

Formularios

Los **formularios HTML** son elementos que permiten recoger datos del cliente desde una página web (en el lado del cliente) y enviarlos al servidor.

Dentro del formulario, podemos introducir componentes HTML de distintas naturalezas para este fin. Por ejemplo, **campos** y **cajas de texto**, **selectores**, **checkbox**, **radio**, etc.

Desde un formulario podemos, incluso, mandar **ficheros** al servidor. Usando formularios, tenemos un mecanismo para que el usuario pueda interactuar con nuestra aplicación. Un formulario también puede ser utilizado para mostrar datos al cliente.

Un formulario permite enviar al **servidor** la información recogida por los diferentes elementos que lo componen. Los datos se envían como una **lista de parámetros**. Cada parámetro se corresponde con un elemento del formulario.

La lista de parámetros se envía en formato **clave / valor**, donde la clave es el nombre del elemento del formulario al que corresponde y el valor es la información introducida o seleccionada por el usuario.

La información de un formulario se envía en una petición HTTP, por lo que debemos indicar en el formulario cual va a ser la URL del servidor que va a atender nuestra petición.

Una petición HTTP admite varios métodos o verbos como vimos en el modulo uno. Podemos indicar en el formulario si la información se va a enviar usando el método **GET** o **POST**.

Si no se especifica el método, por defecto se envía por **GET**.

Métodos HTTP

HTTP es un protocolo para la obtención de recursos, como los documentos HTML. Es la base de cualquier intercambio de datos en la Web y es un

protocolo cliente-servidor, lo que significa que las solicitudes son iniciadas por el destinatario, normalmente el navegador web.

Un documento completo se reconstruye a partir de los diferentes subdocumentos obtenidos, por ejemplo, el texto, la descripción del diseño, las imágenes, los vídeos, los scripts, etc.

Los **clientes** y los **servidores** se comunican mediante el intercambio de mensajes individuales (en lugar de un flujo de datos). Los mensajes enviados por el cliente, normalmente un navegador web, se denominan peticiones y los mensajes enviados por el servidor como respuesta se denominan respuestas.

El protocolo HTTP consta de diferentes métodos, siendo los más utilizados **GET** y **POST** para solicitar y enviar información al servidor.

El método GET

Método **GET**: la información recogida del formulario se envía en la propia URL de la petición con el siguiente formato:

```
url?param_1=valor_1&param_2=valor_2&.....param_n=valor_n
```

Al enviarse en la propia URL, los parámetros serían visibles para el usuario.

El método POST

Método **POST**: los datos del formulario se envían dentro del cuerpo del mensaje HTTP.

```
valor_1&param_2=valor_2&.....param_n=valor_n
```

Aunque los parámetros también son accesibles por el usuario, no se muestran directamente en el navegador. Es necesario utilizar alguna herramienta o plugin del navegador para poder acceder a ellos.

Creación de formularios con HTML

La etiqueta **form** es el elemento principal en la creación de un formulario. Se encarga de agrupar los elementos que recogerán los valores que serán enviados en la petición **HTTP**.

Estos son algunos de sus atributos:

- **action** (obligatorio): URL que va a ser llamada cuando se envíen los datos del formulario.
- **method**: método HTTP utilizado en la llamada al servidor. Están permitidos los valores get y post.
- **target**: define donde se va a mostrar la respuesta del servidor. Puede tener los mismos valores que el atributo target de la etiqueta a: *self*, *blank*, *parent*.

Para añadir un formulario a una página web, necesitamos conocer las etiquetas con las que definimos los campos de entrada de datos del formulario.

Los elementos de entrada de datos de un formulario pueden poseer los siguientes atributos comunes:

- **name**: este atributo identifica el elemento dentro del formulario. Pueden existir elementos con el mismo atributo name puesto que es relativo al formulario en el que están incluidos.
- **disabled**: el elemento aparece como desactivado. No es modificable por el usuario y visualmente, se presenta de diferente forma para que el usuario identifique claramente que el elemento está inactivo.

Puede ser útil en casos en los que, dependiendo del perfil del usuario, queramos que no pueda introducir un valor, y queramos hacérselo saber expresamente. Un ejemplo puede ser un formulario en el que deben darse de alta los datos de una empresa. Si se sabe que el usuario es autónomo, no podemos permitirle seleccionar un tipo de empresa.

Elementos de un formulario

input

A continuación veremos que etiquetas podemos incluir dentro de un formulario (etiqueta `form`).

`input`: elemento genérico de entrada de datos. Algunos atributos interesantes son:

- `type`: define el tipo de entrada de datos que queremos utilizar. Dependiendo del valor de este atributo, el componente interactuara de una manera concreta con el usuario y tendrá un aspecto u otro.
- `value`: valor por defecto del elemento.
- `readonly`: si aparece este atributo, el elemento es de solo lectura y no sería modificable por el usuario.

Tipos de `input`:

- `reset`: carga los valores por defecto de todos los campos del formulario. Se presenta al usuario como un botón. El valor de su atributo `value` define el texto que contiene el botón.
- `submit`: realiza la petición HTTP usando la URL definida en el atributo `action` del formulario, y añadiendo los parámetros correspondientes a los elementos contenidos en el formulario. Se presenta al usuario como un botón. El valor de su atributo `value` define el texto que contiene el botón.
- `text`: entrada de texto estándar. Solo admite una sola línea.
- `password`: es igual que un `input` de tipo `text` solo que cuando se escribe texto en el campo, se oculta con un asterisco o cualquier otro símbolo de ocultación.
- `hidden`: campo invisible para el usuario. Se suele utilizar para enviar datos al servidor cuyo valor no debe ser conocido ni editado por el usuario.
- `radio`: componente que permite seleccionar una opción entre las presentadas al usuario. Tiene la misma función que el `select` simple (solo una opción) solo que siempre se muestran todas las opciones en el formulario. Atributos.
 - `checked`: si se presenta este atributo en alguno de los `input`, este será el que aparezca seleccionado por defecto.
 - `checkbox`: elemento para marcar y desmarcar. Si se desmarca, no se envía el atributo al cliente. Si se marca, se envía el parámetro, y como valor se añade el atributo `value`.

textarea

textarea: caja de texto que admite múltiples líneas. Se utiliza para entradas de texto de gran tamaño y admite saltos de línea. Si se añade contenido a la etiqueta, este aparecerá como valor por defecto de la misma.

Si no se indica lo contrario mediante estilos, el área de texto permitirá ser redimensionada por el usuario.

Atributos del elemento **textarea**:

- **rows**: número de filas.
- **cols**: número de columnas.
- **readonly**: al igual que en el elemento input, si aparece este atributo, el elemento es de solo lectura y no sería modificable por el usuario.

select

select / **option**: elemento que permite elegir entre una lista de opciones. El elemento principal es la etiqueta select que contendrá la lista de posibles opciones.

Atributos de la etiqueta **select**:

- **múltiple**: si aparece este atributo, el componente permite seleccionar más de un elemento de la lista. Manteniendo pulsada la tecla **Ctrl** y haciendo click en cada elemento de los que queremos seleccionar.

Como contenido de la etiqueta **select**, podemos tener una o más etiquetas **option**. Por cada etiqueta option que añadimos a select, se añade un elemento a la lista desplegable que se muestra al usuario.

El contenido de la etiqueta **option** es el texto que se mostrara al usuario.

Atributos de la etiqueta **option**:

- **value**: valor que se enviara al servidor si se selecciona la opción.
- **disabled**: no solo podemos desactivar el componente completo añadiendo este atributo a la etiqueta select, sino que también podemos desactivar

algunas de las opciones concretas.

- **selected**: si se presenta este atributo en una opción, aparecerá como seleccionada. Si no, se selecciona por defecto la primera opción.

Ejemplo

Ejemplo de formulario completo:

```
<form action="/users" method="POST">

  <label for="first-name">First name:</label><br>

  <input type="text" id="first-name" name="first-name"
placeholder="Alan"><br>

  <label for="last-name">Last name *</label><br>

  <input type="text" id="last-name" name="last-name"
placeholder="Sastre"><br>

  <label for="email">Email:</label><br>

  <input type="email" id="email" name="email"
placeholder="alan@alansastre.co"><br>

  <label for="password">Password:</label><br>

  <input type="password" id="password" name="password"><br>

  <label for="web">Web:</label><br>

  <input type="url" id="web" name="web"
placeholder="www.alansastre.co"><br>
```

```
<label for="price">Price: </label><br>
```

```
<input id="price" type="number" step="0.01" value="0.00" ><br>
```

```
<label for="mensaje">Mensaje:</label><br>
```

```
<textarea id="mensaje" name="mensaje" placeholder="Escribe tu  
mensaje"
```

```
rows="6" cols="40"></textarea>
```

```
</form>
```