

# Guía Diseño Web

### Índice

1.0 ¿Qué es HTML? .....	1
2.0 Etiquetas básicas de HTML.....	1
3.0 Crear documento HTML .....	4
4.0 Metadatos.....	4
5.0 Hipervínculos.....	6
6.0 Incrustar videos, audios e imágenes .....	6
7.0 Formularios .....	10
8.0 Tablas.....	16
9.0 Secciones.....	20
10.0 Que es CSS .....	22
11.0 Uso de CSS desde y fuera de un documento HTML.....	25
12.0 Bootstrap .....	25

### 1.0 ¿Qué es HTML?

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).

"Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «World Wide Web» (Red Informática Mundial).

### 2.0 Etiquetas básicas de HTML

Las etiquetas más básicas de Html son las siguientes:

Todos los documentos HTML deben comenzar con una **<!DOCTYPE>** declaración.

La declaración no es una etiqueta HTML. Es una "información" para que el navegador sepa qué tipo de documento esperar.

**<html></html>** — el elemento **<html>**. Este elemento encierra todo el contenido de la página entera y, a veces, se le conoce como el elemento raíz (root element).

**<head></head>** — el elemento **<head>**. Este elemento actúa como un contenedor de todo aquello que quieres incluir en la página HTML que no es contenido visible por los visitantes de la página. Incluye cosas como palabras clave (keywords), una descripción de la página que quieres que aparezca en resultados de búsquedas, código CSS para dar estilo al contenido, declaraciones del juego de caracteres, etc.

**<meta charset="utf-8">** — **<meta>**. Este elemento establece el juego de caracteres que tu documento usará en utf-8, que incluye casi todos los caracteres de todos los idiomas humanos. Básicamente, puede manejar cualquier contenido de texto que puedas incluir. No hay razón para no establecerlo, y puede evitar problemas en el futuro.

**<title></title>** — el elemento **<title>** establece el título de tu página, que es el título que aparece en la pestaña o en la barra de título del navegador cuando la

página es cargada, y se usa para describir la página cuando es añadida a los marcadores o como favorita.

**<body></body>** — el elemento **<body>**. Encierra todo el contenido que deseas mostrar a los usuarios web que visiten tu página, ya sea texto, imágenes, videos, juegos, pistas de audio reproducibles, y demás.

Código básico en html que se visualiza de la siguiente forma:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mi página de prueba</title>
  </head>
  <body>

  </body>
</html>
```

Etiqueta básica para la llamada de una imagen es:

****

Como ya se dijo antes, incrusta una imagen en la página, en la posición en que aparece. Lo logra a través del atributo **src (source)**, el cual contiene el path (ruta o ubicación) de tu archivo de imagen.

También se incluye un atributo **alt** (alternative) el cual contiene un texto que debería describir la imagen, y que podría ser accedido por usuarios que no pueden ver la imagen, quizás porque:

Se produjo algún error en el código que impide que la imagen sea cargada. Como ejemplo, modifica deliberadamente la ubicación dentro del atributo **src** para que este sea incorrecto.

### Encabezados

Los elementos de encabezado permiten especificar que ciertas partes del contenido son encabezados, o subencabezados del contenido. De la misma forma que un libro tiene un título principal, y que a su vez puede tener títulos por cada capítulo individual, y subtítulos dentro de ellos, un documento HTML puede tenerlos también.

HTML posee seis niveles de encabezados, **<h1>–<h6>**, aunque probablemente solo llegues a usar 3-4 como mucho:

```
<h1>Mi título principal</h1>
<h2>Mi título de nivel superior</h2>
<h3>Mi subtítulo</h3>
<h4>Mi sub-subtítulo</h4>
```

### Párrafos

Como se explicó más arriba, los elementos **<p>** se utilizan para encerrar párrafos de texto; los usarás frecuentemente para el marcado de contenido de texto regular:

```
<p>Este es un simple parrafo</p>
```

### Listas

Mucho del contenido web está dado por listas, así que HTML tiene elementos especiales para ellas. El marcado de listas se realiza siempre en al menos dos elementos. Los dos tipos de listas más comunes son las listas ordenadas y las desordenadas:

Las listas desordenadas son aquellas en las que el orden de los items no es relevante, como en una lista de compras. Estas son encerradas en un elemento **<ul>** (unordered list).

Las listas ordenadas son aquellas en las que el orden sí es relevante, como en una receta. Estas son encerradas en un elemento **<ol>** (ordered list).

Cada elemento de la lista se coloca dentro de un elemento **<li>** (list item).

### Lista desordenada

```
<ul>
  <li>tecnólogos</li>
  <li>pensadores</li>
  <li>constructores</li>
</ul>
```

```
<ol>
  <li>tecnólogos</li>
  <li>pensadores</li>
  <li>constructores</li>
</ol>
```

### Vínculos

Los vínculos o enlaces son muy importantes —son los que hacen de la web, la web. Para implementar un vínculo, necesitas usar un vínculo simple — **<a>** — la **a** es la abreviatura de la palabra inglesa «anchor» («ancla»). Para convertir algún texto en un vínculo, sigue estos pasos:

```
<a href="https://www.google.com">dar clic aquí</a>
```

### 3.0 Crear documento HTML

Para crear un documento HTML, se debe escribir código HTML en algún editor de texto, por ejemplo el bloc de notas, y al momento de guardar el documento se debe guardar con extensión **.html**

Ejemplo sería el famoso archivo index, que debe guardarse de la siguiente forma:

**Index.html**

### 4.0 Metadatos

Metadatos: el elemento **<meta>**

Los metadatos son datos que describen datos, y HTML tiene una forma «oficial» de introducir metadatos en un documento — el elemento **<meta>**. Por supuesto, el resto de los elementos de los que hablaremos en este artículo también se podrían considerar metadatos. Hay muchos diferentes tipos de elementos **<meta>** que se pueden incluir en el **<head>** de tu página, pero no vamos a intentar explicarlos todos en esta etapa porque resultaría demasiado confuso. En lugar de ello vamos a explicar algunas cuestiones con las que seguramente te vas a encontrar, solo para que te hagas una idea.

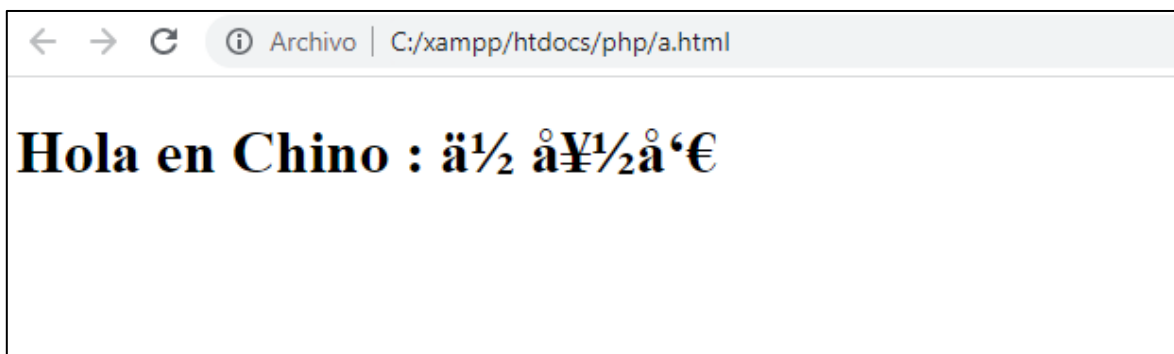
Etiqueta meta

```
<meta charset="utf-8">
```

Este elemento simplemente especifica la codificación de caracteres del documento — es decir, el conjunto de caracteres que el documento puede usar. utf-8 es un conjunto de caracteres universal que incluye casi todos los caracteres de casi cualquier idioma humano. Esto significa que tu página web podrá gestionar la visualización de cualquier idioma; por lo tanto, resulta una buena idea configurarlo en cada página web que crees. Por ejemplo, tu página podría gestionar inglés y japonés sin problemas:



Si configuras tu codificación de caracteres en ISO-8859-1, por ejemplo (el juego de caracteres para el alfabeto latino), la representación de tu página puede aparecer desordenada:



**Nota:** Algunos navegadores (Chrome, por ejemplo) automáticamente arreglan la codificación de caracteres incorrecta, de modo que según el navegador que uses, puede que no te topes con este problema. Aun así, deberías incluir una codificación de caracteres utf-8 en tu página web para evitar que se presenten potenciales problemas con otros navegadores.

### Añadir un autor

Muchos elementos **<meta>** incluyen atributos **name**.

**name** especifica el tipo de metadato del que se trata; es decir, qué tipo de información contiene.

Dos de esos metadatos que resultan útiles de incluir en tu página definen al autor de la página y proporcionan una descripción concisa de la página. Veamos un ejemplo:

```
<meta name="author" content="Alexis Lozada">
```

### 5.0 Hipervínculos

Los hipervínculos son una de las innovaciones más interesantes que ofrece la Web. Han formado parte de esta desde el principio. Los hipervínculos nos permiten vincular documentos a otros documentos o recursos, vincular a partes específicas de documentos o hacer que las aplicaciones estén disponibles en una dirección web. Prácticamente cualquier contenido web se puede convertir en un enlace que, al pulsarlo (activarlo), dirija el navegador a la dirección web a la que apunta el enlace (URL).

```
<a href="https://www.google.com">dar clic aquí</a>
```

### 6.0 Incrustar videos, audios e imágenes

#### Imágenes

Al principio, la web solo era texto y resultaba más bien aburrido. Afortunadamente, no pasó mucho tiempo antes de que se añadiera la capacidad de insertar imágenes (y otros tipos de contenido más interesantes) en las páginas web. Hay otros tipos de elementos multimedia que tener en cuenta, pero es lógico comenzar con el humilde elemento <img> utilizado para insertar una imagen simple en una página web.

Para poner una imagen simple en una página web, utilizamos el elemento <img>. Se trata de un elemento vacío (lo que significa que no contiene texto o etiqueta de cierre) que requiere de por lo menos de un atributo para ser utilizado: src (a veces denominado por su nombre completo, source). El atributo src contiene una ruta que apunta a la imagen que quieres poner en la página, que puede ser una URL relativa o absoluta.

```

```

También se puede incrustar la imagen usando la URL absoluta, por ejemplo:

```

```

Como se vería nuestra imagen en el navegador:

### Images in HTML



### Videos

El elemento video se utiliza para incrustar vídeos en un documento HTML.

### Atributos

#### autoplay

Un atributo booleano; si se especifica, el video comenzará a reproducirse automáticamente tan pronto como sea posible, sin detenerse para terminar de cargar los datos.

#### autobuffer

Un atributo booleano; si se especifica, el video comenzará automáticamente a almacenarse en el búfer, incluso si no está listo para reproducirse de forma

automática. Esto se debe utilizar para los casos en los que se considera probable que el video se reproduzca (por ejemplo, si el usuario accedió a esa página específica para reproducir el video, no si hay un video insertado junto con otros contenidos). El video se almacena en el búfer hasta que el caché de medios esté lleno.

Nota de implementación: aunque forma parte de los primeros borradores de la especificación HTML 5, el atributo autobuffer se ha eliminado en versiones posteriores. Se ha quitado de Gecko 2.0 y otros navegadores, y en algunos nunca llegó a implementarse. La especificación define un nuevo atributo enumerado, preload, para sustituir el atributo autobuffer, con una sintaxis diferente. error 548523.

### **buffered**

Un atributo que se puede leer para determinar qué intervalos de tiempo del multimedia se han almacenado en búfer. Este atributo contiene un objeto TimeRanges (en-US).

### **controls**

Si está presente este atributo, Gecko ofrecerá controles para permitir que el usuario controle la reproducción de video, incluyendo volumen, búsqueda y pausar/reanudar reproducción.

### **height**

La altura del área de visualización del vídeo en píxeles CSS.

### **loop**

Un atributo booleano; si se especifica, al alcanzar el final del video, buscaremos automáticamente hasta el principio.

**none:** sugiere bien que el autor cree que el usuario no tendrá que consultar ese video, bien que el servidor desea minimizar su tráfico; es decir, esta sugerencia indica que no se debe almacenar en caché este video.

**metadatos:** sugiere que aunque el autor piensa que el usuario no tendrá que consultar este video, es razonable capturar los metadatos (p. ej. longitud).

**auto:** sugiere que el usuario necesita tener prioridad; es decir, esta sugerencia indica que, si es necesario, se puede descargar el video completo, incluso aunque el usuario no vaya a usarlo.

La cadena vacía: que es un sinónimo del valor auto.

Si no está configurado, su valor predeterminado está definido por el navegador (es decir, cada navegador puede elegir su propio valor predeterminado), aunque la especificación aconseja que se establezca a metadata.

### **notas de uso:**

El atributo `autoplay` tiene prioridad sobre éste si se desea reproducir automáticamente un video, el navegador obviamente tendrá que descargarlo. La especificación permite establecer los atributos `autoplay` y `preload`.

La especificación no fuerza al navegador a seguir el valor de este atributo; es tan sólo una sugerencia.

### **poster**

Una URL que indica un marco de póster para mostrar el resultado hasta que el usuario reproduzca o busque. Si este atributo no se especifica, no se muestra nada hasta que el primer cuadro está disponible, entonces se muestra el primer marco como el marco de póster.

### **src**

La URL del vídeo que se va a insertar. Es opcional; podrás optar, en su lugar, por el elemento `<source>` dentro del bloque de vídeo para especificar el video que se va a incrustar.

### **width**

La anchura del área de visualización del vídeo en píxeles CSS.

Las compensaciones de tiempo se especifican actualmente como valores float que representan el número de segundos que se va a compensar.

```
<video src="videofile.ogg" autoplay poster="posterimage.jpg">
  Tu navegador no admite el elemento <code>video</code>.
```

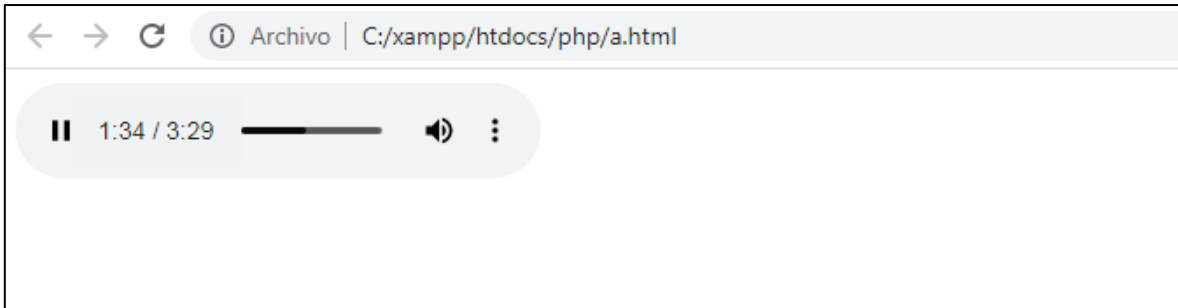
### **Audio**

El elemento audio se usa para insertar contenido de audio en un documento HTML o XHTML. El elemento audio se agregó como parte de HTML 5.

Ejemplo de código:

```
<audio controls>
  <source src="pastor.mp3" type="audio/mp3">
  Tu navegador no soporta audio HTML5.
</audio>
```

Salida en el navegador:



### 7.0 Formularios

Son elementos de las páginas web que nos sirven para recuperar información del usuario y enviarla al servidor para que pueda ser procesada.

La etiqueta que delimita un formulario es la etiqueta **<form>...</form>**. Los atributos más importantes de la etiqueta **<form>** son:

- **action**: contiene el nombre del agente que procesará los datos remitidos al servidor (por ejemplo, un script de PHP).
- **method**: define la manera de enviar los datos al servidor. Los valores posibles son:
  - **get**: los valores enviados se añaden a la dirección indicada en el atributo action.
  - **post**: los valores se envían de forma separada.

Si el atributo method no está establecido, el formulario se comporta como si el valor fuera get.

- Campos de entrada de datos:
  - Campos de textos.
  - Contraseñas.
  - Checkbox.
  - Radios.
  - Botones de envío.
  - Botones de borrado.

- Ficheros.
- Campos Ocultos.
- Imágenes.
- Botones.
- Áreas de texto.
- Combos de selección.

### **Campos de entrada de datos**

Estos son los elementos básicos de un formulario ya que son los que nos permiten recuperar información del usuario de diferentes formas. El elemento que representa los campos de entrada de datos es **<input>...</input>**.

**<input type="tipo" id="identificador" size="tamaño" name="nombre" value="texto por defecto"/>**

Si vemos los atributos que tiene el elemento input nos encontramos los siguientes:

- **type**, indica el tipo de campo de entrada de datos que vamos a utilizar. Dependiendo del tipo que indiquemos obtendremos un resultado u otro. Los valores que puede tener el atributo type son: “text”, “password”, “checkbox”, “radio”, “submit”, “reset”, “file”, “hidden”, “image” y “button”.
- **id**, es el identificador del campo. Es importante ya que será el nombre por el cual podremos identificar, de forma unívoca, al campo.
- **size**, será el tamaño del campo. Es decir, el número de caracteres que podríamos insertar en el campo de texto.
- **name**, es el nombre del campo el cual se enviará desde el formulario al servidor.
- **value**, será el valor por defecto que tendrá el campo de texto y que le aparecerá al usuario al cargar el formulario.

### **Campos de texto**

El campo de texto, como bien indica su nombre, es el que nos permite insertar texto en un formulario. El usuario podrá insertar texto visible sobre el campo.

En este caso el tipo de elemento input que utilizaremos será “text”. Así, para definir un campo de texto lo haremos de la siguiente forma:

**<input type="text" id="identificador" size="tamaño" name="nombre" value="texto por defecto"/>**

Cuando un usuario inserte una contraseña dentro de un formulario necesitaremos, casi seguro, que el valor de la contraseña no aparezca y que por el contrario aparezcan caracteres como asteriscos.

Para insertar un campo que acepte contraseñas dentro de un formulario vamos a utilizar un tipo “password” dentro del elemento input.

```
<input type="password" id="identificador" size="tamaño" name="nombre"/>
```

### Checkbox

Un checkbox nos permite capturar un dato del usuario mediante un elemento de check. El check puede tener dos valores, seleccionado o no seleccionado. El tipo del elemento input que utilizaremos será “checkbox”. Así lo definiremos de la siguiente forma:

```
<input type="checkbox" id="identificador" name="nombre"/>
```

En el caso del checkbox no tienen sentido el atributo tamaño ni el valor por defecto. Ya que, recordemos que solo podemos tener el check seleccionado o no. Pero lo que sí podemos hacer es generar un checkbox que esta preseleccionado. Para ello utilizamos el atributo **checked**.

Los **checkbox** suelen ir en grupos para seleccionar varias opciones. Por ejemplo, podríamos tener el siguiente código con el que podamos seleccionar qué lenguaje de programación queremos aprender.

```
<input type="checkbox" name="lenguaje" value="html">HTML  
<input type="checkbox" name="lenguaje" value="javascript">Javascript  
<input type="checkbox" name="lenguaje" value="css">CSS  
<input type="checkbox" name="lenguaje" value="xml">XML
```

### Radios

Con los elementos de radio podemos ofrecer un conjunto de opciones al usuario de tal manera que solo pueda elegir una de ellas. El tipo de elemento input que utilizamos es “radio”. La sintaxis que seguiremos en los elementos input de tipo radio será la siguiente:

```
<input type="radio" id="identificador" value="valor" name="nombre"/>
```

En el caso de los elementos radio toma un papel principal el atributo name. Ya que para poder agrupar opciones deberemos de tener el mismo valor de atributo name.

Así, si queremos crear un grupo de radios para que nos elija una edad le podremos crear de la siguiente forma:

```
<input type="radio" id="menos18" value="menos18" name="edad"/>Menos de 18  
<input type="radio" id="18a30" value="18a30" name="edad"/>18 a 30  
<input type="radio" id="31a50" value="31a50" name="edad"/>31 a 50  
<input type="radio" id="mas50" value="mas50" name="edad"/>Más de 50
```

### Ficheros

Cuando diseñemos un formulario es posible que necesitemos enviar un fichero de datos al servidor. En este caso el campo de entrada de datos nos tiene que dar la posibilidad de acceder al sistema de fichero del dispositivo para poder seleccionar uno.

En este caso vamos a necesitar un campo de entrada de tipo “file”. La sintaxis de los campos de entrada para tipos file sería:

```
<input type="file" id="identificador" value="valor" name="nombre"/>
```

### Campos Ocultos

Otra de las opciones que nos podemos encontrar dentro de los formularios es con la necesidad de enviar información oculta. Es decir, información que tiene que ir anexa al formulario pero que no necesita ser introducida por el usuario. Para ello tenemos la posibilidad de crear campos de datos ocultos.

La sintaxis para los campos de entrada ocultos es:

```
<input type="hidden" id="identificador" value="valor" name="nombre"/>
```

En estos casos el campo valor siempre va relleno, ya que no hay otra forma de que el usuario le asigne un valor.

### Imágenes

Uno de los tipos de elementos input es el tipo “image”. Mediante el tipo “image” podemos crear un botón de envío que sea una imagen. La imagen se cargará mediante el atributo src. La estructura para elementos input de este tipo es:

```
<input type="image" src="url-image" name="nombre" alt="texto-alternativo"/>
```

Como sucede cada vez que manipulamos imágenes hay que rellenar el atributo alt con un texto alternativo por motivos de accesibilidad.

### Áreas de texto

Un elemento algo más avanzado que el campo de entrada de datos es el área de texto. Mediante un área de texto tenemos la capacidad de que el usuario inserte una gran cantidad de datos y además que esos datos puedan estar en diferentes líneas.

Para poder insertar un área de texto en nuestro formulario utilizamos el elemento `textarea`. La sintaxis del elemento `textarea` será la siguiente:

```
<textarea rows="numerofilas" cols="numerocolumnas"
name="nombre"></textarea>
```

A diferencia del elemento `input`, el `textarea` tiene una etiqueta de inicio y una de fin.

Los atributos que nos encontramos en un `textarea` son:

- **rows**, indica el número de filas que tiene el área de texto.
- **cols**, indica el número de columnas que tiene el área de texto.
- **name**, al igual que sucede con otros elementos del formulario, `name` contiene el nombre del campo, el cual será enviado al servidor para ser procesado.

De esta forma, si queremos crear un área de texto de 20 filas por 100 columnas en el que un usuario pueda insertar un comentario tendríamos el siguiente código:

```
<textarea rows="20" cols="100" name="comentario"></textarea>
```

### Combos de opciones

Otro elemento que podemos insertar dentro de un formulario es un combo de opciones. Es decir, un elemento en el que el usuario pueda elegir un elemento o varios de una lista.

El elemento que nos permite insertar un combo de opciones es `select`. La sintaxis básica de un elemento `select` es:

```
<select name="nombre" id="nombre">
  <option label="etiqueta" value="valor"></option>
</select>
```

Dónde el atributo label indica el texto que aparecerá para poder ser seleccionado en el combo y value el valor realmente de ese item.

Si queremos que una de las opciones del combo vaya marcada recurriremos al atributo selected. Así, en nuestro ejemplo si marcamos como predefinido el equipo del Betis tendríamos el siguiente código:

```
<select>
  <option>Atletico de Madrid</option>
  <option selected="selected">Betis</option>
  <option>FC. Barcelona</option>
  <option>Real Madrid</option>
  <option>Zaragoza</option>
</select>
```

### Botones

Una vez que hemos insertado campos de texto en nuestro formulario es hora de insertar botones. Mediante los botones podremos realizar operaciones de envío del formulario, de manipulación de datos, borrado.

Existen dos formas de insertar botones dentro de un formulario: el elemento input y el elemento button. La primera es recurrir nuevamente al elemento input que hemos visto anteriormente para los campos de texto.

### Botones de envío

En el caso del elemento input podemos poner botones de otras dos formas y en estos casos ya con funcionalidad. Estos son los tipos "submit" y "reset". Para crear un botón que nos envíe los datos del formulario al servidor tenemos el tipo submit. Su sintaxis es la siguiente:

```
<input type="submit" value="TextoBotón"/>
```

Una vez que pulsemos sobre el botón se enviarán los datos que el usuario haya insertado en el formulario.

El otro tipo de botón con funcionalidad es el que nos permite el borrado de los datos del formulario. Para ello tenemos el tipo "reset". La sintaxis de este botón será:

```
<input type="reset" value="TextoBotón"/>
```

Cuando el usuario pulse sobre el botón de borrado. Todos los valores que el usuario haya insertado en el formulario se eliminarán.

### El elemento button

Como hemos visto hasta ahora los botones que hemos insertado han sido mediante el elemento input, si bien contamos con otro elemento para poner botones en el formulario que es el elemento button. Cuya funcionalidad es la misma que la del elemento input. La sintaxis del elemento button es:

```
<button name="nombre" type="TipoBoton" value="ValorBoton"></button>
```

Dependiendo del tipo que asignamos al atributo type obtendremos un comportamiento u otro:

- **submit**, crea un botón para el envío de formulario.
- **reset**, crea un botón para el borrado de datos del formulario.
- **button**, crea un botón normal.

## 8.0 Tablas

En documentos HTML una tabla puede ser considerada, resumidamente, como un grupo de filas donde cada una contiene a un grupo de celdas. Una tabla básica puede ser declarada usando tres elementos, a saber, **<table>** (el contenedor principal), **<tr>** (representando a las filas contenedoras de las celdas) y **<td>** (representando a las celdas).

```
<table class="default" border="1">  
  <tr>  
    <td>Celda 1</td>  
    <td>Celda 2</td>  
    <td>Celda 3</td>  
  </tr>  
  <tr>  
    <td>Celda 4</td>  
    <td>Celda 5</td>  
    <td>Celda 6</td>  
  </tr>  
</table>
```

Una celda de encabezado es un tipo especial de celda utilizada para organizar y categorizar otras celdas en la tabla. Dicho esto, es difícil imaginar una tabla donde una celda de encabezado no tenga utilidad. Casi cualquier tabla puede beneficiarse de un grupo de celdas de encabezado bien ubicado. Aquí, las celdas

de encabezado, representadas por el elemento **<th>**, son ubicadas en la primera fila de la tabla, encima de las celdas comunes.

```
<table class="default" border="1">
  <tr>
    <th></th>
    <th>Hoy</th>
    <th>Mañana</th>
    <th>Domingo</th>
  </tr>
  <tr>
    <th>Clima</th>
    <td>Soleado</td>
    <td>Mayormente soleado</td>
    <td>Parcialmente nublado</td>
  </tr>
  <tr>
    <th>Temperatura</th>
    <td>19°C</td>
    <td>17°C</td>
    <td>12°C</td>
  </tr>
</table>
```

### Título de Tabla

El título de una tabla es otro componente que aporta a la legibilidad. La idea de un título es la de proveer una descripción concisa acerca de la información presentada en la tabla o su propósito, aportando contexto y haciéndola significativamente más fácil de comprender. En HTML, los títulos de las tablas son insertados mediante el elemento **<caption></caption>**, justo después de la etiqueta de apertura de la misma y antes que todos los demás elementos de su estructura.

```
<caption>Estatus del clima</caption>
```

### Unir celdas

La unificación de celdas es un mecanismo mediante el cual un autor puede combinar o fusionar dos o más celdas adyacentes (td y th) en una tabla. Este efecto puede lograrse utilizando los atributos colspan, para unificación horizontal, y

rowspan, para unificación vertical. Estos atributos pueden tomar un valor entero mayor a cero, que representa el número de celdas que participarán en la unificación.

```
<table class="default" border="1">
```

```
<tr>
```

```
<th></th>
```

```
<th>Día 1</th>
```

```
<th>Día 2</th>
```

```
<th>Día 3</th>
```

```
<th>Día 4</th>
```

```
</tr>
```

```
<tr>
```

```
<th>Mike (lastimado)</th>
```

```
<td colspan="3">0 km</td>
```

```
<td>4 km</td>
```

```
</tr>
```

```
<tr>
```

```
<th>Susan</th>
```

```
<td>23 km</td>
```

```
<td>18 km</td>
```

```
<td>19 km</td>
```

```
<td>15 km</td>
```

```
</tr>
```

```
</table>
```

### Organización

HTML provee tres elementos que pueden ayudar a los autores a organizar la información presentada en una tabla. Éstos son: **<thead>**, para agrupar las filas que presentan información de encabezado; **<tfoot>** para contener a las filas que representan un pie o resumen; y **<tbody>**, para representar un bloque de filas que consiste en un cuerpo de datos.

```
<table class="default" border="1">
```

```
<thead>
```

```
<tr>
```

```
<th></th>
```

```
<th colspan="2">Anthony</th>
```

```
<th colspan="2">Lione</th>
```

```
</tr>
```

```
<tr>
```

```

<th>Fecha</th>
<th>Entrada</th>
<th>Salida</th>
<th>Entrada</th>
<th>Salida</th>
</tr>
</thead>
<tbody>
<tr>
<th>15/07</th>
<td>$200,00</td>
<td>$50,00</td>
<td>$0</td>
<td>$0</td>
</tr>
<tr>
<th>28/07</th>
<td>$0,00</td>
<td>$100,00</td>
<td>$100,00</td>
<td>$0,00</td>
</tr>
<tr>
<th>09/08</th>
<td>$0,00</td>
<td>$50,00</td>
<td>$0,00</td>
<td>$80,00</td>
</tr>
<tr>
<th>22/08</th>
<td>$40,00</td>
<td>$0,00</td>
<td>$0,00</td>
<td>$110,00</td>
</tr>
<tr>
<th>25/08</th>
<td>$0,00</td>
<td>$20,00</td>
<td>$0,00</td>
<td>$0,00</td>
</tr>
</tbody>

```

```
<tfoot>
<tr>
<th>Balance</th>
<td colspan="2">$20,00</td>
<td colspan="2">$10,00</td>
</tr>
</tfoot>
</table>
```

### 9.0 Secciones

Las secciones son las partes temáticas en las que se puede dividir el contenido de una página web. En HTML 4 / XHTML 1 la única etiqueta para definir agrupaciones de contenido era prácticamente la etiqueta `<div>`. Por ello, cualquier página de tamaño medio acababa teniendo muchos elementos `<div>`, en algunos casos para organizar el contenido de la página, pero también para aplicar estilos de carácter decorativo. El código fuente resultaba así difícil de entender y mantener.

En HTML 5 se ha querido mejorar esa situación introduciendo varias etiquetas para definir diferentes tipos de secciones: **`<article>`**, **`<section>`**, **`<nav>`**, **`<aside>`**, **`<header>`** y **`<footer>`**

Al utilizar etiquetas específicas para tipos de contenido específico, se mejora la legibilidad y facilidad de reutilización tanto del código HTML como de las hojas de estilo.

**Cuerpo del documento: `<body>`** La etiqueta `<body>` abarca todo el contenido de la página web que se visualiza en el navegador. Por ello, una página web sólo puede tener un elemento `<body>`.

**Encabezamiento: `<header>`** La etiqueta `<header>` está pensada para agrupar el contenido inicial de una página web. Puede contener únicamente los elementos que aparecen en las páginas de un sitio (el logotipo y un menú general, por ejemplo) o incluir también elementos particulares de la página (el título principal `<h1>` de la página, enlaces de navegación dentro de la página, texto de introducción, por ejemplo).

El elemento `<header>` puede estar incluido en `<body>` y entonces se entiende que es el encabezado de la página en su conjunto, pero también se puede incluir en `<article>`, `<section>`, `<nav>` y `<aside>` y entonces se entiende que es el encabezado del elemento que lo contiene.

**Navegación: <nav>** La etiqueta <nav> está pensada para agrupar los enlaces a otras páginas o los enlaces a los apartados de la propia página.

Una página puede contener varios elementos <nav>, aunque no es necesario que todos los grupos de enlaces estén en elementos <nav> (por ejemplo, los enlaces de contacto que se suelen incluir en el elemento <footer> no requieren el uso de <nav>).

- El <nav> de <header> puede contener enlaces globales a otras páginas de sitio web.
- El <nav> de <body> puede contener enlaces a las secciones de la página.
- El <nav> de <aside> puede contener enlaces a otros sitios webs relacionados con el contenido de la página.

**Sección: <section>** La etiqueta <section> está pensada para agrupar los apartados del contenido principal del documento. Normalmente, las secciones <section> empiezan con un título (<h1>, <h2>, etc.) que hace referencia al tema tratado en la sección.

**Artículo: <article>** La etiqueta <article> está pensada para agrupar el contenido de la página que forma una unidad en sí misma desde el punto de vista temático. Es decir, que un artículo debería poder publicarse y leerse como documento independiente, aunque una página puede estar formada por varios artículos de temática relacionada o no.

Un elemento <article> puede contener otros elementos <article> que están relacionados con el contenido del <article> que los contiene.

**Lateral: <aside>** La etiqueta <aside> está pensada para agrupar contenido secundario y tangencial al contenido al que acompaña (por ejemplo, un bloque de anuncios, un grupo de enlaces externos relacionados, una cita del texto). Normalmente se suele mostrar en los lados del documento.

El elemento <aside> puede estar incluido en <body> y entonces se entiende que se trata de contenido secundario con respecto a la página en su conjunto, pero también se puede incluir en <article> o <section> y entonces se entiende que se trata de contenido secundario con respecto al elemento que lo contiene.

**Pie: <footer>** La etiqueta <footer> está pensada para contener información general sobre el documento, información que se suele poner al final del documento: autor, direcciones de contacto, licencia o condiciones de uso, enlaces a otros documentos relacionados, etc.

El elemento <footer> puede estar incluido en <body> y entonces se entiende que es el pie de página de la página en su conjunto, pero también se puede incluir en <article>, <section>, <nav> y <aside> y entonces se entiende que es el pie de página del elemento que lo contiene.

### 10.0 Que es CSS

CSS significa Cascade Style Sheets, que se traduce al español como “Hojas de Estilo en Cascada”.

CSS es un mecanismo complementario del lenguaje HTML que permite indicarle al navegador el estilo que debe darles a los distintos elementos al desplegar la información de un sitio web.

Mientras que los comandos HTML le indican al navegador que ciertas partes del texto son títulos, subtítulos, enlaces o párrafos, CSS es lo que le dice, entre otras cosas, qué fuente, tamaño, color o alineación debe tener un elemento de la página.

#### Componentes de una hoja de estilo en CSS

Una hoja de estilo se compone de un listado de reglas o comandos que constan de un selector y una declaración:

- Selector: Es el elemento sobre el cuál va a aplicarse cierto estilo, puede ser un H1, H2, un párrafo, un formulario, etc.
- Declaración: Se compone de una propiedad (color, ancho, fuente, etc) y un valor (rojo, verdana, alineación izquierda, etc).

Si tenemos, por ejemplo:

```
h1 {  
    color: red;  
    text-align: center;  
}
```

Le estamos indicando al navegador que despliegue el Título 1 del documento en color rojo y centrado.

Si tenemos:

```
p {  
    font-size: 60px;  
    color: black;
```

```
text-align: left  
}
```

Le estamos diciendo que despliegue los párrafos con un tamaño de fuente de 60px, en color negro y con alineación a la izquierda.

### Formas de uso

1 – `<h1 style="color:blue; font-size:14px;">Esto es un encabezado</h1>`

2 – 

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {color:blue; font-size:14px;}  
</style>  
</head>  
<body>  
<h1> Esto es un encabezado </h1>  
<p>Esto es un párrafo</p>  
</body>  
</html>
```

3 – **.HTML**

```
<!DOCTYPE html>  
<html>  
<head>  
<link rel="stylesheet" href="stylesheet.css">  
</head>  
<body>  
<h1 id="id1" class="clase1"> Esto es un encabezado </h1>  
<p>Esto es un párrafo</p>  
</body>  
</html>
```

### **.CSS**

#### **Elemento**

```
H1 {  
    color: blue;  
    font-size:14px;  
}
```

### Identificación

```
#id1 {  
    color: blue;  
    font-size:14px;  
}
```

### Clase

```
.clase1 {  
    color: blue;  
    font-size:14px;  
}
```

### Propiedades de CSS

**Font-family:** Define la familia tipográfica. Es conveniente poner una lista de dos o tres tipografías separadas por coma, porque si el usuario no tiene instalada la tipografía que nosotros elegimos, el navegador opta por mostrar la siguiente que debería ser una similar, si tampoco la tiene instalada, mostrará la tipografía por defecto.

**Font-size:** Define el tamaño de la fuente y el valor se puede escribir en pixels. El valor por defecto de los textos en todos los navegadores es de 16px.

**Color:** Define el color de la tipografía. Los colores se pueden escribir de 3 formas distintas: con sistema hexadecimal, por ejemplo: #FF0000 (es rojo). Con los nombres de los colores (más limitado) por ejemplo: black, red, green. O usando RGB, esta paleta permite agregar el canal alfa para hacer transparencias.

**Width:** Define el ancho de un elemento, el valor se puede escribir en pixels, ems o porcentaje.

**Max-width** o min-width: Definen el ancho máximo o mínimo de un elemento. Muy importante en sitios adaptables

**Height:** Define el alto de un elemento, el valor se puede escribir en pixels, ems o porcentaje.

**Max-height** o min-height: Definen el alto máximo o mínimo de un elemento. Muy importante en sitios adaptables

**Padding:** Es la distancia desde el borde de un elemento hasta su contenido.

**Margin:** Es la distancia entre un elemento y otro (desde el borde de un elemento hacia afuera).

**Border:** Define el borde de un elemento, su color, su estilo y grosor.

**Background:** Define los fondos de un objeto. El fondo puede ser una imagen o un color. El color puede ser pleno o degradado. La imagen se puede repetir formando una trama (es lo que ocurre por defecto) o se puede especificar que no repita y que se coloque en determinada posición.

### 11.0 Uso de CSS desde y fuera de un documento HTML

La forma más ordenada para llamar un archivo **CSS** desde nuestro documento **HTML**, es crear una carpeta llamada **css** en nuestro directorio raíz, y dentro de ella depositamos todo nuestros archivos **css**.

La llamada de este archivo desde nuestro **HTML** lo haremos de la siguiente manera:

```
<head>  
  <link rel="stylesheet" href="css/styles.css">  
</head>
```

Dentro de nuestra cabecera **HTML**, con la etiqueta **link** llamamos nuestro archivo, primero accedemos a nuestra carpeta y después de la / llamamos nuestro archivo **css**.

### 12.0 Bootstrap

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en **HTML** y **CSS**, así como extensiones de **JavaScript** adicionales. A diferencia de muchos **frameworks** web, solo se ocupa del desarrollo front-end.

Página oficial : <https://getbootstrap.com/>