

## Tema 8.

# Formularios

*El estándar HTML/XHTML permite crear formularios para que los usuarios interactúen con las aplicaciones web.*

*El formulario solo sirve para recabar información del usuario, pero el tratamiento posterior de esta información lo hace normalmente un programa externo al documento.*

*Estos programas realizan diversas tareas como:*

- *Verificar y manipular los datos.*
- *Almacenar los datos en un archivo o en una base de datos.*
- *Reenviar los datos por correo electrónico.*
- *Leer los datos y devolver algún resultado al usuario.*

*Para construir estos programas se utiliza un lenguaje de programación como C, Java, PHP.*

*La etiqueta `<form>` encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables).*

*La mayoría de formularios utilizan sólo los atributos `action` y `method`. El atributo `action` indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador.*

*Ejemplo: `<form action="registroCliente.php">`*

*También puede enviar los datos a una dirección de correo.*

*Ejemplo: `<form action="mailto:direccion@correo.com">`*

*El atributo `method` establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son `GET` y `POST`.*

*Al margen de otras diferencias técnicas, el método `POST` permite el envío de mucha más información que el método `GET`. `GET` no permite el envío de archivos adjuntos con el formulario. Además, los datos enviados mediante `GET` se ven en la barra de direcciones del navegador (se añaden al final de la URL de la página), mientras que los datos enviados mediante `POST` no se pueden ver tan fácilmente.*

Si no sabes que método elegir para un formulario, existe una regla general que dice que el método **GET** se debe utilizar en los formularios que no modifican la información (por ejemplo en un formulario de búsqueda). Por su parte, el método **POST** se debería utilizar cuando el formulario modifica la información original (insertar, modificar o borrar alguna información).

Nosotros siempre utilizaremos **POST**

Del resto de atributos de la etiqueta **<form>**, el único que se utiliza ocasionalmente es **enctype**. Se utiliza para indicar la forma en la que viajará la información que se mande por el formulario. En el caso más corriente, enviar el formulario por correo electrónico, el valor de este atributo debe de ser "text/plain".

Así conseguimos que se envíe el contenido del formulario como texto plano dentro del email.

Si no ponemos esta etiqueta la información del formulario llegaría así:

CAMPO1=Esto+es+el+primer+campo&CAMPO2=esto+es+el+segundo+campo

Si la ponemos la información llegaría así:

CAMPO1=Esto es el primer campo

CAMPO2=Esto es el segundo campo

Ejemplo de etiqueta **<form>** completa

**<form action="mailto:direccion@correo.com" method="post" enctype="text/plain">**

## 8.1. Elementos de formulario

También se denominan "campos de formulario" y "controles de formulario". La mayoría de controles se crean con la etiqueta **<input>**, la etiqueta **<input>** permite definir varios tipos diferentes de elementos (botones y cuadros de texto) por lo que su definición formal y su lista de atributos es muy extensa:

A continuación se indica los diferentes atributos que puede contener la etiqueta **<input>**

**type = "text | password | checkbox | radio | submit | reset | button"**

Indica el tipo de control que se incluye en el formulario

**name = "texto"** - Asigna un nombre al control (es imprescindible para que el servidor pueda procesar el formulario)

**value = "texto"** - Valor inicial del control. Para botones de acción es el título del botón

**size = "unidad\_de\_medida"** - Longitud de los controles de texto. Por defecto suelen tener 20 caracteres

**maxlength = "numero"** - Máximo número de caracteres para los controles de texto y de password

*minlength = "numero" - Mínimo número de caracteres para los controles de texto y de password*

*placeholder= "texto" permite mostrar en los controles de texto un texto que desaparece al escribir en el control.*

*checked = "checked" - Para los controles checkbox y radiobutton permite indicar qué opción aparece seleccionada por defecto*

*A continuación se muestran ejemplos para los diez controles que se pueden crear con la etiqueta `<input>`.*

### **8.1.1. Cuadro de texto**

*Se trata del elemento más utilizado en los formularios. En el caso más sencillo, se muestra un cuadro de texto vacío en el que el usuario puede escribir cualquier texto:*

Nombre

*A continuación se muestra el código HTML correspondiente al ejemplo anterior:*

—Nombre `<br/>`  
`<input type="text" name="nombre" value="" />`

*El atributo `type` diferencia a cada uno de los diez controles que se pueden crear con la etiqueta `<input>`. Para los cuadros de texto, su valor es `text`. El atributo `name` es el más importante en los campos del formulario. De hecho, si un campo no incluye el atributo `name`, sus datos no se envían al servidor. El valor que se indica en el atributo `name` es el nombre que utiliza la aplicación del servidor para obtener el valor de este campo de formulario.*

*El atributo `value` se emplea para establecer el valor inicial del cuadro de texto.*

*Si no se especifica un tamaño, el navegador muestra el cuadro de texto con un tamaño predeterminado. El atributo `size` permite establecer el tamaño, en caracteres, con el que se muestra el cuadro de texto.*

*Además de controlar el tamaño con el que se muestra un cuadro de texto, también se puede limitar el tamaño del texto introducido. El atributo `maxlength` permite establecer el máximo número de caracteres que el usuario puede introducir en un cuadro de texto.*

*Ejemplo campo texto con todos los atributos:*

`<input type="text" name="nombre" value="" size="10" maxlength="5"/>`

### 8.1.2. Cuadro de contraseña

La única diferencia entre este control y el cuadro de texto normal es que el texto que el usuario escri-b e en un cuadro de contraseña no se ve en la pantalla. En su lugar, los navegadores ocultan el texto utilizando asteriscos o círculos,.

Contraseña

Contraseña <br/>

<input type="password" name="contrasena" value="" />

Cambiando el valor del atributo `type` por `password` se transforma el cuadro de texto normal en un cuadro de contraseña. Todos los demás atributos se utilizan de la misma forma y tienen el mismo significado.

### 8.1.3. Checkbox

Los checkbox o "casillas de verificación" son controles de formulario que permiten al usuario seleccionar y deseleccionar opciones individualmente. Se utilizan cuando el usuario puede activar y desactivar varias opciones relacionadas pero no excluyentes.

Puestos de trabajo buscados

☐ Dirección

☐ Técnico

☐ Empleado

Puestos de trabajo buscados <br/>

<input name="puesto\_directivo" type="checkbox" value="direccion"/>

Dirección <input name="puesto\_tecnico" type="checkbox"

value="tecnico"/> Técnico

<input name="puesto\_empleado" type="checkbox" value="empleado"/>

Empleado

El valor del atributo `type` para estos controles de formulario es `checkbox`. Si no se añade un texto al lado de la etiqueta `<input />` del checkbox, el usuario sólo ve un pequeño cuadrado sin ninguna información relativa a la finalidad de ese checkbox.

Si se quiere mostrar un checkbox seleccionado por defecto, se utiliza el atributo `checked`.

<input type="checkbox" checked="checked" ... /> Checkbox seleccionado por defecto

### 8.1.4. Radiobutton

Son mutuamente excluyentes. Los **radiobutton** se utilizan cuando el usuario solamente puede escoger una opción entre las distintas opciones relacionadas que se le presentan. Cada vez que se selecciona una opción, automáticamente se deselecciona la otra opción que estaba seleccionaba.

Sexo

☒ Hombre

☐ Mujer

Sexo <br/>

```
<input type="radio" name="sexo" value="hombre" checked="checked" /> Hombre  
<input type="radio" name="sexo" value="mujer" /> Mujer
```

El valor del atributo **type** para estos controles de formulario es **radio**. El atributo **name** se emplea para indicar los radiobutton que están relacionados. Por lo tanto, cuando varios radiobutton tienen el mismo valor en su atributo **name**, el navegador sabe que están relacionados y puede deseleccionar una opción del grupo de radiobutton cuando se seleccione otra opción.

### 8.1.5. Botón de envío de formulario

```
<input type="submit" name="enviar" value="Enviar" />
```

El valor del atributo **type** para este control de formulario es **submit**. El navegador se encarga de enviar automáticamente los datos cuando el usuario pincha sobre este tipo de botón. El valor del atributo **value** es el texto que muestra el botón. Si no se establece el atributo **value**, el navegador muestra el texto predefinido **Enviar consulta**.

### 8.1.6. Botón de reseteo del formulario

Se trata de un botón especial que borra todos los datos introducidos por el usuario y devuelve el formulario a su estado original:

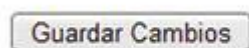
Borrar datos del formulario

```
<input type="reset" name="limpiar" value="Borrar datos del formulario" />
```

El valor del atributo **type** para este control de formulario es **reset**. Cuando el usuario pulsa este botón, el navegador borra toda la información introducida y muestra el formulario en su estado original.

### 8.1.7. Botón

HTML define un botón de tipo genérico:



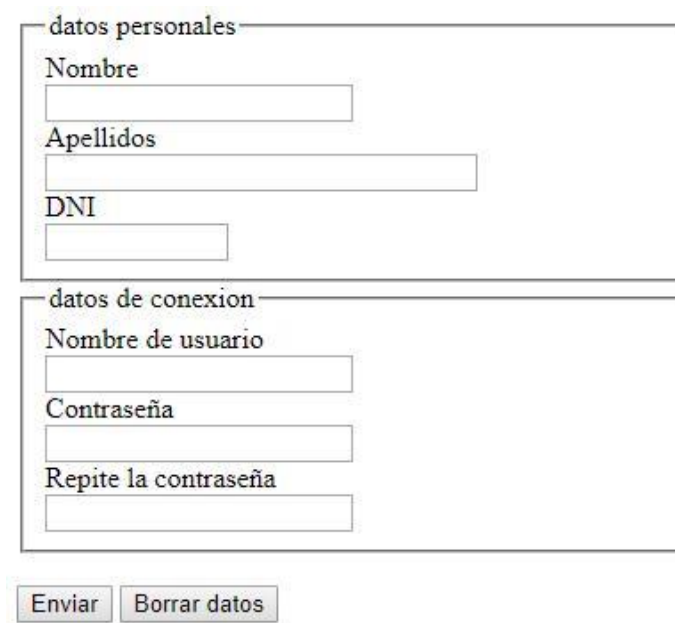
— `<input type="button" name="guardar" value="Guardar Cambios" />`

Si pruebas a pulsar un botón de este tipo, verás que el navegador no hace nada: no envía los datos al servidor y no borra los datos introducidos. Este tipo de botones sólo son útiles si se utilizan junto con el lenguaje de programación JavaScript.

## Realizar el ejercicio 1 formularios subido a aules

### 8.2. Formularios avanzados

La siguiente imagen muestra un formulario que agrupa sus elementos y añade etiquetas a cada campo para mejorar su estructura:

The image shows a web form with two distinct sections, each enclosed in a box with a legend at the top. The first section, titled 'datos personales', contains three text input fields labeled 'Nombre', 'Apellidos', and 'DNI'. The second section, titled 'datos de conexion', contains three text input fields labeled 'Nombre de usuario', 'Contraseña', and 'Repite la contraseña'. Below these two sections are two buttons: 'Enviar' and 'Borrar datos'.

La etiqueta `<fieldset>` agrupa campos del formulario y la etiqueta `<legend>` asigna un nombre a cada grupo.

`<fieldset>`

`<legend>datos personales</legend>`

....  
....

`</fieldset>`

La etiqueta `<fieldset>` agrupa todos los controles de formulario a los que encierra. El navegador muestra por defecto un borde resaltado para cada agrupación. La etiqueta `<legend>` se incluye dentro de cada etiqueta `<fieldset>` y establece el título que muestra el navegador para cada agrupación de elementos.

## Realizar el ejercicio 2 formularios subido a aules

### 8.2.1 Etiqueta `<label>`

El lenguaje HTML incluye una etiqueta denominada `<label>` y que se utiliza para establecer el título de cada campo del formulario. Su definición formal es la siguiente:

```
<label for="nombre">Nombre</label> <br/>
<input type="text" id="nombre" name="nombre" value="" />
```

La principal ventaja de utilizar `<label>` es que el código HTML está mejor estructurado y se mejora su accesibilidad. Además, en CSS puedo poner un formato a las etiquetas e incluso uno distinto a cada etiqueta refiriéndome al nombre del atributo `id`.

## 8.3. Otros elementos de formulario

### `<textarea>`

Las áreas de texto son útiles cuando se debe introducir una gran cantidad de texto, ya que es mucho más cómodo de introducir que en un campo de texto normal:

La definición formal de la etiqueta `<textarea>` es:

```
<textarea id="descripcion" name="descripcion" cols="40" rows="5"></textarea>
```

Los atributos más utilizados en las etiquetas `<textarea>` son los que controlan su anchura y altura. La anchura del área de texto se controla mediante el atributo `cols`, que indica las columnas o número de caracteres que se podrán escribir como máximo en cada fila. La altura del área de texto se controla mediante `rows`, que indica directamente las filas de texto que serán visibles.

El principal inconveniente de los elementos `<textarea>` es que el lenguaje HTML no permite limitar el número máximo de caracteres que se pueden introducir.



## `<select>`

Se utiliza para incluir listas desplegables:

Los tres tipos de listas desplegables se definen con la misma etiqueta `<select>` y cada elemento de la lista se define mediante la etiqueta `<option>`:

La inmensa mayoría de listas desplegables que utilizan las aplicaciones web son simples, por lo que el código HTML habitual de las listas desplegables es:

```
<label for="so">Sistema operativo</label>
<br/> <select id="so" name="so">
  <option value="" selected="selected">- selecciona -
</option> <option value="windows">Windows</option>
<option
value="mac">Mac</option>
<option
value="linux">Linux</option>
```



```
<option  
value="otro">Otro</option>  
</select>
```

La etiqueta `<select>` define la lista y encierra todas las opciones que muestra la lista. Cada una de las opciones de la lista se define mediante una etiqueta `<option>`. El atributo `value` de cada opción es obligatorio. Para seleccionar por defecto una opción al mostrar la lista, se añade el atributo `selected` a la opción deseada.



La imagen anterior muestra los tres tipos de listas desplegables disponibles. El primero es el de las listas más utilizadas que sólo muestran un valor cada vez y sólo permiten seleccionar un valor. El segundo tipo de lista es el que sólo permite seleccionar un valor pero muestra varios a la vez. Por último, el tercer tipo de lista desplegable es aquella que muestra varios valores y permite realizar selecciones múltiples con ayuda de la tecla Control o Mayúsculas). En este caso el atributo `name` debe incluir corchetes (`[]`) para que se puedan enviar en su caso varios valores en forma de matriz.

El código HTML del ejemplo anterior se muestra a continuación:

```
<label for="so">Sistema operativo</label>
<br/> <select id="so" name="so">
  <option value="" selected="selected">- selecciona -
</option> <option value="windows">Windows</option>
  <option value="mac">Mac</option> <option value="linux">Linux</option>
  <option value="otro">Otro</option>
</select>
```

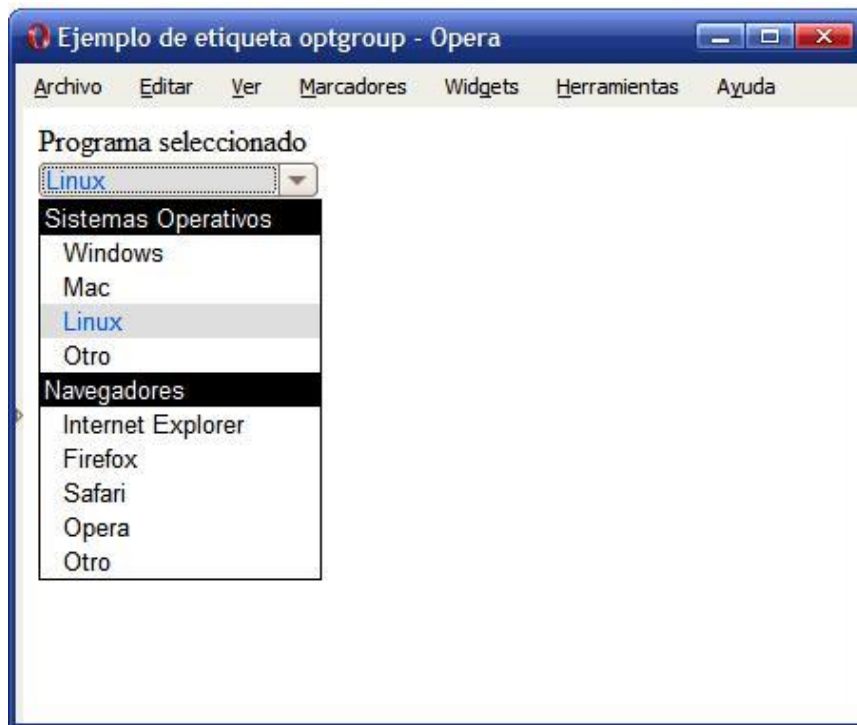
```
<label for="so2">Sistema operativo</label>
<br/> <select id="so2" name="so2" size="5">
  <option value="windows"
selected="selected">Windows</option> <option
value="mac">Mac</option>
  <option
value="linux">Linux</option>
  <option
value="otro">Otro</option>
</select>
```

```
<label for="so3">Sistema operativo</label> <br/>
<select id="so3" name="so3" size="5" multiple="multiple">
  <option value="windows"
selected="selected">Windows</option> <option
value="mac">Mac</option>
  <option
value="linux">Linux</option>
  <option
value="otro">Otro</option>
</select>
```

En la lista 2 se añade el atributo `size` que indica el número de opciones que se visualizan.

En la lista 3 también utiliza el atributo `size` y añade el atributo `múltiple` para poder elegir más de una opción.

Por otra parte, las listas desplegables permiten agrupar sus opciones de forma que el usuario pueda encontrar fácilmente las opciones cuando la lista es muy larga:



El código HTML correspondiente a la imagen anterior se muestra a continuación:

```
<form id="formulario" method="post" action="">
<label for="programa">Programa
seleccionado</label> <br/> <select id="programa"
name="programa">
  <optgroup label="Sistemas Operativos">
    <option value="Windows"
selected="selected">Windows</option> <option
value="Mac">Mac</option>
    <option
value="Linux">Linux</option>
    <option
value="Other">Otro</option>
  </optgroup>
  <optgroup label="Navegadores">
    <option value="Internet Explorer" selected="selected">Internet
Explorer</option> <option value="Firefox">Firefox</option>
```

```
<option  
value="Safari">Safari</option>  
<option  
value="Opera">Opera</option>  
<option  
value="Other">Otro</option>  
</optgroup>  
<p> </select>  
  
</form>
```

La etiqueta `<optgroup>` permite agrupar opciones relacionadas dentro de una lista desplegable.

El único atributo que suele utilizarse con la etiqueta `<optgroup>` es `label`, que indica el nombre de cada agrupación. Los navegadores muestran de forma destacada el título de cada agrupación.

**Realizar el ejercicio 3, 4, 5 y 6 formularios subidos a aules**