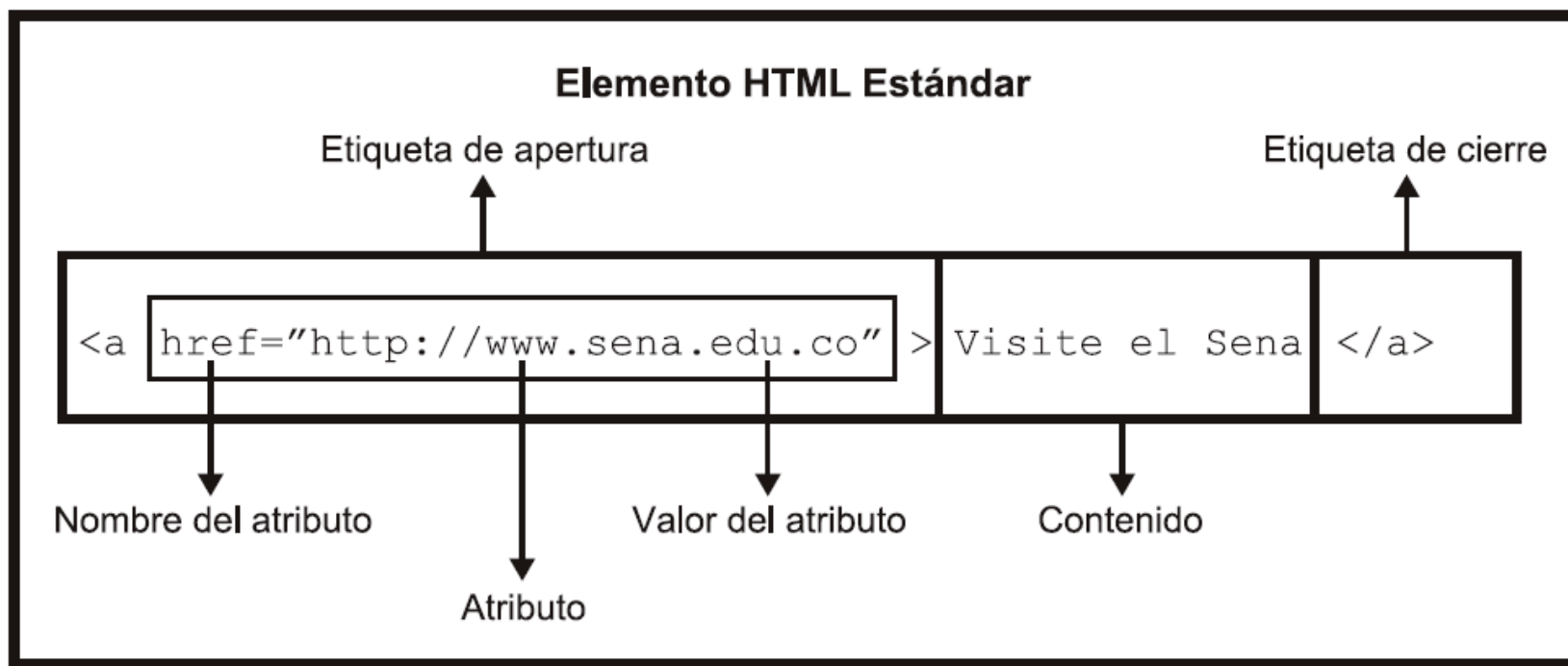


- a)** Interfaz de usuario (user interface): incluye la barra de direcciones, los botones de “adelante” y “atrás”, el menú de bookmarks. En resumen todo lo que se muestra en el navegador excepto donde se dibujan las páginas.
- b)** El motor del browser (browser engine): coordina todos los llamados de la interfaz de usuario y el motor de dibujo (rendering engine).
- c)** El motor de dibujo: responsable por dibujar la página solicitada. Si la página solicitada está en HTML, el motor de dibujo analiza el código HTML y CSS. Luego dibuja el contenido analizado en la pantalla.
- d)** Redes (Networking): para hacer solicitudes de redes tales como solicitudes HTTP.
- e)** Soporte de Interfaz de Usuario (UI Backend): usada para dibujar figuras geométricas básicas como combo boxes y ventanas. Este soporte es independiente de la plataforma (Windows, Linux, Mac). Por debajo se usan los métodos propios de la plataforma usada.
- f)** Intérprete de Javascript: usado para procesar y ejecutar código en lenguaje Javascript.
- g)** Almacenamiento de datos (Data Persistence): es la capa de persistencia. El browser puede necesitar almacenar muchos tipos de datos tales como las cookies.

Los browsers también soportan otros mecanismos de almacenamiento tales como localStorage, IndexedDB, WebSQL y filesystem.



La figura 1.3a muestra la anatomía de una etiqueta HTML estándar.

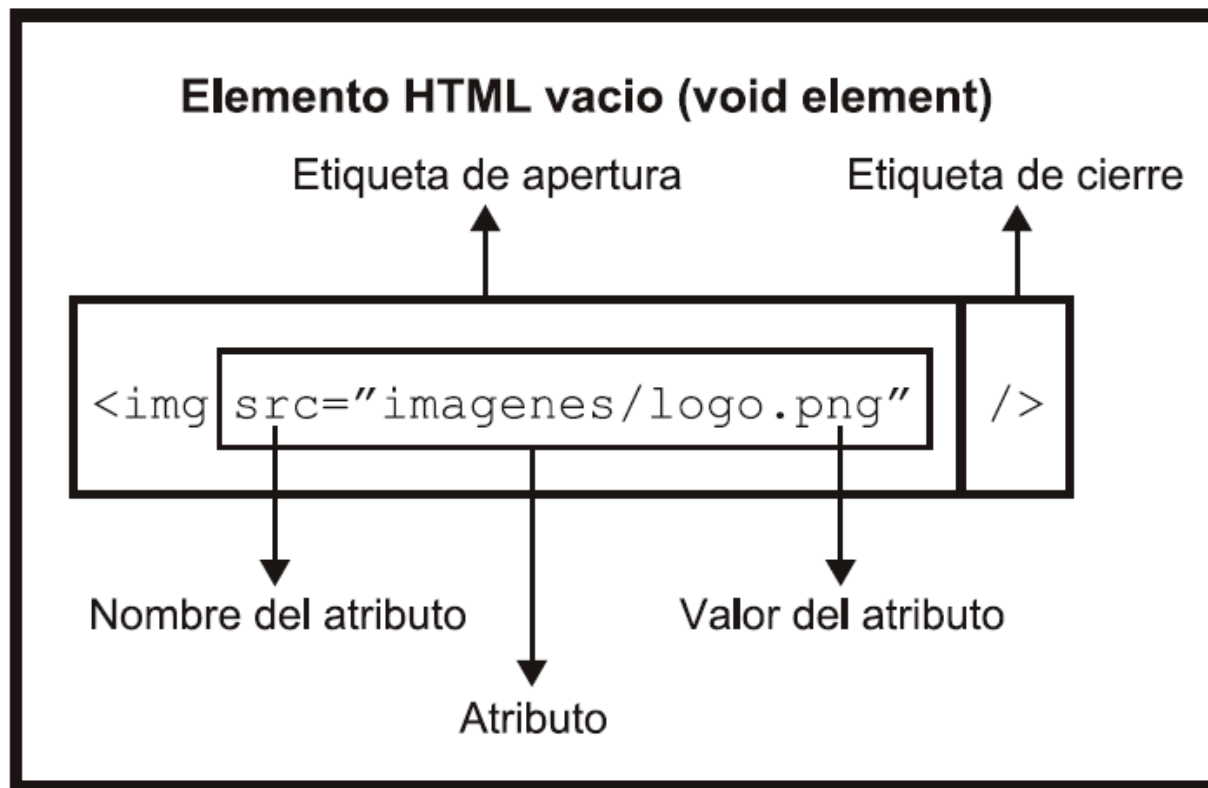


**Figura 1.3a. Anatomía de una etiqueta en HTML estándar.**





Algunos tags del lenguaje HTML no requieren tag final y se denominan etiquetas vacías.



**Figura 1.3b. Anatomía de una etiqueta en HTML vacía.**



Los tag vacíos son: area, base, br, col, embed, hr, img, input, keygen, link, menuitem, meta, param, source, track y wbr.

A continuación un ejemplo sencillo de una página HTML básica.

```
1 <html>
2     <body>
3         <b1> ¡¡hola mundo!!
4     </body>
5 </html>
```

**Figura 1.4. Ejemplo de código HTML.**

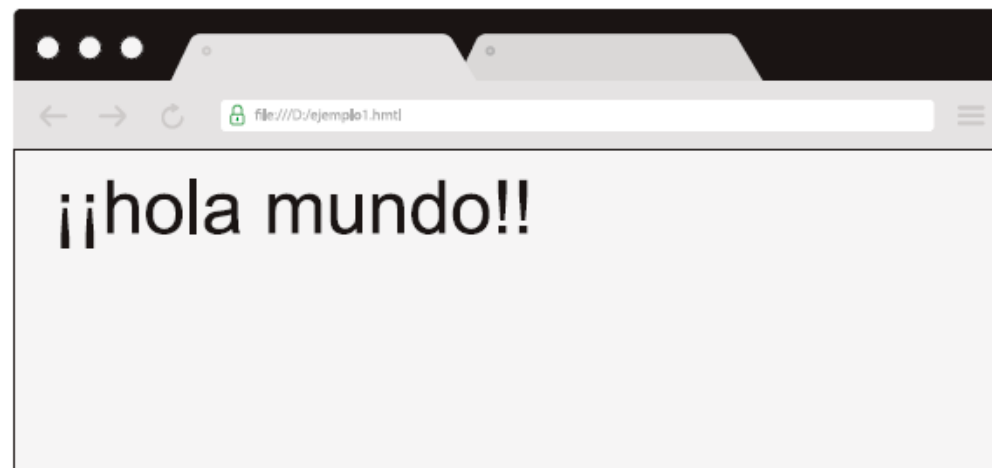


El futuro digital  
es de todos

MinTIC



Al guardar este código como un archivo con extensión html se obtiene lo siguiente cuando se abre con un navegador:



***Figura 1.5. Página web de ejemplo***

La estructura básica de un archivo HTML contiene lo siguiente:



Etiqueta	Descripción
<!DOCTYPE html>	Esta etiqueta es informativa e indica al navegador que el tipo de documento es html5.
<html>	Todo el documento está contenido en esta etiqueta también llamado nodo raíz.
<head>	La sección <head> provee información sobre el documento en sí mismo. También se usa para invocar archivos externos Javascript o de estilos. Esta sección no se dibuja en la pantalla.
<meta charset="utf-8">	La etiqueta <meta> se usa para introducir información acerca del documento. En este caso se está indicando que use la codificación de caracteres 'utf-8' que permite visualizar caracteres en español, entre otros.
<title>Título aquí </title>	La etiqueta <title> define un título para la página que aparecerá en el borde superior de la página.
</head>	Esta etiqueta indica el final del bloque <head>.
<body>	La sección <body> es el cuerpo de la página y contiene las etiquetas que se visualizarán o dibujarán en la pantalla del navegador.
Contenido de la página	
</body>	Esta etiqueta indica el final del bloque <body>
</html>	Por último esta etiqueta indica el final del documento html.

**Figura 1.6. Estructura básica de una página HTML.**



La especificación oficial y completa de los elementos que componen el lenguaje HTML están disponibles en <https://www.w3.org/TR/html5/>

A continuación se enunciarán algunas de las más importantes (W3C, 2016).

### 1.2.1 Etiquetas básicas

Etiqueta	Descripción	Etiqueta	Descripción
<code>&lt;h1&gt;&lt;/h1&gt;</code>	Encabezados Grande	<code>&lt;ul&gt;&lt;/ul&gt;</code>	Define una lista de elementos sin importar su orden
<code>&lt;h2&gt;&lt;/h2&gt;</code>	Encabezados Mediano	<code>&lt;ol&gt;&lt;/ol&gt;</code>	Define una lista de elementos donde importa el orden de los mismos.
<code>&lt;h3&gt;&lt;/h3&gt;</code>	Encabezados pequeño	<code>&lt;li&gt;&lt;/li&gt;</code>	Representa un item de una lista tipo <code>&lt;ul&gt;</code> o <code>&lt;ol&gt;</code>
<code>&lt;p&gt;</code>	Define un párrafo de texto	<code>&lt;strong&gt;&lt;/strong&gt;</code>	Para resaltar texto importante.
<code>&lt;pre&gt;&lt;/pre&gt;</code>	Define un párrafo de texto ya formateado	<code>&lt;em&gt;&lt;/em&gt;</code>	Para enfatizar un contenido.
<code>&lt;br&gt;</code>	Salto de línea	<code>&lt;!-- --&gt;</code>	Se usan para realizar comentarios.





Un ejemplo para las etiquetas anteriores.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Ejemplo con etiquetas básicas </title>
6   </head>
7   <body>
8     <h1>Etiquetas HTML</h1>
9     <p>
10      este ejemplo muestra como combinar algunas de las etiquetas más básicas de HTML5. <br>Recuerde
11      que <strong>es importante entender la diferencia entre ellas</strong>
12    </p>
13    <h2>Etiqueta ul</h2>
14    <p>
15      si hacemos una lista de mercado donde el orden no es importante, debemos usar <ul></ul>
16    </p>
17    <ul>
18      <li>Piña</li>
19      <li>Manzanas</li>
20      <li>Uchuva</li>
21      <li>Alvas</li>
22    </ul>
23    <h2>Etiqueta ol</h2>
24    <p>
25      En el caso de que estemos listando elementos donde el orden es importante, <br>con o por ejemplo
26      la clasificación mundial de tenistas, debemos usar <ol></ol>
27    </p>
28    <ol>
29      <li>Andy Murray</li>
30      <li>Rafael Nadal</li>
31      <li>Stan Wawrinka</li>
32      <li>Novak Djokovic</li>
33    </ol>
34  </body>
35 </html>
```

Figura 1.7. Ejemplo de utilización de etiquetas básicas.



Al visualizar la página aparece lo siguiente:



## Etiquetas HTML

este ejemplo muestra como combinar algunas de las etiquetas más básicas de HTML5.  
Recuerde que es importante entender la diferencia entre ellas

### Etiqueta `ul+li`

si hacemos una lista de mercado donde el orden no es importante, debemos usar `ul`

- Piña
- Manzanas
- Uchuva
- Uvas

### Etiqueta `ol+li`

En el caso de que estemos listando elementos donde el orden es importante, como por ejemplo la clasificación mundial de tenistas, debemos usar `ol`

1. Andy Murray
2. Rafael Nadal
3. Stan Wawrinka
4. Novak Djokovic

**Figura 1.8. Página web de ejemplo con etiquetas básicas.**



### 1.2.2 Etiquetas para definir bloques o contenedores.

- El elemento `<div>` se usa para definir agrupamientos lógicos dentro de un página HTML. Estos agrupamientos son útiles ya que facilitan el manejo de los elementos agrupados o contenidos en el bloque al momento de asignar estilos o agregar dinámicamente nuevos nodos hijos (W3C, 2016).

El elemento `<span>` también sirve para agrupar elementos pero aplica dentro de una misma línea. Es útil para el manejo de estilos dentro de una misma línea de texto.

### 1.2.3. Etiquetas para elaborar tablas.

Tag	Descripción	Tag	Descripcion
<code>&lt;table&gt; &lt;/table&gt;</code>	Define una tabla	<code>&lt;tr&gt;&lt;/tr&gt;</code>	Define una fila
<code>&lt;thead&gt;&lt;/thead&gt;</code>	Define los elementos que conforman el encabezado de la tabla	<code>&lt;td&gt;&lt;/td&gt;</code>	Define un campo
<code>&lt;tbody&gt; &lt;/tbody&gt;</code>	Define los elementos que componen el cuerpo de la tabla	<code>&lt;th&gt;&lt;/th&gt;</code>	Define un encabezado de una columna

Para la elaboración de tablas se utilizan las siguientes etiquetas (W3C, 2016):



× A continuación un ejemplo de una tabla definida en lenguaje HTML:  
×  
×

```
1  <html>
2    <head>
3      <meta charset="utf-8">
4      <title>Ejemplo con etiquetas básicas</title>
5    </head>
6    <body>
7      <table border="1">
8        <thead>
9          <tr>
10             <th>Meses</th>
11             <th>Días</th>
12             <th>Números</th>
13             <th>Ciudades</th>
14          </tr>
15        </thead>
16        <tbody>
17          <tr> <td>Enero</td> <td>Lunes</td> <td>Uno</td> <td>Girón</td> </tr>
18          <tr> <td>Febrero</td> <td>Martes</td> <td>Dos</td> <td>Bucaramanga</td> </tr>
19          <tr> <td>Marzo</td> <td>Miércoles</td> <td>Tres</td> <td>Florida Blanca</td> </tr>
20        </tbody>
21      </table>
22    </body>
23  </html>
```

**Figura 1.9. Ejemplo de una tabla en HTML.**





Una vez visualizada aparecerá lo siguiente:

x  
x  
x



Meses	Dias	Números	Ciudades
Enero	Lunes	Uno	Girón
Febrero	Martes	Dos	Bucaramanga
Marzo	Miércoles	Tres	Floridablanca

*Figura 1.10. Página web con una tabla.*





#### × 1.2.4 Etiquetas para el manejo de formularios.

- × Los formularios HTML son una herramienta de gran utilidad para el desarrollo de aplicaciones de bases de datos. Un formulario almacena temporalmente los campos que luego de ser validados serán enviados a la base de datos. La estructura básica del un formulario en HTML es la siguiente (W3C, 2016):

```
<form action="procesarFormulario.php">  
  Enunciado del campo:  
  <input type="text" name="campo1">  
  <input type="submit" value="Enviar">  
</form>
```

La figura 1.10 describe la estructura de un formulario.

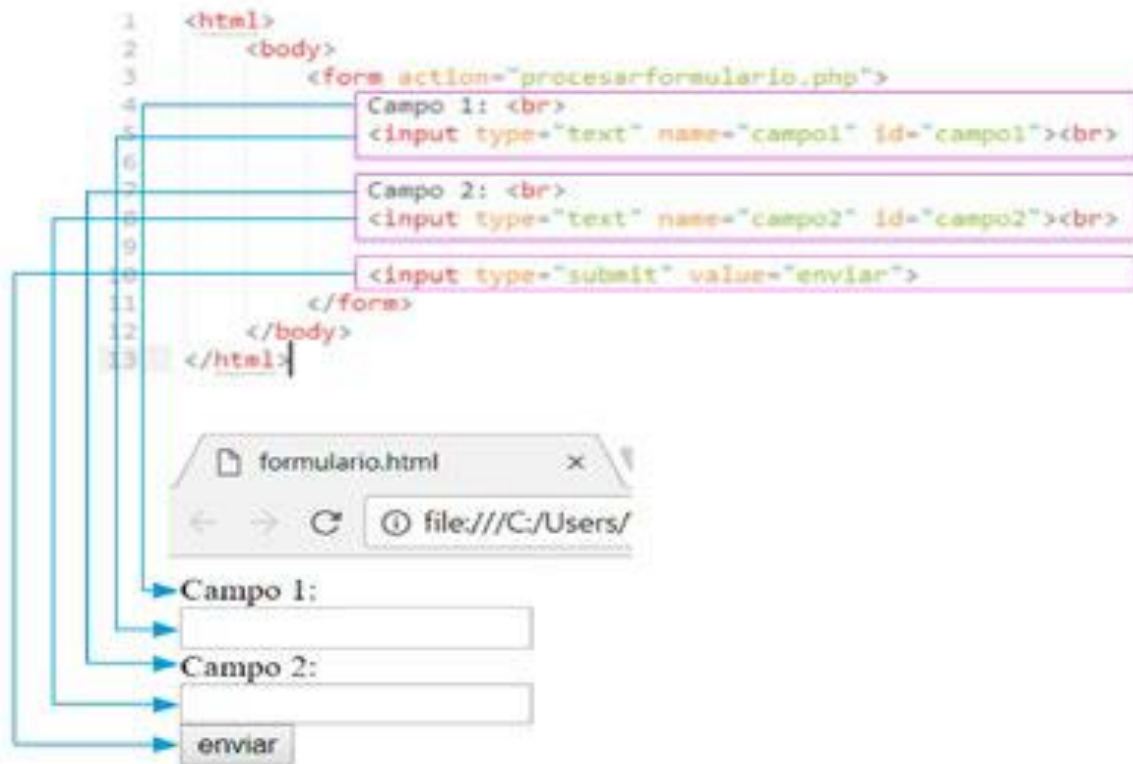


Etiqueta	Función
<form>	Inicia un bloque de formulario
<form action="programa.php">	El atributo "action" indica al navegador cual programa procesará el formulario una vez se haya oprimido el botón tipo "submit" o botón de enviar.
<input type="tipo de campo">	Define un campo del formulario. El atributo "type" indica el tipo de campo.
<input type="text">	Define un campo de tipo texto. Además de "text" existen otros 21 tipos de campos. Entre ellos están : number, date, password, hidden, color, checkbox.
<input type="text" name="nombre_de_campo">	Define un campo tipo texto y define "nombre_de_campo" como el nombre que se asociará al valor ingresado en dicho campo. Este valor es el que se envía cuando se oprime el botón de enviar.
<input type="submit">	Define un tipo de campo especial que indica al navegador que ya puede procesar el formulario
</form>	Finaliza un bloque de formulario

**Figura 1.10. Estructura de un formulario HTML**



A continuación un ejemplo de un formulario y su correspondiente representación en un navegador.



**Figura 1.11. Formulario en HTML.**



### 1.2.5 Etiqueta para el manejo de Imágenes.



- Un documento HTML puede contener una o varias imágenes las cuales se pueden incluir usando la etiqueta vacía `<img>` y su estructura es la siguiente (W3C, 2016):

```

```

Donde:

src : archivo fuente de la imagen.

alt : nombre alternativo que aparece cuando la imagen no se puede cargar

width : ancho definido

height : altura de la imágenes.

A continuación un ejemplo de una página web que incorpora una imagen.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ejemplo Imagen</title>
5   </head>
6   <body>
7     <h2>Glaciar Argentino </h2>
8     
9   </body>
10 </html>
11
```

**Figura 1.12. Ejemplo diseño web con una imagen incorporada.**



xxx

Al visualizar la página en un navegador se obtiene lo siguiente:



**Figura 1.13. Página web con una imagen.**





### 1.3. Contenido estático vs contenido dinámico.



En el diseño web existen dos tipos de contenido:

- a) Contenido estático: proviene de páginas desarrolladas y almacenadas en el sitio web.
- b) Contenido dinámico: proviene de páginas o elementos que son creados al instante por el navegador a petición de un programa.

### 1.4. Fases en el diseño de un sitio web.

Para el diseño web existen muchas metodologías. En este recurso se utilizarán los modelos de desarrollo de software vistos en la actividad de proyecto 1 en especial el modelo en cascada. Bajo este modelo de procesos aplicado al diseño web se toman las siguientes etapas: planeación, diseño, desarrollo e implantación.

**a) Planeación:** se determina el concepto de diseño que se quiere realizar. En esta etapa es importante que participe un diseñador gráfico que aporte su criterio en conceptos como: manejo de colores, elementos visuales, manejo de la marca.

**b) Diseño:** se realizan las maquetas con los distintos elementos que conforman el sitio como son: menús, formularios, encabezados, pies de página. Estas maquetas se pueden realizar en herramientas ya mencionadas como son Balsamic Mockups, Gomockingbird, entre otras. Algunos de los productos de esta etapa son: maqueta del sitio (wireframe), diagrama del sitio, entre otros.



El futuro digital  
es de todos

MinTIC



X  
X  
X

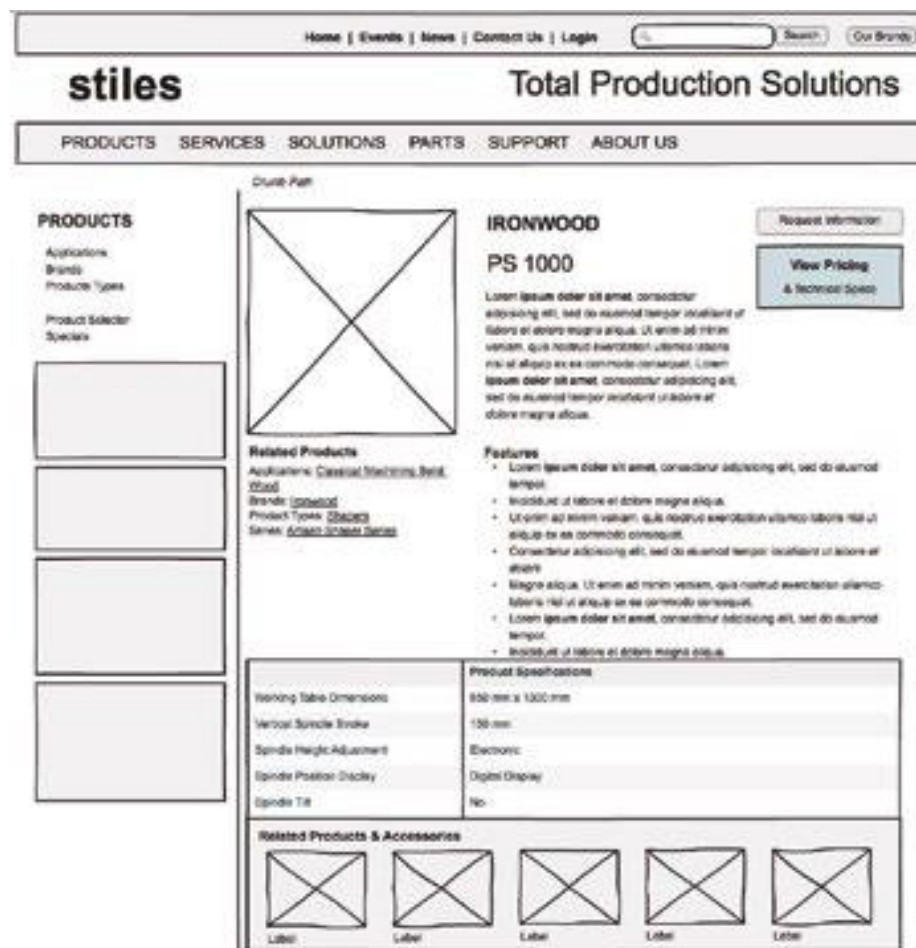


Figura 1.14: maqueta de un sitio web.



El futuro digital  
es de todos

MinTIC



- × c) **Desarrollo:** en esta fase se crean los distintos archivos html, css y javascript en un servidor de desarrollo o pruebas. También se realizan pruebas al código fuente javascript y se atiende la realimentación por parte del cliente.



**Figura 1.15. Página web en producción.**



×  
×  
×  
d) **Implantación:** en esta etapa el sitio web se instala en el servidor de producción para que entre a operar (ver figura 1.15) .

### 1.5. Herramientas para el diseño y desarrollo web.

Además de HTML, CSS y Javascript es importante que el diseñador web conozca de las siguientes herramientas:

- a) **Editor de texto:** como notepad, notepad++,
- b) **FTP:** como winSCP.
- c) **Editor de imágenes:** Gimp, Adobe
- d) **Frameworks CSS:** Como Bootstrap
- e) **Git y Github:** para administrar el código fuente.

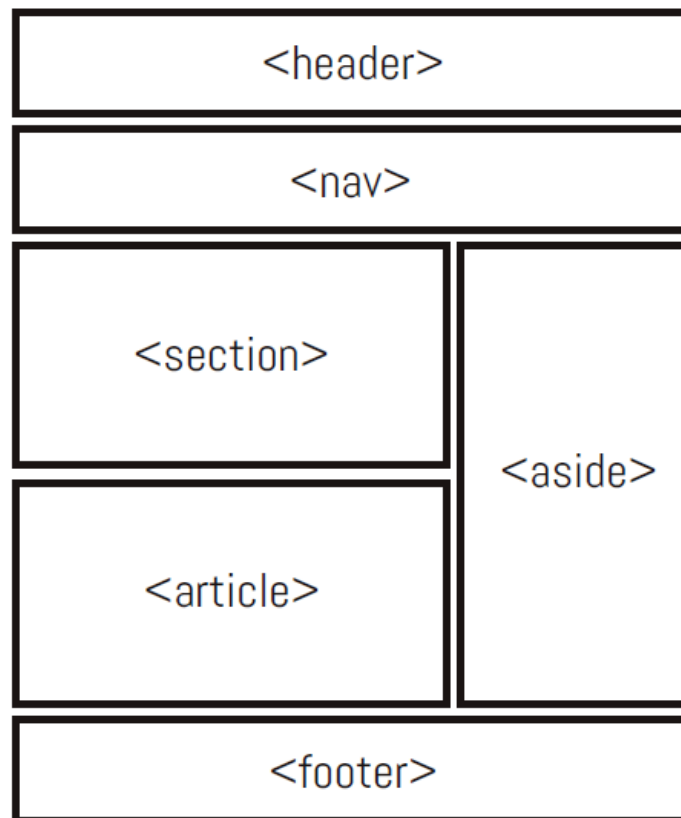
### 1.6 Maquetado (Layouts) en HTML.

La maqueta o “layout” de una página HTML define la ubicación o distribución de sus elementos dentro de la misma página.





Una distribución típica de una página web es la siguiente:



**Figura 1.16. Distribución de elementos en una página web.**





Donde:



Etiqueta	Función
<header>	Define el encabezado para un documento o sección.
<nav>	Define un contenedor para los links de navegación.
<section>	Define una sección en un documento.
<article>	Define un artículo independiente y autocontenido.
<aside>	Define un contenido "al lado" del contenedor principal.
<footer>	Define un pie de página para un documento o sección.
<details>	Define detalles adicionales.
<summary>	Es el encabezado del elemento <details>.



## 1.7 Novedades en la versión 5 de HTML.

La versión 5 del lenguaje introduce nuevos elementos que se relacionan más adelante, sin embargo, es importante anotar que HTML5 se refiere a un conjunto de nuevas características que se pueden utilizar individualmente si están soportadas por el navegador (NIEDERST, 2012) .

Para determinar si un navegador soporta una característica de HTML5 se utiliza una librería en Javascript llamada “Modernizr”. De esta forma un desarrollador puede hacer su código compatible tanto para nuevos navegadores como para versiones anteriores de los mismos.

### 1.7.1. El elemento <canvas>

Es un “lienzo” digital sobre el cual se pueden hacer dibujos en formato de mapa de bits usando el lenguaje Javascript.



### 1.7.2. Video.

- x HTML5 define un nuevo elemento llamado <video> para incorporar video en las páginas web de forma nativa. Anteriormente los videos se introducían a las páginas web usando productos de terceros como Adobe Flash, Microsoft ActiveX o Apple Quicktime. Con el tiempo esos productos de terceros generaban problemas de incompatibilidades entre plataformas, versiones, etc.

### 1.7.3. Almacenamiento local.

El almacenamiento local provisto por HTML5 permite al sitio web almacenar datos en el computador para consultarlos posteriormente. Este sistema es parecido a las “cookies” pero fue diseñado para manejar mayor cantidad de información. Por otra parte las “cookies” son enviadas al servidor cada vez que se consulta una página lo cual incrementa el tráfico de la red.

### 1.7.4. Web Workers.

Los “web workers” permiten a los navegadores ejecutar código Javascript en background. Lo anterior quiere decir que un “web worker” hace uso de las capacidades multiproceso de los procesadores modernos para ejecutar tareas de manera simultánea con las tareas normales de un navegador.



### 1.7.5. Aplicaciones web fuera de línea.

- × Además de ejecutar páginas web en línea una nueva característica del HTML5 permite ejecutar sitios web aún estando fuera de línea. Para que esta característica funcione se requiere que el navegador visite el sitio y solicite todos los archivos requeridos para su ejecución fuera de línea. Posteriormente cuando el navegador se encuentre sin conexión puede seguir navegando en la copia almacenada.

Una vez se restablezca la conexión se actualizan los archivos almacenados y el navegador sigue funcionando en línea.

### 1.7.6. Geolocalización.

Es una característica que permite determinar la localización del computador que está ejecutando el browser. El objeto de geolocalización suministrará las coordenadas del sitio pero previamente solicitará autorización al usuario.





### 1.7.7. Nuevos tipos de cajones para formularios.

- × HTML5 define los siguientes nuevos “input types”:

Etiqueta	Descripción
<input type="search">	Utilizada para textos de búsqueda.
<input type="number">	Utilizada para capturar números.
<input type="range">	Utilizada para capturar número en un rango específico.
<input type="color">	Utilizada para capturar el código de un color en particular.
<input type="tel">	Utilizada para capturar números de teléfono.
<input type="url">	Utilizada para capturar direcciones de internet.
<input type="email">	Utilizada para capturar direcciones de correo.
<input type="date">	Utilizada para capturar fechas.
<input type="month">	Utilizada para capturar un mes en particular.
<input type="week">	Utilizada para capturar una semana del año en particular.
<input type="time">	Utilizada para capturar una hora del día.
<input type="datetime">	Utilizada para capturar una fecha y hora en formato de máquina.
<input type="datetime-local">	Utilizada para capturar una fecha y hora en formato local.



También se introducen los siguientes restricciones que se aplican a los contenidos en los campos tipo <input>:

Atributo	Valor esperado	Descripción
placeholder	Texto	Muestra una sugerencia en el cajón para el texto esperado.
required	No aplica	Valida que el campo no se envíe vacío.
pattern	Expresión regular	Permite validar el texto ingresado usando una expresión regular.
min, max	Número	Especifica el valor mínimo y máximo que pueden tomar los valores numéricos.
step	Número	Especifica los intervalos permitidos para el campo numérico.

### 1.7.8. Nuevas etiquetas introducidas en la versión 5 del lenguaje HTML.

Además de las etiquetas header, nav, section, aside, footer, con las cuales se define la estructura básica de una página web con HTML5, a continuación, se presentan nuevas etiquetas a utilizar dentro del contenido de la página web:



Etiqueta	Descripción
<code>&lt;article&gt; &lt;/article&gt;</code>	Define unidades independientes de contenido dentro del elemento <code>&lt;section&gt;</code> .
<code>&lt;hgroup&gt;</code> <code>&lt;h1&gt;titulo&lt;/h1&gt;</code> <code>&lt;h2&gt;subtitulo&lt;/h2&gt;</code> <code>&lt;/hgroup&gt;</code>	Etiqueta usada dentro de la sección header cuando esta contiene dos o más elementos de tipo h como h1, h2, h3. Su objetivo es ayudar al navegador a interpretar y procesar correctamente la página Web.
<code>&lt;mark&gt;</code>	Permite hacer énfasis en una palabra o frase.
<code>&lt;small&gt;</code>	Muestra una palabra o frase en letra pequeña con fuente de información legal.
<code>&lt;address&gt;</code>	Para definir información de contacto, normalmente usado dentro de la sección <code>&lt;footer&gt;</code> .
<code>&lt;time&gt;&lt;/time&gt;</code>	Permite formatear campos de hora para fácil entendimiento.



También se introducen los siguientes restricciones que se aplican a los contenidos en los campos tipo `<input>`:



Etiqueta	Descripción
<code>&lt;audio&gt;</code>  Forma 1: <code>&lt;audio src="miAudio.mp3"&gt;&lt;/audio&gt;</code>  Forma 2: <code>&lt;audio&gt;</code> <code>&lt;source src="miAudio.mp3"&gt;</code> <code>&lt;source src="miAudio.ogg"&gt;</code> <code>&lt;/audio&gt;</code>	Usada para reproducir archivos de audio con extensión ogg o mp3. Dependiendo del navegador podrá reproducir un tipo de archivo u otro, por esta razón es recomendable incluir ambos archivos y especificar las dos posibles fuentes con la etiqueta interna source, para que el navegador procese el archivo apropiado.  Atributos.  src: url o nombre archivo a reproducir. controls: presenta los controles de reproducción. autoplay: reproducción automática del audio. loop: reproducciones infinitas del audio.
<code>&lt;video&gt;</code>  Forma 1: <code>&lt;video src="miVideo.mp4"&gt;&lt;/video&gt;</code>  Forma 2: <code>&lt;video&gt;</code> <code>&lt;source src="miVideo.mp4"&gt;</code> <code>&lt;source src="miVideo.ogg"&gt;</code> <code>&lt;/video&gt;</code>	Define unidades independientes de video con extensión ogg o mp4. Dependiendo del navegador podrá reproducir un tipo de archivo u otro, por esta razón es recomendable incluir ambos archivos y especificar las dos posibles fuentes con la etiqueta interna source, para que el navegador procese el archivo apropiado.  Atributos.  src: url o nombre archivo de video a reproducir. controls: presenta los controles de reproducción. autoplay: reproducción automática del video. loop: reproducciones infinitas del video. poster: provee una imagen que será mostrada.





El futuro digital  
es de todos

MinTIC



# BIBLIOGRAFÍA



Niederst, J. (2012). Learning Web Design. Cambridge. O'Reilly, Fourth Edition.

W3C The World Wide Web Consortium. (2016). HTML 5.1

W3C Recommendation. Recuperado de <https://www.w3.org/TR/html>

W3C The World Wide Web Consortium. (2017).

CSS Snapshot 2017. Recuperado de <https://www.w3.org/TR/CSS>

Introducción al Diseño WEB – HTML. (2021). Servicio Nacional de Aprendizaje SENA (Creative Commons).