

## Tema 4

# Lenguaje de marcas HTML (III)

### Objetivos

- Reconocer la utilidad del lenguaje HTML en los sistemas de gestión de la información.
- Conocer la funcionalidad de las principales etiquetas y atributos del lenguaje HTML.
- Conocer y aplicar el uso de las tablas, los marcos y los formularios para la construcción de páginas y sitios web, así como la identificación y agrupación de los elementos de la página.

### Introducción

En el tema anterior se estudiaron algunos de los elementos más utilizados a la hora de construir documentos HTML, así como sus etiquetas y atributos. En este tema se continúa avanzando en el estudio del lenguaje HTML en relación con la construcción de páginas y sitios Web. El estudio del lenguaje HTML se ha estructurado en varios temas debido a su amplitud, facilitándose de este modo su aprendizaje de una forma más detallada, ordenada y concreta. Estos temas se han planificado de forma consecutiva dentro de la periodificación de contenidos del módulo profesional.

### Índice

4.1 TABLAS	2
4.2 MARCOS	5
4.3 IDENTIFICACIÓN Y AGRUPACIÓN DE ELEMENTOS	6
4.4 FORMULARIOS	10
4.5 OTROS ELEMENTOS Y ETIQUETAS	18
Glosario de términos	20

## 4.1 TABLAS

Las tablas facilitan la presentación de información en formato tabular, es decir, organizada en filas y columnas. Las tablas en HTML son estructuras que están formadas por varias filas, cada una de las cuales incluye un conjunto de celdas.

Las tablas se crean con la etiqueta <table>. Dentro de las etiquetas de apertura y cierre, se puede colocar la etiqueta <caption> que permite incluir un título a la tabla. A continuación, se identifica cada fila de la tabla con la etiqueta <tr>. Finalmente, cada celda dentro de una fila se identifica con la etiqueta <td>. La etiqueta <th> permite definir celdas de encabezado. Por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Tablas - Ejemplo 1</title>
  <meta charset="utf-8" />
</head>
<body>
  <table border="1">
    <caption>Horario de comercio</caption>
    <tr>
      <th></th>
      <th>Mañana </th>
      <th>Tarde </th>
      <th>Sábado </th>
    </tr>
    <tr>
      <td>Invierno </td>
      <td>9:00-13:30 </td>
      <td>16:30-20:00 </td>
      <td>10:00-21:00</td>
    </tr>
    <tr>
      <td>Verano </td>
      <td>9:30-14:00 </td>
      <td>17:30-21:00 </td>
      <td>10:00-21:00</td>
    </tr>
  </table>
</body>
</html>
```

El ejemplo anterior, produce el siguiente resultado en la ventana de visualización del navegador:

Horario de comercio			
	Mañana	Tarde	Sábado
Invierno	9:00-13:30	16:30-20:00	10:00-21:00
Verano	9:30-14:00	17:30-21:00	10:00-21:00

Los atributos más importantes de la etiqueta <table> son los siguientes:

- width. Anchura total de la tabla en píxeles o porcentaje.
- border. Indica la anchura del borde de la tabla en píxeles. El valor por defecto es 0.
- rules. Indica qué líneas de división entre las celdas serán visibles. Los posibles valores son:
  - none: ninguna, es el valor por defecto.
  - groups: solo entre grupos de filas o grupos de columnas.
  - rows, cols: solo entre filas, solo entre columnas.
  - all: todas.
- frames. Indica qué lados de la tabla serán visibles. Sus posibles valores son:
  - void: ninguna, es el valor por defecto.
  - above, below, lhs, rhs: solo arriba, solo abajo, solo izquierda, solo derecha.
  - box: todos los lados.
- cellpadding. Especifica el espacio entre el borde de una celda y su contenido.
- cellspacing. Especifica el espacio entre celdas.

Los atributos más importantes de la etiqueta <tr> que se utiliza para crear una fila dentro de una tabla son los siguientes:

- align. Indica la alineación horizontal del texto a nivel de fila. Sus posibles valores son:
  - left: alineado a la izquierda.
  - right: alineado a la derecha.
  - center: centrado.
- valign. Indica la alineación vertical del texto a nivel de fila. Sus posibles valores son:
  - top: alineado arriba.
  - middle: alineado al medio de la fila
  - bottom: alineado abajo.

Los atributos más importantes de las etiquetas <td> y <th> que se utilizan para crear celdas dentro de una fila son los siguientes:

- align. Indica la alineación horizontal del texto a nivel de celda. Sus posibles valores son los mismos: left, right y center.
- valign. Indica la alineación vertical del texto a nivel de celda. Sus posibles valores son los mismos: top, middle y bottom.
- rowspan. Indica el número de filas que ocupa la celda (combinar filas).
- colspan. Indica el número de columnas que ocupa la celda (combinar columnas).

La mayor parte de los atributos anteriores han sido declarados obsoletos o en desuso en las últimas versiones de HTML. En especial, aquellos que se refieren a los aspectos de la apariencia visual de la tabla. Consecuentemente, se utilizarán las características de las hojas de estilo en cascada para especificar los efectos de apariencia deseados sobre las tablas.

Para hacer que una celda ocupe varias celdas, se utilizan los atributos `rowspan` y `colspan` dentro de la etiqueta `<td>`, o bien, dentro de la etiqueta `<th>` si se utilizan las celdas de encabezado. El atributo `rowspan` permite que una celda ocupe un número de filas concreto. El atributo `colspan` permite que ocupe varias columnas. Por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Tablas - Ejemplo 2</title>
  <meta charset="utf-8" />
</head>
<body>
  <table border="1">
    <caption>Horario de comercio</caption>
    <tr>
      <th></th>
      <th>Mañana </th>
      <th>Tarde </th>
      <th>Sábado </th>
    </tr>
    <tr>
      <td>Invierno </td>
      <td>9:00-13:30 </td>
      <td>16:30-20:00 </td>
      <td rowspan="2" valign="middle">10:00-21:00</td>
    </tr>
    <tr>
      <td>Verano </td>
      <td>9:30-14:00 </td>
      <td>17:30-21:00 </td>
    </tr>
    <tr>
      <td colspan="4">Domingos cerrado </td>
    </tr>
  </table>
</body>
</html>
```

El ejemplo anterior, produce el siguiente resultado en la ventana de visualización del navegador:

Horario de comercio			
	Mañana	Tarde	Sábado
Invierno	9:00-13:30	16:30-20:00	10:00-21:00
Verano	9:30-14:00	17:30-21:00	
Domingos cerrado			

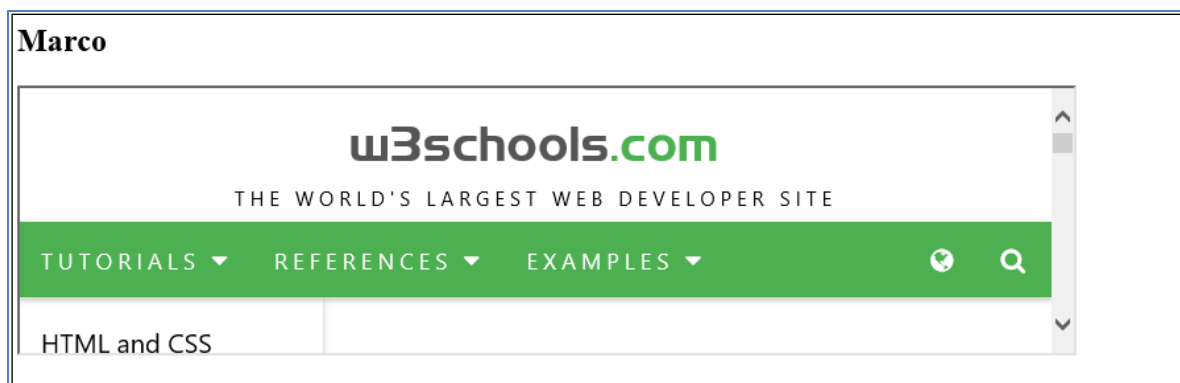
Solo se han estudiado las características más básicas de las tablas. Existen más etiquetas relacionadas con el uso avanzado de las tablas en HTML. Actualmente, la utilización de las tablas en documentos HTML está en declive, debido a los problemas que plantea su uso en relación con el cumplimiento de las normas de Accesibilidad Web.

## 4.2 MARCOS

La etiqueta <iframe> inserta un marco en línea, también denominado marco flotante, dentro de un documento, es equivalente a insertar una página dentro de otra con el tamaño establecido para el marco. El marco en línea es un elemento que se utiliza de la misma forma que cualquier otro elemento dentro del documento. Se suele utilizar para mostrar publicidad. Por ejemplo:

```
<h2>Marco</h2>
<iframe name="iframe_a" src="http://www.w3schools.com/" width="800" height="200"
        frameborder="1" scrolling="auto"></iframe>
```

El ejemplo anterior, produce el siguiente resultado en la ventana de visualización del navegador:



Los atributos más importantes de la etiqueta <iframe> son:

- name. Especifica el nombre que se asigna al marco.
- src. Especifica la URI del contenido del marco.
- frameborder. Especifica el ancho del borde.
- scrolling. Especifica si se muestran las barras de desplazamiento.
- width. Especifica el ancho de marco.
- height. Especifica el alto del marco.

Un elemento <iframe> puede ser usado como marco destino de un enlace. Para ello, en el atributo target del enlace se especifica valor del atributo name del marco correspondiente. Por ejemplo:

```
<h2>Marco como destino de un enlace</h2>
<p>
  <a href="http://www.w3schools.com" target="iframe_b">W3Schools.com</a>
  <br /> <br />
  <iframe name="iframe_b" src="Tablas1.html" width="800" height="200"
    frameborder="1" scrolling="auto"></iframe>
</p>
```

En el ejemplo anterior, inicialmente se muestra el contenido de la página Tablas1.html en el marco denominado "iframe\_b". Ahora bien, al hacer clic sobre el enlace se mostrará el contenido de la página <http://www.w3schools.com> en este mismo marco. El ejemplo anterior, produce el siguiente resultado en la ventana de visualización del navegador:

**Marco como destino de un enlace**  
[W3Schools.com](http://www.w3schools.com)  

	<b>Mañana</b>	<b>Tarde</b>	<b>Sábado</b>
Invierno	9:00-13:30	16:30-20:00	10:00-21:00
Verano	9:30-14:00	17:30-21:00	10:00-21:00

Antiguamente, la mayoría de las páginas y sitios Web se estructuraban utilizando marcos. En la actualidad, el uso de marcos ha sido abandonado casi completamente, debido a los problemas que plantea el uso de marcos en relación con el cumplimiento de las normas de Accesibilidad Web. Y, también, porque han aparecido nuevos elementos y técnicas en el lenguaje HTML que permiten realizar de forma fácil, un control efectivo y de la estructura de los contenidos en las páginas Web.

### 4.3 IDENTIFICACIÓN Y AGRUPACIÓN DE ELEMENTOS

En este apartado se comienza a estudiar los fundamentos del diseño Web más actual que se basa en estructurar el contenido de las páginas Web mediante divisiones. Las divisiones son secciones independientes de código HTML cuya apariencia se especifica utilizando las características de las hojas de estilo.

Las tendencias más actuales de diseño Web se apoyan en el denominado modelo de cajas, en el cual el uso de divisiones juega un papel básico. La identificación y agrupación de los elementos de la página son aspectos clave para poder estructurar el contenido de las páginas Web en divisiones y, consecuentemente, poder aplicar el modelo de cajas en la construcción de páginas Web. En los próximos temas, se continuará avanzando los diversos aspectos que intervienen en el diseño Web basado en el modelo de cajas.

### Identificación de elementos: atributos id y class

Para identificar a cada elemento de la página de forma individualizada, se le asigna un nombre que permita hacer referencia a él. Esto se consigue mediante el **atributo id**. Este atributo se puede utilizar con cualquier etiqueta de HTML. El atributo id permite asignar un nombre o identificador a un elemento concreto de la página. Este identificador ha de ser único dentro del documento HTML, es decir, no puede haber dos elementos con el mismo identificador. Por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Identificación y agrupación de elementos</title>
  <meta charset="utf-8" />
</head>
<body>
  <div id="contenedor">
    <div id="titulo">
      <!-- Aquí se incluye el código HTML de la cabecera-->
      Una nueva disciplina, la automática.
    </div>
    <div id="contenido">
      <!-- Aquí se incluye el código HTML de la cabecera-->
      <p>
        En sus "Ensayos sobre automática" publicados por primera vez en 1914, Torres
        Quevedo formula lo que será en adelante una nueva rama de la ingeniería, la
        automática. En este texto propone agregar a la teoría de máquinas una sección
        especial, la automática, que examinara los procedimientos que pueden aplicarse
        a la construcción de autómatas dotados de una vida más o menos complicada.
      </p>
      <p>
        Con el desarrollo del Telekino, Torres Quevedo llegó a la conclusión de que con
        él no sólo había fabricado el primer control remoto de la historia, sino que
        esta máquina, era en sí un autómata, es decir, una máquina que podía funcionar
        de forma autónoma ejecutando acciones y respondiendo a órdenes en función de
        ciertas circunstancias de su entorno.
      </p>
      <p>
        Partiendo de esta conclusión, Torres explotó las posibilidades que le brindaba
        esta nueva rama de la teoría de máquinas y la aplicó al desarrollo de máquinas
        de cálculo. Gracias a ello, pudo salvar las numerosas dificultades que hasta
        entonces había planteado la creación de estas máquinas por métodos
        exclusivamente mecánicos, y donde Charles Babbage no obtuvo ningún resultado
        práctico satisfactorio. En sus "Ensayos sobre automática", Torres Quevedo
        desarrolla la teoría de lo que posteriormente será su aritmómetro: una máquina
        electromecánica capaz de realizar cálculos de forma autónoma con un dispositivo
        de entrada de comandos, una unidad de procesamiento y registros de valores y,
        un dispositivo de salida. Es, en definitiva, lo que debería consagrar
        internacionalmente al ingeniero español como el inventor del primer ordenador
        de la historia, tal como son concebidos actualmente.
      </p>
    </div>
  </div>
</body>
</html>
```

Por otra parte, el **atributo class** permite asignar un nombre de una clase de estilo a un elemento de la página para especificar de este modo su apariencia visual. En los próximos temas el atributo class se utilizará ampliamente. Este atributo se puede utilizar con cualquier etiqueta de HTML, de modo que puede asignarse una clase de estilo concreta a cualquier elemento de la página. Además, puede haber más de un elemento que pertenezca a la misma clase de estilo y un elemento de la página puede pertenecer a varias clases de estilo. Por ejemplo:

```
<div id="cabecera" class="cab">  
    El Nuevo Diseño de Webs  
</div>
```

Los elementos <div> y <span>, que se estudian a continuación, son particularmente útiles para combinarse con los formatos de estilo visual definidos empleando las características de las hojas de estilo. Las hojas de estilo se estudian con profundidad en los siguientes temas.

### Agrupación de elementos: etiquetas <div> y <span>

La **etiqueta <div>** sirve para establecer divisiones en la página para repartir en secciones o bloques de los contenidos de un documento HTML. Así, se pueden agrupar varios elementos dentro de una sección y, de esta forma, poder asignarle un identificador, mediante el atributo id, y una clase de estilo, mediante el atributo class. Funciona de forma parecida al elemento párrafo, pero a diferencia de este no provoca un salto de línea de separación con el elemento anterior y posterior. El elemento división es un elemento de bloque por lo que puede contener elementos de línea y otros elementos de bloque. Un elemento división puede contener, a su vez, otros elementos división. Por ejemplo:

```
<h2>DIV y CLASS</h2>  
<!-- Primer bloque o sección del documento llamado cabecera -->  
<div id="cabecera" class="cab">  
    <h2>LAS CAPAS</h2>  
    <p>El Nuevo Diseño de Webs</p>  
</div>  
<!-- Segunda sección del documento llamado pie -->  
<div id="cuerpo" class="texto">  
    <p>  
        Las Capas tienen muchas ventajas frente a los marcos  
        a la hora de diseñar páginas webs.  
    </p>  
    <div>  
        Características:  
        <ul>  
            <li>Permiten estructurar los contenidos en los documentos</li>  
            <li>Es más sencillo su mantenimiento</li>  
            <li>Facilitan la asignación de estilos a los elementos</li>  
            <li>Permiten la construcción de páginas Web accesibles y adaptativas</li>  
        </ul>  
    </div>  
</div>  
<!-- Última sección del documento llamado pie -->  
<div id="pie" class="pie2">  
    <div>Lenguajes de Marcas y Sistemas de Gestión de la Información </div>  
    <div>IES mare Nostrum de Alicante</div>  
</div>
```



En el código anterior puede apreciarse que se han asignado las clases de estilo denominadas cab, texto y pie2 a las divisiones identificadas como cabecera, cuerpo y pie, respectivamente. El código del ejercicio anterior está incompleto, porque no se ha incluido la especificación de las clases de estilo correspondientes. En los próximos temas se estudiará cómo crear clases de estilo y cómo definir las características de estilo asociadas a estas. El ejemplo anterior, podría producir el siguiente resultado en la ventana de visualización del navegador:

## DIV y CLASS

### LAS CAPAS

El Nuevo Diseño de Webs

Las Capas tienen muchas ventajas frente a los marcos a la hora de diseñar páginas webs.

Características:

- Permiten estructurar los contenidos en los documentos
- Es más sencillo su mantenimiento
- Facilitan la asignación de estilos a los elementos
- Permiten la construcción de páginas Web accesibles y adaptativas

Lenguajes de Marcas y Sistemas de Gestión de la Información  
IES mare Nostrum de Alicante

La **etiqueta <span>** sirve para identificar un elemento de línea, que está contenido dentro de un elemento de bloque o de línea, y de esta forma asignarle un identificador y una clase de estilo. A través de la etiqueta <span> se puede especificar, entre otras características de estilo, el tamaño de la fuente, el color, el alto de línea, la posición del texto dentro de la línea, etc. a un elemento de menor longitud, como una frase o una palabra, dentro de un párrafo. Por ejemplo:

```
<h2>Hojas de estilo en cascada</h2>
<div id="contenido">
  Las <span class="negrita">hojas de estilo en cascada</span> o CSS, en inglés
  <span class="cursiva">Cascading Style Sheets</span>, es un lenguaje usado para
  definir y crear la presentación de un documento estructurado, escrito en HTML
  o XML. El <span class="cursiva">World Wide Web Consortium (W3C)</span> es el
  organismo encargado de formular la especificación de las hojas de estilo que
  servirán de estándar para los navegadores.
</div>
```

En el código anterior no se ha incluido la especificación de las características de estilo negrita y cursiva. Sin embargo, sirve perfectamente como un ejemplo de demostración de la utilización de la etiqueta <span>. El ejemplo anterior, podría producir el siguiente resultado en la ventana de visualización del navegador:

## Hojas de estilo en cascada

Las **hojas de estilo en cascada** o CSS, en inglés *Cascading Style Sheets*, es un lenguaje usado para definir y crear la presentación de un documento estructurado, escrito en HTML o XML. El *World Wide Web Consortium (W3C)* es el organismo encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores.

## 4.4 FORMULARIOS

Cuando se utiliza el servicio de información Web, lo más inmediato es pensar que la información fluirá, principalmente, desde el servidor Web hacia el cliente o navegador. Este flujo de información proveniente del servidor Web contiene la información solicitada por el usuario a través del navegador. Sin embargo, los formularios permiten que la información fluya desde el cliente Web hacia el servidor Web. Se trata, por tanto, de un elemento de la página que permite a los usuarios enviar información hacia el servidor Web y, por tanto, facilitan que los usuarios puedan interactuar en el servicio de información Web. Los servidores Web pueden incluir los procedimientos adecuados para que la información proveniente del cliente Web a través de formularios sea procesada de manera adecuada y produzca los resultados deseados. El ámbito de estudio de este módulo profesional solo se refiere a la construcción de formularios empleando HTML.

Los formularios son, por tanto, un elemento importante e HTML. Su importancia radica en que convierten al servicio de información Web en un medio de comunicación bidireccional e interactivo, puesto que fueron diseñados para que el usuario pudiera enviar datos hacia el servidor Web.

Un formulario es un elemento de un documento HTML que contiene, a su vez, una serie de elementos característicos denominados controles o campos. Los usuarios que recuperan un documento HTML que contiene un formulario pueden cumplimentar sus campos introduciendo texto, marcando casillas de verificación, seleccionando opciones, etc. y, posteriormente, pueden enviar la información introducida hacia un componente de software para que sea procesada por este en el entorno del servidor Web.

Al enviar un formulario, la información introducida se envía como pares de nombre-valor para cada control del formulario, por lo que cada control del formulario debe tener un nombre único. El valor de cada control es, precisamente, la información introducida por el usuario en ese control.

### Crear un formulario

Un formulario se crea mediante la **etiqueta <form>**. Los formularios son elementos de bloque, por lo que pueden contener otros elementos de bloque o de línea. En efecto, un formulario contendrá los controles que lo formen y, además, puede contener otros elementos de código HTML, por ejemplo: listas, imágenes, divisiones, etc. Los contenidos de un formulario deben incluirse entre la etiqueta de apertura y de cierre del formulario.

Los atributos más importantes de la etiqueta <form> son los siguientes:

- **id.** Especifica el nombre o identificador del formulario. También se puede utilizar el atributo **name**, aunque se recomienda utilizar el atributo **id** para identificar un formulario.
- **action.** Indica la URI del componente de software que procesará la información enviada por el formulario una vez haya sido completado y enviado. Aunque es menos habitual, también pueden enviarse los datos del formulario a una dirección de correo, aunque en estos casos no existe procesamiento de datos en ningún servidor, tan solo envío de datos.
- **method.** Indica el método de envío de los datos del formulario hacia el servidor. Existen dos métodos: **GET** y **POST**. Por razones de seguridad y funcionalidad en el envío se recomienda siempre el uso del método de envío **POST**. El valor predeterminado es **GET**.
- **enctype.** Especifica el formato de codificación de los datos del formulario durante el envío hacia el servidor, cuando el método de envío empleado es **POST**. El valor predeterminado de este atributo es **"application/x-www-form-urlencoded"** que representa un formato de codificación seguro y ampliamente aceptado y utilizado.
- **accept-charset.** Indica una codificación de caracteres que debe ser aceptada por el servidor para poder manejar este formulario.

El siguiente ejemplo muestra la especificación de un formulario que utiliza una dirección de correo electrónico como destino del envío.

```
<form id="formulario1" action="mailto:correo@correo.es?subject=asunto"
      method="post" enctype="text/plain" accept-charset="utf-8">

    ...Contenido del formulario

</form>
```

### Controles de los formularios

A continuación, se estudian las etiquetas que se utilizan para definir los controles dentro de un formulario. Cada una de ellas tiene sus atributos específicos según el tipo de control de que se trate, aunque en todos los casos se utiliza el atributo **name** para especificar el nombre del control.

- **Etiqueta <input>.** Es la etiqueta más utilizada para definir controles en formularios. Esta etiqueta permite definir diferentes tipos de controles para los formularios: cuadros de texto, casillas de selección única y múltiple, botones, etc. El significado de los atributos de la etiqueta <input> es diferente según el tipo de control que define. Los principales atributos de esta etiqueta son los siguientes:
  - **type.** Es el atributo más importante y característico. Define el tipo de control de formulario. Los posibles valores de este atributo son: **text**, **password**, **checkbox**, **radio**, **submit**, **reset**, **file**, **hidden**, **image** y **button**.

- name. Especifica el identificador o nombre del control.
- size. Establece el ancho del control, en pixels o en caracteres.
- maxlength. Establece el número máximo de caracteres admitidos para el texto introducido en los controles de tipo text o password.
- checked. Activa o marca la casilla para un control de tipo checkbox o radio.
- src. Especifica la URI donde se encuentra la imagen para controles de tipo image.
- disabled. Deshabilita el control, de modo que se puede introducir información.
- readonly. Impide que el usuario puede hacer cambios en el control.
- accept. Indica una lista de tipos de archivos admitidos en un control file.
- value. Especifica un valor para el control. Es opcional, excepto para controles de tipo radio o checkbox ya que, en estos casos, especifica el valor que se enviará al seleccionar la casilla correspondiente del formulario. Para un campo de tipo text especifica el valor inicial que tendrá el control.

A continuación, se estudian los diferentes tipos de controles de formulario que pueden definirse utilizando para ello la etiqueta <input>:

- **Cuadro de texto.** Se crea con la etiqueta y atributo <input type="text">. Acepta líneas simples de texto. El tamaño del control puede especificarse mediante los atributos size y maxlength.
- **Cuadro de texto de tipo contraseña.** Se crea mediante el uso de la etiqueta y atributo <input type="password">. Oculta mediante asteriscos (\*) el texto que sea introducido por el usuario.
- **Casillas de verificación de selección múltiple.** Se crean con la etiqueta y atributo <input type="checkbox">. Al hacer clic sobre ella, la casilla de verificación se activa y queda seleccionada. Admite selección múltiple, por lo que pueden activarse varias casillas de verificación a la vez. Además, puede haber varias casillas de verificación con el mismo nombre de control, lo que permite que los usuarios indiquen varios valores para el mismo identificador o nombre. El atributo checked permite especificar que la casilla estará activada como valor predeterminado.
- **Casillas de verificación de selección única.** Se crean con la etiqueta y atributo <input type="radio">. Este tipo de control es similar al anterior, pero con la restricción de que solo una de las casillas definidas con el mismo nombre puede ser activada. Por tanto, aquellas casillas que tengan el mismo nombre de control serán mutuamente excluyentes.

- **Botón de cancelación.** Se crea con la etiqueta y atributo `<input type="reset">`. Al hacer clic sobre él, se borran los datos introducidos en los todos controles del formulario. El atributo value almacena el texto que aparecerá sobre el botón.
- **Botón de envío.** Se crea con la etiqueta y atributo `<input type="submit">`. Al hacer clic sobre él, se enviarán los datos introducidos en el formulario hacia el agente de software especificado en el valor del atributo action del formulario. El atributo value almacena el texto que aparecerá sobre el botón.
- **Botón de envío alternativo.** Se crea con mediante el uso de la etiqueta y atributo `<input type="image">`. Se trata de una alternativa personalizada botón de envío, ya que se utiliza la imagen especificada en el atributo src como botón de envío del formulario.
- **Selección de archivo.** Se crea con la etiqueta y atributo `<input type="file">`. Se utiliza para enviar archivos mediante un formulario. Solo puede utilizarse cuando el método de envío del formulario es POST.
- **Control oculto.** Se crea con la etiqueta y atributo `<input type="hidden">`. Se utiliza para enviar información transparente para el usuario. Cuando se utiliza este tipo de control, el usuario no interviene en el proceso de cumplimentación del control, ni lo puede ver y ni siquiera sabe que existe. Su finalidad es enviar datos al servidor Web que no son útiles para el cliente, por ejemplo, la fecha y hora del envío.
- **Botón de uso a definir.** Se crea con la etiqueta y atributo `<input type="button">`. La acción que se realizará cuando se haga clic sobre el botón se especifica mediante código script. Al hacer clic sobre un control de tipo button, el formulario no será enviado. El atributo value almacena el texto que aparecerá sobre el botón.

Ejemplo. A continuación, se muestra un ejemplo en el que se ha creado un formulario que incluye diversos tipos de controles especificados mediante la etiqueta `<input>`. Puede apreciarse el uso que se hace de los atributos de la etiqueta input para definir cada control del formulario.

```
<!DOCTYPE html>
<html>
<head>
  <title>Formulario 2</title>
  <meta charset="utf-8" />
</head>
<body>
  <h2>Datos del socio</h2>
  <form action="http://servidor5/manejador.php" method="post">
    <p>
      Usuario: <input type="text" name="usuario" value="user">
      <br />
      Contraseña: <input type="password" name="pass">
    </p>
    <p>
      Nombre: <input type="text" name="socio" size="40">
    </p>
  </form>
</body>
</html>
```

```
<br />
Correo electrónico:<input type="text" name="socio" size="20">
</p>
Sexo:
<input type="radio" name="genero" value="m" checked> Masculino
<input type="radio" name="genero" value="f"> Femenino
<br />
<p>
  Temas preferidos:
  <input type="checkbox" name="aventuras" value="a" checked> Acción
  <input type="checkbox" name="historica" value="h" checked> Histórica
  <input type="checkbox" name="ficción" value="f"> Ficción
  <br />
  Descuento a aplicar:
  <input name="descuento" value="15%" size="5" readonly>
</p>
Subir fotografía (en formato .zip)
<input type="file" name="fichero" accept="application/x-zip-compressed">
<p>
  <input type="submit" value="Enviar">
  <input type="reset" value="Cancelar">
  <input type="button" value="Cerrar" onclick="window.close();">
</p>
</form>
</body>
</html>
```

El ejemplo anterior, produce el siguiente resultado en el navegador:

## Datos del socio

Usuario:

Contraseña:

Nombre:

Correo electrónico:

Sexo: ☒ Masculino ☐ Femenino

Temas preferidos: ☒ Acción ☒ Histórica ☐ Ficción

Descuento a aplicar:

Subir fotografía (en formato .zip)

En el código del ejemplo anterior, el control denominado usuario tiene el valor "user" como valor por defecto. El primer conjunto de casillas de selección única, o de tipo radio, tienen asignados el mismo nombre ("genero"), por lo que solo podrá ser activada una de ellas. El segundo conjunto de casillas es de tipo checkbox, por lo que admite selección múltiple. Además, se han activado varias de ellas de forma predeterminada mediante el atributo checked y podrán ser desactivadas por el usuario. El control denominado "descuento" es de solo lectura, por lo que el usuario no podrá cambiar su valor. El control de tipo file solo admite archivos de tipo .zip. El control button define un botón genérico al cual se le ha asociado la acción de cerrar la ventana del navegador mediante código javascript.

- **Etiqueta <select>**. Permite definir cuadros combinados y listas de selección múltiple dentro de un formulario. Dentro de la etiqueta <select> se definen las opciones mediante la etiqueta <option>. Una etiqueta <select> debe contener al menos una opción definida mediante la etiqueta <option>. Los atributos más importantes de la etiqueta <select> son:
  - name. Especifica el identificador o nombre del control.
  - size. Indica el número de opciones visibles. El valor predeterminado es 1, por lo que solo se visualizará una de las opciones y el control se presentará como un cuadro combinado. Para mostrar el resto de las opciones debe hacer clic en el botón de la derecha del control para desplegar la ventana de opciones al completo.
  - multiple. Permite seleccionar varias opciones desde la ventana de opciones del control. Si no se incluye este atributo solo se permite seleccionar una opción.

Los principales atributos de la etiqueta <option> son los siguientes:

- value. Especifica el valor que envía el formulario cuando se selecciona esa opción.
- selected. Indica que se permite la preselección de uno o varios elementos del control de manera predeterminada.

Ejemplo. En el siguiente código de ejemplo se crea un cuadro combinado y una lista de selección múltiple mediante las etiquetas <select> y <option>.

```
<h1>Cuadro combinado. Selección única</h1>
<div> Departamento: </div>
<div>
  <select name="departamento">
    <option value="Dir">Dirección</option>
    <option value="Fab" selected>Fabricación</option>
    <option value="Adm">Administración</option>
    <option value="Rhu">Recursos humanos</option>
    <option value="Cal">Calidad</option>
    <option value="Log">Logística</option>
  </select>
</div>
<h1>Lista de selección múltiple</h1>
<div> Ciudades: </div>
<div>
```



```
<select name="Localiza" multiple size="4">
  <option value="Alcoy">componente_1</option>
  <option value="Alicante" selected>componente_2</option>
  <option value="Benidorm" selected>componente_3</option>
  <option value="Elche">componente_4</option>
  <option value="Orihuela">componente_5</option>
  <option value="Villena">componente_6</option>
</select>
</div>
```

El código de ejemplo anterior, produce el siguiente resultado en el navegador:

## Cuadro combinado. Selección única

Departamento:

Fabricación ▼

## Lista de selección múltiple

Ciudades:

componente\_1

componente\_2 ▲

componente\_3 ▼

componente\_4

- **Etiqueta <textarea>**. Permite crear un área de texto. Este control permite introducir una cierta extensión de texto en una ventana. Puede ser considerado como un cuadro de texto multilínea. El atributo name especifica el identificador o nombre del control. Los atributos rows y cols definen el número de filas y columnas visibles, respectivamente. Por ejemplo:

```
<div>Comentarios:</div>
<form action="http://servidor5/manejador1.php" method="post">
  <textarea name="comenta" rows="4" cols="50">Introduce un comentario...</textarea>
</form>
```

El código de ejemplo anterior produce el siguiente resultado en el navegador:

Comentarios:

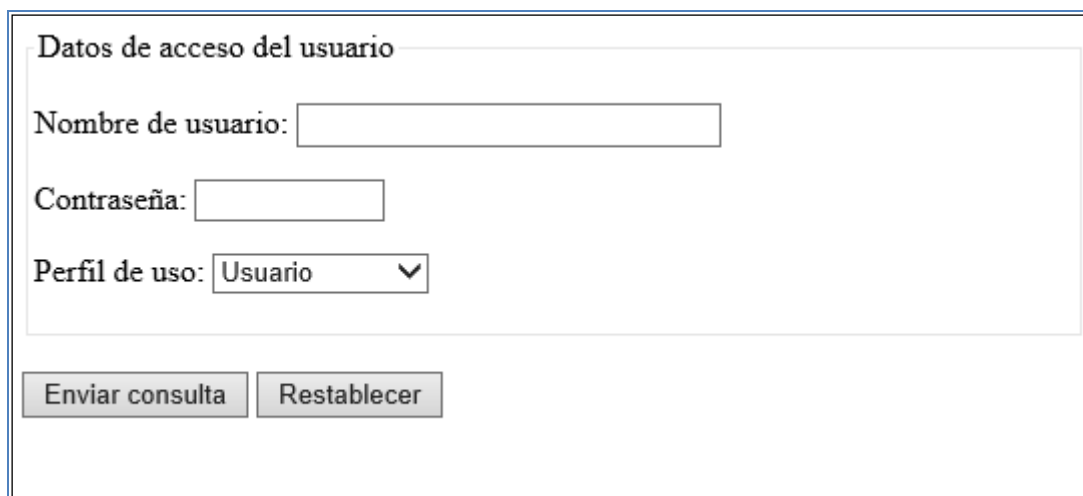
Introduce un comentario...



Los formularios admiten también las **etiquetas <fieldset> y <legend>**. A diferencia de las etiquetas anteriores que permiten especificar los controles que forman un formulario y ciertos aspectos sobre los valores que introducen los usuarios, las etiquetas <fieldset> y <legend> solo afectan a los aspectos de apariencia visual y facilidad de acceso a los controles del formulario. La etiqueta <fieldset> es un elemento visual que se utiliza para agrupar los controles. Y la etiqueta <legend> sirve para asignar un título a un elemento fieldset. Por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Formularios 5</title>
  <meta charset="utf-8" />
</head>
<body>
  <form action="http://servidor5/manejador1.php" method="post">
    <fieldset width="300">
      <legend>Datos de acceso del usuario</legend>
      <p>Nombre de usuario: <input type="text" name="usuario" size=30></p>
      <p>Contraseña: <input type="text" name="passwd" size=10></p>
      <p>
        Perfil de uso:
        <select id="perfil">
          <option value="U" selected="selected">Usuario</option>
          <option value="G">Gestor</option>
          <option value="A">Administrador</option>
        </select>
      </p>
    </fieldset>
    <br />
    <input type="submit">
    <input type="reset">
  </form>
</body>
</html>
```

El código de ejemplo anterior produce el siguiente resultado en el navegador:



## 4.5 OTROS ELEMENTOS Y ETIQUETAS

En este apartado se describen otros elementos y etiquetas del lenguaje HTML que permiten realizar diversas acciones y efectos.

### Icono en la barra de navegación del explorador

Para añadir un icono que se muestre en la barra de navegación del navegador, se utiliza la etiqueta de cabecera <link>, de la siguiente forma:

```
<link rel="shortcut icon" href="imagen.ico">
```

Al atributo href se le asigna el camino relativo del archivo que contiene la imagen del icono que se pretenda incorporar. Este archivo deberá tener la extensión .ico. Cuando se construye un sitio Web únicamente es necesario incorporar esta etiqueta en la página principal del sitio Web para que el icono se muestre en todas las demás.

### Redirección automática a otra página

Puede hacerse que una página Web redirija automáticamente hacia otra. Este efecto se consigue mediante la etiqueta de cabecera <meta>, de la siguiente forma:

```
<meta http-equiv="refresh" content="5;url=Formulario2.html">
```

Al atributo content se le asignan dos valores separados por punto y coma. El primero indica el número de segundos que tardará en producirse la redirección, si se pone el valor 0 la redirección se producirá en cuanto se cargue la página. Y el segundo, indica el nombre de la página Web destino de la redirección. Esta etiqueta configura el navegador para que realice la redirección deseada.

### Inserción de objetos multimedia

La etiqueta <object> permite insertar un objeto en el documento. El tipo de objeto viene determinado por sus atributos y puede ser una imagen, un applet de java, un vídeo, una animación, otro documento HTML, etc. Habitualmente, esta etiqueta se utiliza para insertar objetos multimedia, como pueden ser videos. Sus atributos más importantes son los siguientes:

- data. Especifica la URI donde están los datos del objeto. En algunos casos, puede utilizarse para especificar la localización del objeto a mostrar.
- type. Indica el tipo de contenido del objeto especificado en el atributo data. Es opcional, pero se recomienda su uso para facilitar al navegador el reconocimiento del tipo de archivo y la carga de la aplicación necesaria para la presentación del objeto. Ejemplos de valores de este atributo: "audio/basic", "audio/mpp3", "video/mpeg", "video/quicktime", etc.
- archive. Permite especificar una lista de URIs separada por espacios en blanco que incluyen fuentes relevantes y útiles relacionadas con el objeto.

- **standby.** Permite incluir un mensaje en el navegador durante la carga de los datos.
- **height, width.** Especifican el alto y ancho del objeto.
- **name.** Especifica el nombre para designar el elemento.

La etiqueta `<object>` suele ir acompañada por la etiqueta `<param>`. La etiqueta `<param>` sirve para inicializar los valores de los parámetros que utilizará el objeto en tiempo de ejecución. Cada etiqueta `<param>` define un par parámetro-valor que se pasará al objeto para su presentación. Los atributos más importantes de la etiqueta `<param>` son **name**, que define el nombre del parámetro, y **value**, que define el valor del parámetro.

Ejemplo. Insertar un video. El siguiente ejemplo muestra cómo insertar un vídeo local en una página web. Puede apreciarse que se utiliza el atributo **data** para especificar la dirección relativa del archivo, así como la etiqueta `<param>` para definir diversos parámetros relativos a la presentación.

```
<object type="video/wmv" id="susto" width="300" height="200" data="./susto.wmv">  
  <param name="quality" value="high">  
</object>
```

En la actualidad, existe un creciente interés por parte de los desarrolladores y usuarios en almacenar los vídeos en plataformas externas, tales como YouTube. La etiqueta `<iframe>` también se puede utilizar para insertar vídeos, especialmente de YouTube. Para insertar un vídeo de YouTube basta con indicar la dirección del vídeo que proporciona YouTube en la opción **Compartir**. Por ejemplo:

```
<p>La nadadora Mireia Belmonte consiguió la primera medalla de oro para la delegación española en los Juegos Olímpicos de Río de Janeiro, tras imponerse este miércoles, en la quinta jornada de los juegos, en la final de los 200 mariposa.</p>  
<iframe width="560" height="315" src="https://www.youtube.com/embed/jaN CHj6jTo" frameborder="0" allowfullscreen></iframe>
```

Las etiquetas de inserción de objetos que se han estudiado en este apartado están en desuso. Ello es debido a la aparición de nuevas etiquetas para la presentación de objetos multimedia que han aparecido en el nuevo estándar HTML5. En un tema posterior se abordará con mayor extensión la inclusión de objetos multimedia mediante las nuevas etiquetas de HTML5 que facilitan las tareas al desarrollador Web.

## Glosario de términos

**Applet.** Es una microaplicación compilada de Java. Se descargan como recursos de la página Web, tal como ocurre con las imágenes u otros elementos que se presentan en la página. Una vez descargadas son ejecutadas por el navegador. Actualmente, están en desuso por razones de seguridad. Para que un applet pueda ejecutarse, el navegador debe estar configurado para de forma que se permita su ejecución.

**Script.** Un script es bloque de código procedural que se incluye en los documentos HTML para realizar acciones que requieren un procesamiento lógico. Estos bloques de código lógico se escriben utilizando la etiqueta <script> y, habitualmente, son interpretados por el navegador. Se utiliza un lenguaje de script para definir su contenido que, suele ser Javascript.

**URL (*Uniform Resource Locator*).** Es un identificador de recursos uniforme (*Uniform Resource Identifier, URI*) cuyos recursos a los que se refiere pueden cambiar, es decir, la dirección puede apuntar a recursos variables en el tiempo. Están formados por una secuencia de caracteres, de acuerdo con un formato estándar que designa recursos en una red, como puede ser Internet. Las URL fueron usadas por primera vez por Tim Berners-Lee en 1991, para permitir a los autores de documentos establecer hipervínculos o enlaces en la *World Wide Web* (WWW). Desde 1994, en los estándares de Internet, el concepto de URL ha sido incorporado dentro del más general de URI, pero el término URL todavía se utiliza ampliamente.

**URI (*Uniform Resource Identifier, URI*).** Es un identificador de recursos formado por una cadena de caracteres que identifica los recursos existentes en una red de una forma unívoca. La diferencia respecto a un localizador de recursos uniforme (URL) es que estos últimos hacen referencia a recursos que, de forma general, pueden variar en el tiempo. La sintaxis del formato URI es:

protocolo://autoridad/ruta/recurso?Consulta

El formato URI consta de las siguientes partes:

- protocolo. Es el protocolo de acceso al recurso, por ejemplo: http:, mailto:, ftp:, etc
- autoridad. Es el elemento jerárquico que identifica la autoridad de nombres, por ejemplo //www.example.com.
- ruta. Información usualmente organizada en forma jerárquica, que identifica la situación del recurso en el ámbito del esquema URI y la autoridad de nombres.
- Recurso. identifica el recurso en el ámbito del esquema URI y la autoridad de nombres. Es el nombre del recurso a utilizar.
- consulta. Es un conjunto de Información con una estructura no jerárquica, usualmente estructurada en pares "clave=valor" que se proporciona al recurso. El comienzo de este componente se indica mediante el carácter '?'.

Aunque es habitual utilizar la expresión dirección URL para referirse a todas las direcciones web, sin embargo, utilizar la expresión dirección URI es más general y correcto, por lo que se recomienda utilizar esta expresión en lugar de la expresión dirección URL.