

**VII Cuatrimestre
Ingeniería en Sistemas Computacionales**

**PROGRAMACIÓN WEB
Profr. Miguel Ángel Piza Larios**

Juan Hugo Reyes Pérez

**RESUMEN UNIDAD 2
Lenguaje de marcado**



17 de septiembre de 2020

ÍNDICE

1.	Lenguaje de marcado	3
1.1	Introducción	3
1.2	Representación de documentos	3
1.3	Tipos de datos básicos	4
1.4	Estructura global de un documento.....	6
1.5	Elementos básicos: texto, vínculos, listas, tablas, objetos, imágenes y aplicaciones.....	7
1.6	Estructura y disposición	15
1.7	Formularios.....	19

ÍNDICE DE IMÁGENES

Figura 1	Primer ejemplo de HTML5.....	6
Figura 2	Los elementos de encabezado h1 a h6.....	8
Figura 3.	Vinculación a otras páginas web.	8
Figura 4.	Creación de un vínculo a una dirección de correo electrónico.....	9
Figura 5.	Inclusión de imágenes en documentos HTML5.....	10
Figura 6.	Imágenes como anclas de vínculos.	11
Figura 7.	Lista desordenada que contiene hipervínculos.	12
Figura 8.	Listas anidadas y listas desordenadas.	13
Figura 9.	Una tabla de HTML5 simple.....	14
Figura 10.	Representación visual de un clásico diseño web.....	15
Figura 11.	Representación visual de un clásico diseño para blogs.....	16
Figura 12.	Representación visual de un diseño utilizando elementos HTML5.	16
Figura 13.	Página con función JavaScript que dibuja rectángulos dentro de <canvas>.	18
Figura 14.	Formulario con controles clásicos.	19

1. Lenguaje de marcado

1.1 Introducción

A diferencia de otros lenguajes de programación como C, C++, C#, Java y Visual Basic, HTML5 es un lenguaje de marcado que especifica la estructura y el contenido de los documentos que se visualizan en los navegadores Web. HTML5 es la versión emergente de HTML. Este lenguaje nos permite crear contenido que se despliega de manera apropiada a través de todo el extraordinario rango de dispositivos conectados a Internet: smartphones, tabletas, computadoras tipo notebook, computadoras de escritorio, dispositivos de propósito especial como pantallas gigantes en auditorios de conciertos y estadios deportivos.

Aunque similares, cada versión de HTML utiliza etiquetas diferentes, ya que en sí mismo el HTML5 tiene objetivos más ambiciosos de los que pretendían cubrir versiones anteriores y tiene su propio y más extenso juego de etiquetas.

HTML4 es ampliamente aceptado y su implementación está prácticamente en todos los navegadores. HTML4 es permisivo con pequeños errores de código, aunque esto puede verse como una desventaja. Si la compatibilidad entre navegadores es un requerimiento primordial puede ser la mejor opción. Sin embargo, aunque las funcionalidades propietarias añadidas a lo largo de los años han creado cierta flexibilidad, no son comparables con las de HTML5. Las capacidades para formularios son pocas y limitadas. La mayoría de los navegadores soportan alguna forma de JavaScript, lo que permite mayor dinamismo, pero las implementaciones siguen teniendo amplias diferencias en algunos casos.

XHTML es un estándar estricto basado en el HTML4 y XML, por lo que su implementación es también casi universal. No es permisivo con los errores, como sí lo es HTML4, lo que obliga a formar correctamente un documento HTML como sucede con un documento XML. Las desventajas que presenta son las mismas que HTML4.

Por su parte, HTML5 posee una gran cantidad de funcionalidades que los desarrolladores sólo lograban con el uso de algún plug-in de terceros como Applets de Java o Flash embebidos en el código. Las mejoras en el manejo de multimedia son ampliamente superiores; imágenes, video, audio y fuentes de texto son mucho más manipulables (Herrera, 2012). En muchos casos, estas mejoras se pueden tomar como una respuesta directa a Flash. La principal desventaja es que HTML5 es tan nuevo que, quizás, algunas personas todavía utilicen navegadores que no pueden visualizarlo. HTML5 nos proporciona una manera de hacer un código más limpio, más fácil de leer y de escribir, cubriendo al mismo tiempo y de mejor manera la cada vez mayor demanda de funcionalidades por parte de programadores, diseñadores y usuarios.

1.2 Representación de documentos

Tal como lo dice la especificación de la W3C, para promover la interoperabilidad se requiere que cada aplicación especifique su conjunto de caracteres del documento. Un conjunto de caracteres del documento consiste en:

- Repertorio: Conjunto de caracteres abstractos.
- Posiciones de códigos: Conjunto de referencias enteras a los caracteres del repertorio.

Cada documento HTML es una secuencia de caracteres del repertorio. Los sistemas informáticos identifican cada carácter según la posición de su código; por ejemplo, en el

conjunto de caracteres ASCII, las posiciones de los códigos 65, 66 y 67 se refieren a los caracteres 'A', 'B' y 'C' respectivamente.

El conjunto de caracteres ASCII no es suficiente para un sistema de información global como la Web, de modo que HTML usa un conjunto de caracteres mucho más completo llamado Conjunto Universal de Caracteres (Universal Character Set, UCS), definido en el estándar ISO/IEC 10646 (ISO, 2014). Este estándar define un repertorio de miles de caracteres usados por comunidades de todo el mundo.

Codificación de caracteres

El parámetro "charset" identifica una codificación de caracteres, que es un método para convertir una secuencia de bytes en una secuencia de caracteres. Esta conversión se adapta naturalmente al modo de funcionamiento de la Web: los servidores envían los documentos HTML a los agentes de usuario como un flujo de bytes; los agentes de usuario los interpretan como un flujo de caracteres. El método de conversión puede ir desde una correspondencia simple uno a uno hasta complejos algoritmos o procesos de intercambio.

Para tener en cuenta las limitaciones de algunos servidores o de sus configuraciones, los documentos HTML pueden incluir información explícita sobre la codificación de caracteres del documento; el elemento META puede utilizarse para proporcionar esta información a los agentes de usuario.

La declaración META sólo debe utilizarse cuando la codificación de caracteres se organice de modo que los bytes con valores ASCII correspondan a caracteres ASCII (al menos hasta que se interprete el elemento META). Las declaraciones META deberían aparecer lo antes posible dentro del elemento HEAD.

En resumen, los agentes de usuario conformes deben tener en cuenta las siguientes prioridades (de prioridad más alta a más baja):

- Un parámetro "charset" HTTP en un campo "Content-Type".
- Una declaración META con "http-equiv" establecido en "Content-Type" y un valor dado para "charset".
- El atributo charset establecido en un elemento que designe un recurso externo.

1.3 Tipos de datos básicos

El SGML es un sistema para definir lenguajes para dar formato a documentos (markup languages). Los autores utilizan para representar información estructural, de presentación y semántica junto con el contenido. El HTML es un ejemplo de lenguaje de formato de documentos.

Elementos

Una definición del tipo de documento SGML declara tipos de elementos que representan estructuras o un comportamiento deseado. HTML incluye elementos que representan párrafos, vínculos de hipertexto, listas, tablas, imágenes, etc. Cada declaración de tipo de elemento describe generalmente tres partes: una etiqueta inicial, un contenido y una etiqueta final. El nombre del elemento aparece en la etiqueta inicial (escrita <nombre-del-elemento>) y en la etiqueta final (escrita </nombre-del-elemento>); obsérvese la barra que precede al

nombre del elemento en la etiqueta final. Por ejemplo, la etiqueta inicial y final del tipo de elemento `ul` delimitan los objetos de una lista:

Algunos tipos de elemento de HTML permiten omitir las etiquetas finales; también permiten omitir la etiqueta inicial; por ejemplo, `head` y `body`. El DTD de HTML indica para cada tipo de elemento si la etiqueta inicial y la etiqueta final son necesarias. Algunos tipos de elemento de HTML no tienen contenido. Por ejemplo, el elemento de salto de línea `br` no tiene contenido; su único papel es terminar una línea de texto. Estos elementos vacíos nunca tienen etiquetas finales. La definición del tipo de documento y el texto de la especificación indican si un tipo de elemento es vacío (es decir, si no tiene contenido) o, en caso de que pueda tener contenido, lo que se considera como contenido legal. Para los nombres de los elementos no se distingue nunca entre mayúsculas y minúsculas.

Los elementos no son etiquetas. Algunas personas se refieren a los elementos como etiquetas (por ejemplo, "la etiqueta `P`"). Recuerde que el elemento es una cosa, y la etiqueta (sea la inicial o la final) es otra cosa distinta.

Atributos

Los elementos pueden tener propiedades asociadas, llamadas atributos, que pueden tener valores (por defecto o asignados por el autor o por un script). Las parejas atributo/valor aparecen antes del `>` final de la etiqueta inicial de un elemento. En la etiqueta inicial de un elemento puede aparecer cualquier número de parejas (legales) atributo/valor, separadas por espacios. Pueden aparecer en cualquier orden.

Recomendamos usar comillas incluso, cuando sea posible eliminarlas. Para los nombres de atributos no se distingue nunca entre mayúsculas y minúsculas. Para los valores de los atributos en general no se distingue entre mayúsculas y minúsculas.

Referencias de caracteres

Las referencias de caracteres son nombres simbólicos o numéricos de caracteres que pueden incluirse en un documento HTML. Son útiles para hacer referencia a caracteres usados esporádicamente o a aquellos que son difíciles de introducir con las herramientas de creación. Las referencias de caracteres comienzan con un signo `&` y terminan con un punto y coma (;). A continuación, figuran algunos ejemplos comunes:

- `<` representa el signo `<`.
- `>` representa el signo `>`.
- `"` representa el signo `"`.
- `å` (en decimal) representa la letra "a" con un pequeño círculo encima.
- `И` (en decimal) representa la letra mayúscula cirílica "I".
- `水` (en hexadecimal) representa el carácter chino para el agua.

Comentarios

No se permite espacio en blanco entre el delimitador de apertura de declaración de etiqueta (`<!`) y el delimitador de apertura de comentario (`--`), pero sí se permite entre el delimitador de cierre de comentario (`--`) y el delimitador de cierre de declaración de etiqueta (`>`). Un error común es incluir una cadena de guiones (`---`) dentro de un comentario. Los autores deberían evitar poner dos o más guiones adyacentes dentro de un comentario.

1.4 Estructura global de un documento

Un documento HTML5 se compone de tres partes:

1. Una línea que contiene la versión del HTML.
2. Una sección de cabecera declarativa.
3. Un cuerpo que incluye el contenido real del documento.



Figura 1 Primer ejemplo de HTML5.

Declaración de tipo. El fijar un estándar siempre ha sido un problema y una larga cadena de tensión e inconvenientes ha rodeado al HTML como estándar a lo largo de su implementación entre navegadores. Como solución surgió la declaración del tipo de documento DOCTYPE. Con esta etiqueta se pueden establecer hasta hoy 15 modos de estándar diferentes, pero la expuesta es la más corta, amena y simple que proporciona HTML5 (Standard Mode). Algunos navegadores operan en modo quirks para mantener compatibilidad hacia atrás con páginas web que no están actualizadas con los estándares más recientes. En cada documento de HTML5 que se cree, debe incluir la declaración DOCTYPE.

Líneas en blanco. Las líneas en blanco (líneas 2 y 10) se introdujeron para facilitar la legibilidad de los documentos; el navegador ignora estas líneas.

Comentarios. Las líneas 3 y 4 son comentarios en HTML5. Se insertan opcionalmente para mejorar la legibilidad y describir el contenido de un documento. Los comentarios en HTML5 empiezan con `<!--` y terminan con `-->`.

Elementos html, head y body. El marcado de HTML5 contiene elementos que representan el contenido de un documento, por ejemplo, texto, imágenes, gráficos, animaciones, sonidos y videos. Además, contiene elementos que especifican la estructura y el significado de un documento. Algunos elementos importantes son html, que empieza en la línea 5 y termina en la línea 14; head, líneas 6 a la 9; y body, líneas 11 a la 13. El elemento html encierra la sección del encabezado, representada por el elemento head, y la sección del cuerpo, representada por el elemento body. La sección del encabezado contiene información sobre el documento HTML5, como el conjunto de caracteres que utiliza la página, línea 7; en este caso se trata de

UTF-8 el cual es el esquema de codificación de caracteres más popular para la Web. Además, en la línea 8 aparece el elemento title, el cual es el título de la página web.

Etiquetas de inicio y de fin. Los documentos de HTML5 delimitan la mayoría de los elementos con una etiqueta inicial y una etiqueta final. Una etiqueta inicial consiste en el nombre del elemento entre paréntesis angulares, por ejemplo <html> en la línea 5. Una etiqueta final consiste en el nombre del elemento seguido de una barra diagonal / entre paréntesis angulares, por ejemplo </html> en la línea 14. También existen los elementos vacíos que no tienen etiqueta final.

Elemento title. La línea 8 especifica un elemento title. A esto se le conoce como elemento anidado, ya que está encerrado entre las etiquetas inicial y de final del elemento head. El elemento head también es un elemento anidado. El elemento title describe la página web. Por lo general, los títulos aparecen en la barra de título en la parte superior de la ventana del navegador, en la pestaña del navegador en donde se visualiza la página y también como el texto que identifica a una página cuando los usuarios agregan esa página a su lista de Favoritos o Marcadores. En la línea 11 comienza el elemento body del documento, el cual especifica su contenido y puede incluir texto, imágenes, sonidos y videos.

Elemento párrafo. Algunos elementos, como párrafo, que se indica mediante las etiquetas <p> y </p> en la línea 12, ayudan a definir la estructura de un documento. Todo el texto que se coloca entre las etiquetas <p> y </p> forma un párrafo. Cuando un navegador despliega un párrafo, coloca espacio adicional por encima y debajo del texto del párrafo. La línea clave en el programa es la línea 12, que indica al navegador que muestre el mensaje Bienvenido a HTML5.

Etiquetas de fin. Este documento termina con dos etiquetas de fin (líneas 13 y 14) que cierran los elementos body y html, respectivamente. La etiqueta </html> informa al navegador que el marcado de HTML5 está completo.

1.5 Elementos básicos: texto, vínculos, listas, tablas, objetos, imágenes y aplicaciones

Texto

Para incluir texto en un documento HTML5 es <p> </p>, cuando se tiene cierto texto más importante que otro. HTML5 ofrece seis elementos de encabezado, h1 a h6, para especificar la importancia relativa de la información. El elemento de encabezado h1 (línea 12) se considera el más significativo y por lo general se despliega en una fuente más grande que los otros (líneas 13 a 17). Cada elemento de encabezado sucesivo (h2, h3, etc.) se despliega con una fuente cada vez más pequeña.

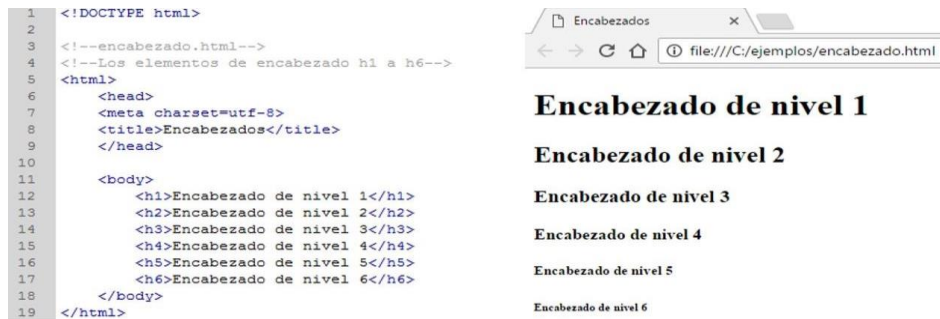


Figura 2 Los elementos de encabezado h1 a h6.

Vínculos

Cuando un usuario hace clic en un hipervínculo, el navegador trata de ejecutar una acción asociada con él. Por lo general, los hipervínculos de texto se subrayan y colorean en azul. En la Figura 3 el texto í es el código especial para acentuar letras, como en la línea 8.



Figura 3. Vinculación a otras páginas web.

En la línea 13 se introduce el elemento strong, el cual indica que su contenido tiene alta importancia. En la línea 16 se define un hipervínculo hacia el URL asignado al atributo href (referencia de hipertexto), el cual especifica la ubicación de un recurso tal como:

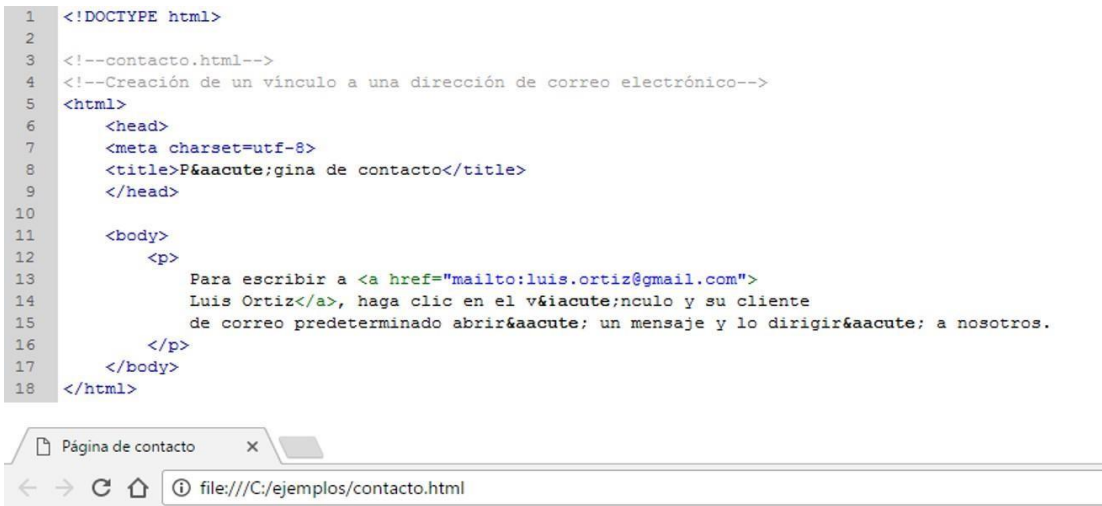
- Una página web o ubicación dentro de una página web
- Un archivo
- Una dirección de correo electrónico

El elemento ancla en la línea 16 vincula el texto Facebook a <http://www.facebook.com>. Cuando un URL no indica un documento específico, el servidor devuelve una página web predeterminada. A esta página se le conoce por lo general como index.html. Si el servidor web no localiza un documento solicitado, devuelve el error 404 y éste muestra una página web que contiene un mensaje de error.

Hipervínculo a una dirección de correo electrónico

Las anclas pueden vincular a direcciones de correo electrónico mediante el uso de un URL mailto: la mayoría de los navegadores inician el programa de correo electrónico predeterminado por el usuario, ejemplo, Mozilla Thunderbird, Microsoft Outlook o Apple Mail). Las líneas 13 y 14 contienen un vínculo de correo electrónico. La forma de un ancla de correo

electrónico es `...`. En la figura 4, creamos un vínculo a la dirección de correo `luis.ortiz@gmail.com`. La línea 13 incluye la dirección de correo electrónico como aparecerá en el mensaje visualizado en el navegador.



Para escribir a [Luis Ortiz](mailto:luis.ortiz@gmail.com), haga clic en el vínculo y su cliente de correo predeterminado abrirá un mensaje y lo dirigirá a nosotros.

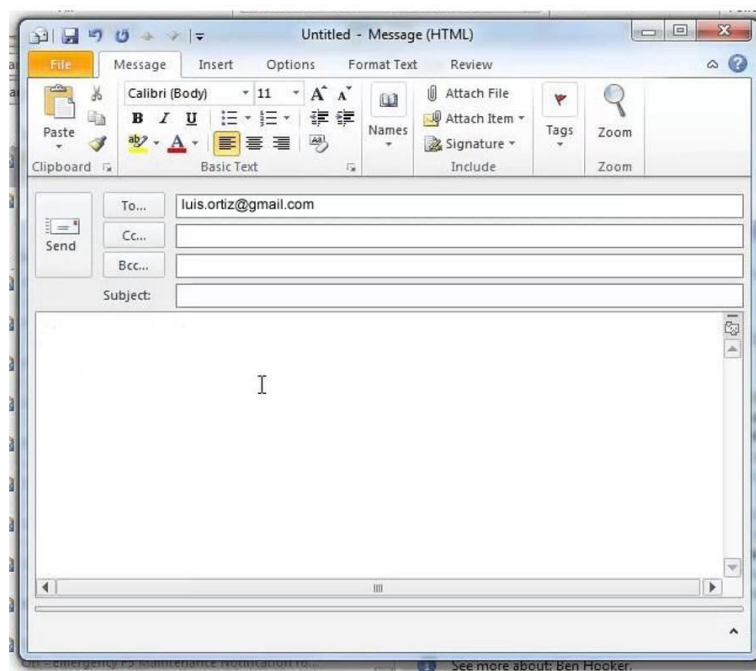


Figura 4. Creación de un vínculo a una dirección de correo electrónico.

Imágenes

Las páginas web no sólo contienen texto; también pueden contener imágenes, animaciones, gráficos, sonidos e incluso videos. PNG y JPG. Es posible adquirir imágenes de sitios web, muchos ofrecen imágenes libres de regalías. Es necesario leer los términos de servicio de cada sitio para determinar si necesitará permiso para utilizar sus imágenes, en especial en aplicaciones comerciales con fines de lucro.

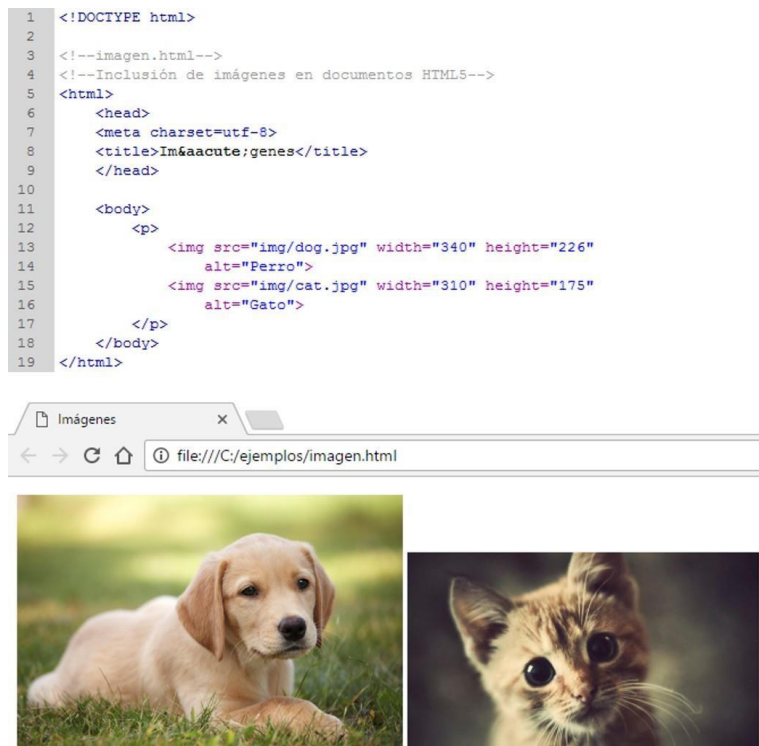


Figura 5. Inclusión de imágenes en documentos HTML5.

Etiquetas img y atributos src, width y height

En las líneas 13 y 14 se utiliza un elemento `img` para incluir una imagen en el documento. La ubicación del archivo se especifica mediante el atributo `src` (origen, source en inglés) esta imagen se encuentra en una carpeta llamada `img`, la cual está dentro de la carpeta que contiene el documento HTML5. Los atributos opcionales `width` y `height` especifican las dimensiones de la imagen. Es recomendable incluir los atributos `width` y `height` para que el navegador cargue el documento HTML5 y sepa cuánto espacio de la pantalla debe proporcionar y pueda visualizar la página en forma correcta, incluso, antes de descargar la imagen.

Atributo alt

Si el navegador no puede desplegar una imagen, debe mostrar el valor del atributo `alt`. El atributo `alt` también es importante por cuestión de accesibilidad: el software sintetizador de voz puede pronunciar su valor, de modo que un usuario con discapacidad visual pueda comprender lo que el navegador visualiza.

Elementos vacíos

Algunos elementos de HTML5 conocidos como elementos vacíos, sólo contienen atributos y no marcan texto, es decir, no se coloca texto entre una etiqueta de inicio y una de fin. Aunque esto no es obligatorio en HTML5, para terminar los elementos vacíos como el elemento `img`, podemos usar el carácter de barra diagonal (`/`) dentro del paréntesis angular derecho de cierre (`>`) de la etiqueta de inicio.

Imágenes como hipervínculos

Al usar imágenes como hipervínculo podemos crear páginas web gráficas que vinculen a otros recursos. Al hacer clic en una imagen, el usuario será transportado a la página web correspondiente.

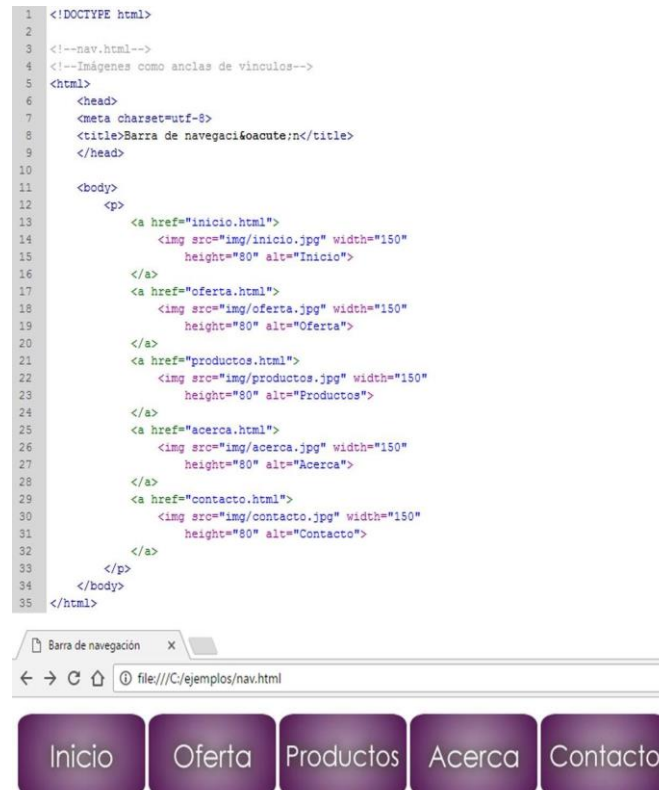


Figura 6. Imágenes como anclas de vínculos.

En las líneas 13 a la 16 se crea un hipervínculo de imagen, para lo cual anidan un elemento `img` en un elemento ancla. Al hacer clic en el hipervínculo de una imagen se transporta a la página web especificada por el atributo `href` del elemento ancla `<a>` ``. Al colocar el mouse sobre un vínculo de cualquier tipo, aparece la URL al que apunta ese vínculo en la barra de estado en la parte inferior de la ventana del navegador.

Listas

En HTML5 se puede utilizar listas para organizar el contenido de la página web. La Figura 7 muestra texto en una lista desordenada, es decir, es una simple lista con viñetas que no orden sus elementos con números o letras. El elemento de lista desordenada `ul` que aparece en las líneas 16 a la 22, crea una lista en donde cada elemento comienza con un símbolo de viñeta. Por lo general, esta viñeta es un círculo relleno. Cada entrada en una lista desordenada es un elemento `li` en una nueva línea con un símbolo de viñeta y con sangría a partir del principio de la línea.

```
1 <!DOCTYPE html>
2
3 <!--vinculos2.html-->
4 <!--Lista desordenada que contiene hipervínculos-->
5 <html>
6   <head>
7     <meta charset=utf-8>
8     <title>V&iacute;nculos</title>
9   </head>
10
11   <body>
12     <h1>Estos son mis sitios favoritos:</h1>
13     <p><strong>Haga clic en un nombre oara visitar ese sitio</strong></p>
14
15     <!--Creación de una lista desordenada-->
16     <ul>
17       <!--La lista contiene cuatro elementos-->
18       <li><a href = "http://facebook.com">Facebook</a></li>
19       <li><a href = "http://twitter.com">Twitter</a></li>
20       <li><a href = "http://foursquare.com">Foursquare</a></li>
21       <li><a href = "http://google.com">Google</a></li>
22     </ul>
23   </body>
24 </html>
```



Estos son mis sitios favoritos:

Haga clic en un nombre oara visitar ese sitio

- [Facebook](http://facebook.com)
- [Twitter](http://twitter.com)
- [Foursquare](http://foursquare.com)
- [Google](http://google.com)

Figura 7. Lista desordenada que contiene hipervínculos.

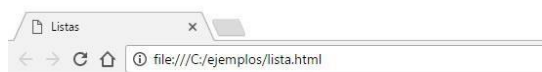
Listas anidadas

Las listas pueden estar anidadas para representar relaciones jerárquicas, como en un esquema multinivel. La Figura 8 muestra el uso de listas anidadas y las listas ordenadas. El elemento de lista ordenada crea una lista en la que cada elemento empieza con un número.

```

1  <!DOCTYPE html>
2
3  <!--lista.html-->
4  <!--Listas anidadas y listas desordenadas-->
5  <html>
6  <head>
7  <meta charset=utf-8>
8  <title>Listas</title>
9  </head>
10
11 <body>
12 <h1>Las mejores característas de Internet</h1>
13
14 <!--Creación de una lista desordenada-->
15 <ul>
16 <li>Puede conocer nuevas personas de países de todo
17 el mundo</li>
18 <li>
19 Tiene acceso a los nuevos medios a medida que se hacen públicos:
20
21 <!--esto empieza una lista desordenada anidada, que usa una -->
22 <!--viñeta distinta. La lista termina al -->
23 <!--cerrar la etiqueta ul.-->
24 <ul>
25 <li>Nuevos juegos</li>
26 <li>Nuevas aplicaciones
27
28 <!-- lista ordenada anidada -->
29 <ol>
30 <li>Por negocios</li>
31 <li>Por placer</li>
32 </ol>
33 </li> <!-- Termina elemento li que inició en la línea 26 y que llama
34 a nuevas aplicaciones-->
35 <li>Noticias las 24 horas</li>
36 <li>Motores de búsqueda</li>
37 <li>Compras</li>
38 <li>Programación
39
40 <!-- otra lista ordenada anidada -->
41 <ol>
42 <li>XML</li>
43 <li>Java</li>
44 <li>HTML5</li>
45 <li>JavaScript</li>
46 <li>Nuevos lenguajes</li>
47 </ol>
48 </li> <!-- Termina elemento li, llamado programación, que inició en la línea 38 -->
49 </ul> <!-- Termina la lista anidada de la línea 24 -->
50 </li>
51 <li>Vínculos</li>
52 <li>Mantenerse en contacto con viejos amigos</li>
53 <li>Es la tecnología del futuro</li>
54 </ul> <!-- termina la lista desordenada de la línea 15 -->
55 </body>
56 </html>

```



Las mejores características de Internet

- Puede conocer nuevas personas de países de todo el mundo
- Tiene acceso a los nuevos medios a medida que se hacen públicos:
 - Nuevos juegos
 - Nuevas aplicaciones
 1. Por negocios
 2. Por placer
 - Noticias las 24 horas
 - Motores de búsqueda
 - Compras
 - Programación
 1. XML
 2. Java
 3. HTML5
 4. JavaScript
 5. Nuevos lenguajes
- Vínculos
- Mantenerse en contacto con viejos amigos
- Es la tecnología del futuro

Figura 8. Listas anidadas y listas desordenadas.

En muchos navegadores se colocan viñetas antes de los elementos en la lista desordenada más externa, como se puede ver en las líneas 15 a 55. Ahora bien, en muchos navegadores se colocan viñetas circulares sin relleno antes de los elementos de la lista anidada que está dentro de la lista desordenada de la línea 15. Un navegador web aplica sangría a cada lista

anidada para indicar una relación jerárquica. La primera lista ordenada, de las líneas 29 a 33, incluye dos elementos. Los elementos en una lista ordenada se enumeran como 1., 2., 3., etc. Las listas ordenadas anidadas se enumeran de la misma forma. Por lo general, se colocan viñetas cuadradas antes de los elementos subsiguientes de las listas desordenadas anidadas. Los estilos de las viñetas pueden variar según el navegador.

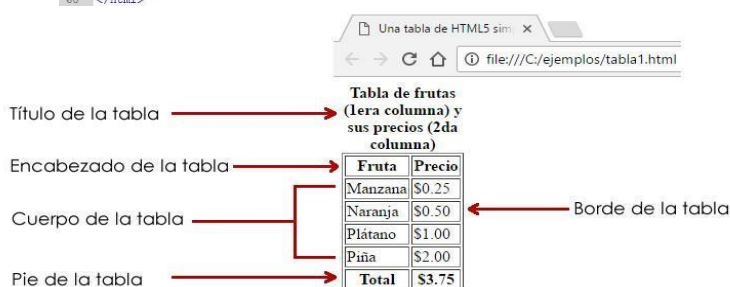
Tablas

Las tablas se usan con frecuencia para organizar datos en filas y columnas. La Figura 9 muestra un ejemplo de una tabla con seis filas y dos columnas que muestra la información de precios de varias frutas. Las tablas se definen mediante el elemento `table` como se puede ver en las líneas 13 a la 58. La línea 13 especifica la etiqueta de inicio del elemento `table`. El atributo `border` con el valor de 1 especifica que el navegador debe colocar bordes alrededor de la tabla y alrededor de sus celdas. Este atributo de herencia se debe evitar. Se recomienda darles formato a las tablas mediante CSS.

```

1 <!DOCTYPE html>
2
3 <!--tabla1.html-->
4 <!--Creación de una tabla básica-->
5 <html>
6   <head>
7     <meta charset=utf-8>
8     <title>Una tabla de HTML5 simple</title>
9   </head>
10
11   <body>
12     <!--La etiqueta <table> abre una tabla-->
13     <table border="1">
14
15       <!-- la etiqueta <caption> sintetiza el contenido de -->
16       <!-- la tabla (esto ayuda a las personas con discapacidad visual)-->
17       <caption><strong>Tabla de frutas (1era columna) y
18         sus precios (2da columna)</strong></caption>
19
20       <!-- la sección <thead> aparece primero en la tabla -->
21       <!-- da formato al área del encabezado de la tabla-->
22       <thead>
23         <tr> <!-- <tr> inserta una fila en la tabla -->
24           <th>Fruta</th> <!-- inserta una celda de encabezado -->
25           <th>Precio</th>
26         </tr>
27       </thead>
28
29       <!-- la sección <tbody> aparece al último en la tabla -->
30       <!-- da formato al pie de página -->
31       <tbody>
32         <tr>
33           <td>Total</td>
34           <td>$3.75</td>
35         </tr>
36       </tbody>
37
38       <!-- todo el contenido de la tabla está encerrado -->
39       <!-- dentro del elemento <tbody> -->
40       <tbody>
41         <tr>
42           <td>Manzana</td>
43           <td>$0.25</td>
44         </tr>
45         <tr>
46           <td>Naranja</td>
47           <td>$0.50</td>
48         </tr>
49         <tr>
50           <td>Plátano</td>
51           <td>$1.00</td>
52         </tr>
53         <tr>
54           <td>Piña</td>
55           <td>$2.00</td>
56         </tr>
57       </tbody>
58     </table>
59   </body>
60 </html>

```



Fruta	Precio
Manzana	\$0.25
Naranja	\$0.50
Plátano	\$1.00
Piña	\$2.00
Total	\$3.75

Figura 9. Una tabla de HTML5 simple.

1.6 Estructura y disposición

La estructura del cuerpo, es decir, el código que se encuentra entre las etiquetas <body>, generará la parte visible del documento. Este es el código que producirá nuestra página web. HTML siempre ofreció diferentes formas de construir y organizar la información dentro del cuerpo de un documento. Uno de los primeros elementos provistos para este propósito fue <table>. Las tablas permitían a los diseñadores acomodar datos, texto, imágenes y herramientas dentro de filas y columnas de celdas, incluso sin que hayan sido concebidas para este propósito. En los primeros días de la web, las tablas fueron una revolución, un gran paso hacia adelante con respecto a la visualización de los documentos y la experiencia ofrecida a los usuarios. Más adelante, gradualmente, otros elementos reemplazaron su función, permitiendo lograr lo mismo con menos código, facilitando de este modo la creación, permitiendo portabilidad y ayudando al mantenimiento de los sitios web.

Organización

La Figura 10 muestra un diseño común encontrado en la mayoría de los sitios webs estos días. A pesar del hecho de que cada diseñador crea sus propios diseños, en general podremos identificar las siguientes secciones en cada sitio web estudiado:

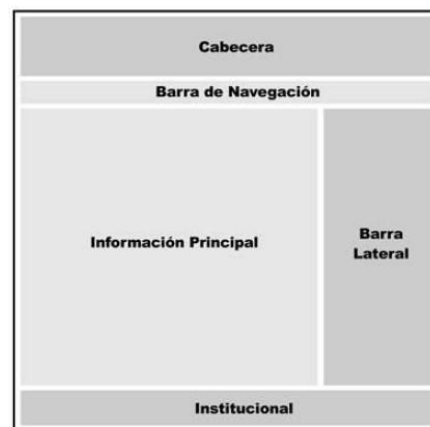


Figura 10. Representación visual de un clásico diseño web.

En la parte superior, descrito como Cabecera, se ubica el logo, título, subtítulos y una corta descripción del sitio web. Inmediatamente debajo, podemos ver la Barra de Navegación, en la cual casi todos los desarrolladores ofrecen un menú para facilitar la navegación. Los usuarios son guiados desde esta barra hacia las diferentes páginas o documentos, normalmente pertenecientes al mismo sitio web.

El contenido más relevante de una página web se encuentra, en casi todo diseño, ubicado en su centro. Esta sección presenta información y enlaces valiosos. La mayoría de las veces es dividida en varias filas y columnas. En el ejemplo de la **¡Error! No se encuentra el origen de la referencia.** Se utilizaron sólo dos columnas: Información Principal y Barra Lateral, pero esta sección es extremadamente flexible y normalmente la adaptan acorde a las necesidades insertando más columnas, dividiendo cada columna entre bloques más pequeños o generando diferentes distribuciones y combinaciones.

La Figura 11 es una representación de un blog normal. En este ejemplo se puede claramente identificar cada parte del diseño considerado anteriormente.

1. Cabecera
2. Barra de Navegación
3. Sección de Información Principal
4. Barra Lateral
5. El pie o la barra Institucional



Figura 11. Representación visual de un clásico diseño para blogs.

Esta simple representación de un blog nos puede ayudar a entender que cada sección definida en un sitio web tiene un propósito. A veces este propósito no es claro, pero en esencia se encuentra siempre allí, ayudándonos a reconocer cualquiera de las secciones descritas anteriormente en todo diseño. HTML5 considera esta estructura básica y provee nuevos elementos para diferenciar y declarar cada una de sus partes. A partir de ahora podemos decir al navegador para qué es cada sección.

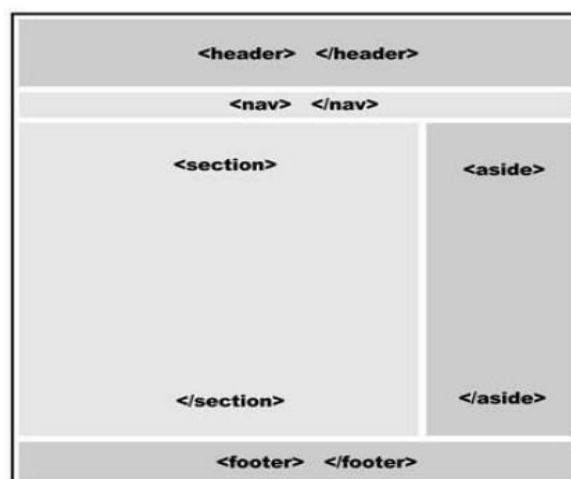


Figura 12. Representación visual de un diseño utilizando elementos HTML5.

Elementos estructurales

Como se mencionó, durante muchos años se han utilizado `<div>` y `class` para definir (casi por regla general) un header, un footer, navs y otras entidades muy comunes en el desarrollo de cualquier sitio web. Finalmente llegó el momento de incorporar formalmente a todas esas entidades como parte del lenguaje y actualmente son parte de la nueva especificación de HTML5.

Article. La etiqueta con cierre `<article>` es como una página pequeña dentro de una página. Debe usarse para definir contenido autónomo e independiente, proveniente de una fuente externa de información y predecible de ser reutilizada de modo aislado.

Aside. Cuando se desea colocar información en un fragmento de la página que está relacionado con el contenido principal, pero debe estar separado de él, se utiliza la etiqueta con cierre `<aside>`. Comúnmente se le ubica como una barra lateral. Se debe tener en cuenta que, si desea colocar una barra lateral porque el diseño visual lo indica, no significa necesariamente que deba utilizar `<aside>`.

Footer. Se utiliza para colocar información de derechos de autor, información de contacto o créditos. `<footer>` sustituye al habitual `<div id="footer"></div>`, tan utilizado en versiones anteriores de HTML.

Header. El elemento con cierre `<header>` se utiliza como tope o cabecera visible de una página. No confunda `<header>` con la etiqueta `<head>` que contiene el título y los metadatos de nuestras páginas.

Nav. La etiqueta `<nav>` junto con su cierre es utilizada para contener información sobre la navegación del sitio web. Se utiliza en enlaces con la etiqueta `<a>`, pero éstos deben ser enlaces sólo para la navegación principal del sitio y no para enlaces externos. El elemento `<nav>` ha sido creado buscando sustituir a `<div id="nav"></div>`.

Section. El nuevo elemento con cierre `<section>` intenta albergar contenido específico sobre algún tema, es muy similar a usar la etiqueta `<div>`, pero `<div>` no tiene un significado semántico y no informa sobre el tipo de contenido alojado en la etiqueta.

Hasta el momento el soporte para `<section>` por parte de los navegadores es reducido. Los elementos `<div>` y `<article>` pueden ser buenos sustitutos.

Elementos media

Ahora HTML5 soporta la inserción de audio y video sin necesidad de ningún componente externo e incluye también soporte para gráficas vectoriales a través del elemento `<canvas>` y SVG.

Audio. Cualquier texto entre las etiquetas `<audio>` y `</audio>` será desplegado en cualquier navegador que no soporte el elemento de audio, opcionalmente también puede colocar en lugar de texto un enlace para descargar el archivo de audio o embeber un reproductor Flash. El elemento `<audio>` cuenta con los atributos siguientes:

- **Atributo autoplay.** Indica si el archivo debe ser automáticamente reproducido o no.
- **Atributo controls.** Especifica si los controles deberían ser o no desplegados.
- **Atributo loop.** Establece si la reproducción debería reiniciar cuando llega al final.
- **Atributo preload.** Especifica si el audio debe cargar al abrir la página.
- **Atributo src.** Indica la dirección URL del archivo de audio a reproducir.

Aunque <audio> ya es soportado por los principales navegadores, los fabricantes no están completamente de acuerdo con el formato de audio y por lo tanto, soportan diferentes formatos entre el estándar libre Ogg y Mp3, el navegador reproducirá el primer formato compatible.

Canvas. Cuando utiliza <canvas> (lienzo) el navegador reserva una parte de la ventana para el uso de gráficos, los cuales serán dibujados justo ahí.

Cualquier cosa que escriba entre <canvas> y </canvas> será interpretado por navegadores que no soportan aún la nueva etiqueta. La Figura 13 muestra el resultado que despliega el navegador.

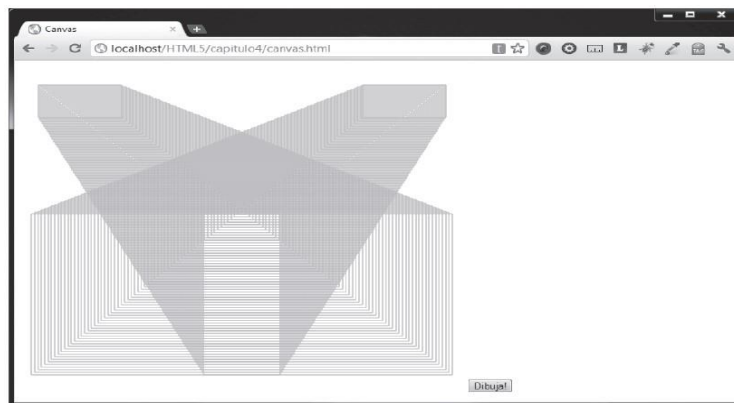


Figura 13. Página con función JavaScript que dibuja rectángulos dentro de <canvas>.

Embed. <embed> es la etiqueta general para embeber cualquier tipo de media en una página. En teoría esta etiqueta permite embeber cualquier tipo de media que no es soportada nativamente por ninguna etiqueta. Utilice esta etiqueta como método de respaldo a etiquetas más confiables como <audio> y <video>. Por otro lado, si desea utilizar contenido Flash en el contexto de HTML5, <embed> es la mejor manera de incluir este tipo de contenido.

Svg. La etiqueta con cierre <svg> es otra opción disponible en HTML5. La característica más popular es que permite escalar imágenes sin perder calidad.

Video. La etiqueta con cierre <video> es otro elemento que ha causado gran expectativa y ha estado involucrado en una interesante polémica (animada por el cofundador de Apple, Steve Jobs). Cuando vemos un video en un reproductor, normalmente lo que vemos son las imágenes (cuadros) del video y su respectivo audio, pero no usamos dos archivos separados para cada uno, si no que tenemos un solo archivo, quizá en formato Mp4, AVI o MOV, entre otros. Pero éstos son sólo formatos contenedores, simplemente definen cómo almacenar la

información de video y audio en un solo archivo. Cuando usted ve un video, su reproductor hace básicamente 3 cosas:

1.7 Formularios

Todo el mundo se ha visto en la necesidad de llenar alguna solicitud de empleo o un documento para completar algún trámite escolar o legal en papel, a estos documentos en papel se les llama formas o formularios. No se puede manipular la información de los formularios sólo con HTML, se necesitan otras tecnologías como ASP, JSP, PHP, etc. y JavaScript. Sin embargo, es importante saber cómo trabaja cada elemento de los formularios.



Figura 14. Formulario con controles clásicos.

Estructura de un formulario

La estructura general del formulario utiliza las siguientes etiquetas:

Etiqueta con cierre <form>. Esta etiqueta utiliza varios atributos, pero los más utilizados son action, que indica a dónde enviar los datos del formulario para que sean procesados y method que especifica la manera en que se enviarán los datos en el protocolo HTTP.

Etiqueta con cierre <fieldset>. Agrupa una serie de elementos de un formulario. No es una etiqueta requerida en la estructura de formulario, pero es útil para organizar mejor los elementos visualmente.

- Etiqueta con cierre <legend>. Coloca un texto que sirve de leyenda para el fieldset.
- Etiqueta con cierre <label>. Es una etiqueta se utiliza para colocar texto referente a un elemento de formulario. Si se utiliza el atributo for entonces <label> queda amarrado con dicho elemento, de manera que si el usuario hace clic sobre <label> el foco lo obtendrá el elemento de formulario referido en el atributo for.

Campo de texto

Muchos de los controles de formulario están basados en el uso de la etiqueta `<input>`. El usuario al hacer clic en el campo colocará el cursor y podrá escribir dentro de él, tenga en cuenta que el campo de texto simple es para información corta y concreta.

Área de texto

Hay momento en donde se requiere de varias líneas de texto para escribir un dato largo, dado que el campo de texto simple es para información breve, el control área de texto se vuelve muy conveniente. Éste no es un control de tipo `<input>` por lo que la sintaxis es algo diferente, ya que se usa la etiqueta con cierre `<textarea>`.

Campo de contraseña

En muchos formularios es común requerir datos personales que no es conveniente que estén a la vista de cualquiera, como, por ejemplo, contraseña entre otros. Para ello se debe crear un campo que no muestre los datos que se escriben en él. Para lograr esto se utiliza la etiqueta `<input>` con el atributo `type` con valor `password`.

Lista de selección simple (combobox)

Las listas de selección simple como combobox son muy comunes y de gran ayuda para los formularios en los que se requiera saber un dato preciso y excluyente de otros, de manera que se limite a opciones planteadas previamente por nosotros. Este tipo de elemento implícitamente nos ofrece la ventaja de poder evitar errores de escritura por parte del usuario y predecir exactamente los posibles valores que podemos obtener.

Lista de selección múltiple (listbox)

Es posible que en lugar de tener una lista de selección simple donde sólo se puede ver una opción a la vez, si se desea ver más de una opción a la vez, se seguirán los mismos pasos que se utilizan para la lista de selección múltiple, únicamente se agregará el atributo `size` a la etiqueta `<select>` y se le asignará un valor numérico correspondiente a la cantidad de opciones que desea ver al mismo tiempo.

Lista de selección múltiple (listbox multiple selection)

Si desea seleccionar más de una opción al mismo tiempo, sólo agregue otros pasos a las listas de selección anteriores (combobox y listbox. Únicamente se requiere colocar el atributo `multiple` a la etiqueta `<select>` sin ningún otro valor.

Lista de botones radio (radio buttons)

Otro tipo de lista para la etiqueta `<input>` son las listas de botones radio. Este tipo de control siempre va en grupos de 2 o más miembros en la lista y por lo menos un elemento siempre debe estar seleccionado.

Lista de cajas (checkboxes)

Se encontrará casos en donde cierta información requiere una respuesta positiva o falsa. El control apropiado para enfrentar esta situación son las cajas de selección, en donde el usuario marcará o demarcará con un clic en la caja correspondiente a la opción deseada. Las cajas

de selección a menudo forman grupos para aparecer en listas, como es nuestro caso, pero actúan independientemente una de otra.

Botón para enviar información

El botón para enviar información o submit, normalmente se utiliza para enviar la información coleccionada por todos los campos del formulario al lado del servidor. Alguna tecnología de servidor como ASP o PHP entre otras procesa esta información.

Botón para borrar información del formulario

Cuando el usuario hace clic en él, todos los campos del formulario borran su valor actual y regresan a su estado original. Trabaja con la etiqueta `<input>` con el valor de type igual a reset, el texto para el atributo value funciona como leyenda dentro del botón.

Botón simple

El botón simple tiene el mismo aspecto que los dos anteriores y tiene la función de trabajar del lado cliente, usualmente este botón trabaja con JavaScript para disparar alguna acción, el texto para el atributo value funciona como leyenda dentro del botón.

Imagen (botón)

Este tipo de botón trabaja de la misma manera que el botón simple. Sin embargo, la diferencia radica en que, en lugar de trabajar con la apariencia por default del botón simple, utiliza en su lugar una imagen indicada. El atributo type en `<submit>` contiene el valor image, se utiliza el atributo src que indicará la ruta donde se encuentra la imagen que desee usar y el atributo alt para un texto alternativo.

Notas adicionales sobre botones

La versátil etiqueta `<input>` permite crear botones, esta forma es la manera tradicional de hacer botones en HTML y es válida en HTML5.

Campo oculto

El campo oculto es una etiqueta `<input>` con el valor de type igual a hidden. Gracias a este atributo se envía al programa de gestión de datos, además de los datos enviados por el usuario, datos predefinidos por nosotros mismos, invisibles para el usuario. Estos datos pueden ser útiles para ayudar al programa servidor en la gestión de los datos enviados por el formulario.