

Son muchos los sitios webs que solicitan datos al usuario como, por ejemplo, sus credenciales, los datos de un nuevo producto o la confirmación de un pedido. Hasta el momento, hemos visto cómo proporcionar texto a presentar al usuario. Ha llegado el momento de examinar cómo solicitar datos al usuario.

Comenzamos con una breve introducción del concepto de formulario. A continuación, examinamos los elementos de entrada de datos, los cuales se representan principalmente mediante elementos `<input>`, `<select>` o `<textarea>`. Después, describimos cómo puede solicitar el usuario el envío de los datos al servidor web para su procesamiento. Finalmente, introducimos el elemento `<label>` con el que asociar títulos para los elementos de entrada.

Al finalizar la lección, el estudiante sabrá:

- Qué es un formulario.
- Cómo solicitar datos al usuario.
- Cómo enviar los datos introducidos por el usuario al servidor web.

## Introducción

Un **formulario** (*form*) es un medio a través del cual solicitar datos al usuario. Son uno de los principales puntos de interacción entre el usuario y la aplicación. Por ejemplo, a través de un formulario se puede proporcionar las credenciales de una cuenta de usuario, los datos de un pedido, de un empleado, etc.

Generalmente, los datos recopilados mediante un formulario son procesados en el servidor, por ejemplo, por una aplicación **Node**.

En un documento web, se representa un formulario mediante un elemento `<form>`, el cual delimita los elementos a través de los cuales el usuario proporciona datos, conocidos formalmente como **campos** (*fields*).

## Elementos de entrada de datos

Un **elemento de entrada de datos** (*data input element*) es aquel a través del cual se solicita un determinado dato al usuario como, por ejemplo, su nombre o su teléfono, etc. Se representan mediante elementos `<input>`. Este elemento puede adaptarse fácilmente al tipo de datos que espera. Así, por ejemplo, podemos solicitar una entrada para obtener texto, otro para números, otro para una dirección de correo electrónico, etc. Esto se indica mediante el valor del atributo **type** del elemento.

El uso del tipo más adecuado es de vital importancia para facilitar la interacción del usuario con los elementos de entrada de datos. Si por ejemplo el usuario debe introducir un número, se recomienda utilizar el tipo **number** para que, así, si el usuario está usando el móvil, le salga un teclado de sólo dígitos. Por otra parte, cuando se utiliza un tipo especializado, el navegador puede validar más fácilmente el dato introducido. Por ejemplo, si indicamos el tipo **email**, el navegador comprobará que el valor introducido por el usuario cumple la sintaxis de las direcciones de correo.

## Entrada de texto

El tipo de datos más frecuentemente introducido por los usuarios es el texto. Este tipo de elemento se representa mediante un cuadro de texto parecido al siguiente:

Usuario

El valor del atributo **type** con el que indicar este tipo de entrada de datos es **text**. Es el valor predeterminado, en caso de omisión del atributo **type**. Ejemplo:

Usuario

### Entrada de texto multilinea

El tipo **text** sólo permite que el usuario introduzca una línea de datos. Si el usuario necesita introducir varias líneas, se puede utilizar el elemento **<textarea>**. Ejemplo ilustrativo:

Descripción de la incidencia  
<textarea name="desc" cols="40" rows="5"></textarea>

Descripción de la incidencia

### Entrada de contraseña

Si lo que deseamos es introducir una contraseña o PIN, la cual no debe mostrarse al usuario, podemos usar el tipo **password**. Ejemplo:

Contraseña

Cada vez que el usuario introduzca un carácter, el navegador mostrará un punto. He aquí un ejemplo:

Contraseña .....

### Entrada de números

Si el elemento de entrada de datos tiene como objeto solicitar un número, se puede hacer mediante el tipo **text** o mejor aún mediante **number**. De esta manera, el navegador sólo permitirá escribir números en el elemento. He aquí un ejemplo ilustrativo:

Año de nacimiento

Año de nacimiento 1975

Se puede definir el intervalo de valores que puede introducir el usuario mediante los atributos **min** y **max**. Con **min**, indicamos el más pequeño; y con **max**, el mayor.

### Entrada de fechas

Cuando deseamos que el usuario introduzca una determinada fecha, se recomienda utilizar el tipo **date**, en vez de **text**. Ejemplo:

Fecha de entrega

Fecha de entrega 09/30/2016

Generalmente, los navegadores muestran el elemento como un cuadro de texto. Pero cuando el usuario entra en él, le asiste mostrando un calendario con el que seleccionar la fecha.

Para restringir la fecha que puede introducir el usuario, se puede utilizar los atributos **min** y **max**. Con **min**, indicamos la fecha más pequeña; y con **max**, la mayor.

### Entrada de número de semana

Es posible indicar un número de semana. Con este fin se utiliza el tipo **week**. Ejemplo ilustrativo:

Semana de entrega

Semana de entrega Week 41, 2016 x ▾

October 2016 ▾

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
39	25	26	27	28	29	30	1
40	2	3	4	5	6	7	8
41	9	10	11	12	13	14	15
42	16	17	18	19	20	21	22
43	23	24	25	26	27	28	29
44	30	31	1	2	3	4	5

Para facilitar la entrada de una determinada semana, el navegador suele mostrar un calendario con el que hacer la selección.

### Entrada de horas

También hay una especialización del elemento `<input>` mediante el cual el usuario puede introducir una determinada hora. En este caso, se utiliza `time` como tipo. A continuación, se muestra un ejemplo:

Entregar entre `<input type="time" name="startTime">` y `<input type="time" name="endTime">`

Entregar entre 08:00 AM y 08:00 PM

### Entrada de URL

Si se necesita un URL, se recomienda usar el tipo `url`. Ejemplo:

Sitio web `<input type="url">`

Sitio web

### Entrada de correo electrónico

Como la introducción de correo electrónico es muy habitual, el elemento `<input>` también permite hacerlo de manera especial usando `email` como tipo. Ejemplo:

Correo-e `<input type="email" name="email">`

Correo-e

### Entrada de número de teléfono

Si el objetivo es solicitar un número de teléfono, se recomienda usar el tipo `tel`. Ejemplo:

Teléfono `<input type="tel" username="phone">`

Teléfono

Como cada país tiene su formato de número de teléfono, generalmente los navegadores permiten insertar tanto dígitos como letras y algunos símbolos.

### Entrada de color

Cuando se necesita que el usuario proporcione un color, se recomienda indicar como tipo `color`. Veamos un ejemplo:

Color favorito `<input type="color" name="favoriteColor">`

Color favorito

Generalmente, cuando el usuario hace clic en el botón de color, el navegador muestra una paleta similar a la siguiente:



### Atributo placeholder

La mayoría de los elementos de entrada presentados pueden mostrar una pequeña ayuda en el cuadro texto que proporciona más información al usuario sobre lo que debe introducir. Esto se puede proporcionar mediante el atributo **placeholder**. Veamos un ejemplo ilustrativo:

Usuario `<input type="text" name="username" placeholder="Tu dirección de correo">`

Usuario

Cuando el usuario comienza a introducir datos, el texto de ayuda desaparece.

### Otros atributos importantes

El elemento `<input>` presenta los siguientes atributos que pueden utilizarse con la mayoría de los tipos:

Atributo	Tipo de datos	Descripción
<b>autofocus</b>	booleano	Indica si al cargarse la página el control de entrada de datos debe posarse en el elemento. Sólo uno de los elementos debe de estar marcado con este atributo.
<b>autocomplete</b>	texto	Indica si el navegador debe asistir al usuario con valores anteriormente introducidos por el usuario: <ul style="list-style-type: none"> <li><b>on</b>, sí.</li> <li><b>off</b>, no.</li> </ul> Es posible definir este atributo a nivel de formulario, esto es, en el elemento <code>&lt;form&gt;</code> . Esto hará que se aplique a todos sus elementos de entrada.
<b>readonly</b>	booleano	Indica si el elemento es de sólo lectura.
<b>value</b>	texto	Valor predeterminado del elemento. Se mostrará cuando se cargue el elemento.
<b>minlength</b>	número	Número mínimo de caracteres que debe introducir el usuario. Se utiliza con los tipos <b>text</b> , <b>email</b> , <b>password</b> , <b>tel</b> y <b>url</b> .
<b>maxlength</b>	número	Número máximo de caracteres que puede introducir el usuario. Se utiliza con los tipos <b>text</b> , <b>email</b> , <b>password</b> , <b>tel</b> y <b>url</b> .
<b>pattern</b>	texto	Expresión regular <b>JavaScript</b> que indica el patrón que debe seguir el valor introducido.

### Elementos de selección

Todavía no hemos visto todos los tipos de elemento `<input>` disponibles. A continuación, vamos a presentar aquellos que están relacionados con la selección de uno o más valores, dado un grupo de opciones.

## Casillas de confirmación

Una **casilla de confirmación** (*checkbox*) es un cuadro que puede seleccionarse, dando a entender que se asegura o corrobora lo indicado. Se definen mediante el tipo **checkbox**. Ejemplo:

```
<input type="checkbox" name="newsletter"> ¿Quiere suscribirse a nuestra newsletter?  
☑ ¿Quiere suscribirse a nuestra newsletter?
```

Si deseamos que la casilla de confirmación aparezca marcada de manera predeterminada o inicialmente, hay que indicar el atributo booleano **checked**.

## Botones de opción

Un **botón de opción** (*radio button*) es similar a una caja de confirmación, pero pertenece a un grupo. De tal manera que si uno del grupo es seleccionado, los demás se deselectan. Siendo sólo posible que haya como máximo uno seleccionado. Ejemplo:

Idioma de contacto ● Español ● Inglés ● Italiano

Se utiliza el tipo **radio**. Y ahora, hay que definir *tantos* elementos de este tipo como opciones haya. La agrupación se consigue mediante el atributo **name**, todos ellos deben tener el mismo valor. Por otra parte, a cada opción se le puede asignar un valor específico que lo diferencie de las demás opciones posibles. Esto se consigue mediante el atributo **value**.

Si necesitamos que una de ellas aparezca seleccionada de manera predeterminada, se indicará su atributo **checked**.

He aquí un ejemplo ilustrativo:

```
Idioma de contacto  
<input type="radio" name="lang" value="es" checked> Español  
<input type="radio" name="lang" value="en"> Inglés  
<input type="radio" name="lang" value="it"> Italiano
```

## Desplegables

Un **desplegable** (*dropdown list*) es similar a un botón de opciones, pero se presenta mediante una lista que se expande. Ejemplo:

Idioma de contacto

▼
Español
Inglés
Italiano

Este tipo de entrada de datos no se representa mediante el elemento **<input>**, sino a través de **<select>**, el cual contiene elementos **<option>** para cada uno de los elementos de la lista. No hay nada como un ejemplo ilustrativo, veamos uno:

```
Idioma de contacto  
<select name="lang">  
  <option value=""></option>  
  <option value="es" selected>Español</option>  
  <option value="en">Inglés</option>  
  <option value="it">Italiano</option>  
</select>
```

Cada opción se indica mediante un elemento **<option>** donde su atributo **<value>** indica el valor a remitir cuando la opción se encuentra seleccionada. **<select>** indica el valor de la opción seleccionada por el usuario mediante el atributo **selected** en la opción correspondiente.

Si deseamos permitir al usuario seleccionar varios elementos de la lista, hay que indicar el atributo booleano **multiple** de **<select>**.

## Desplegables de texto

Cuando usamos un **<select>**, el usuario sólo puede seleccionar uno o más valores de la lista. A veces, necesitamos que el usuario pueda seleccionar uno de los valores de la lista o bien indicar otro que no

esté en ella. Este tipo de desplegables se consigue mediante el uso conjunto de los elementos `<input>` y `<datalist>`.

Por un lado, `<input>` indica que el usuario tiene que introducir un valor manualmente. Por otro lado, con `<datalist>` se proporciona la lista de valores a mostrar al usuario. Veamos primero un ejemplo ilustrativo:

```
Navegador
<input name="browser" list="browserList" placeholder="Indica tu browser">
<datalist id="browserList">
  <option>Chrome</option>
  <option>Edge</option>
  <option>Firefox</option>
  <option>Opera</option>
  <option>Safari</option>
</datalist>
```



Las opciones se indican mediante el elemento `<datalist>` de manera muy similar a `<select>`. Pero no hay que olvidar asignarle un valor a su atributo `id`, para así poder referenciarlo en el `<input>`. Por su parte, el elemento `<input>` representa la entrada de datos e indica, mediante su atributo `list`, el identificador del `<datalist>` que contiene los datos sugeridos al usuario. Observe que son recomendaciones o sugerencias para el usuario. Podrá seleccionar cualquiera de ellos o añadir su propio valor. De ahí que el componente de entrada se represente mediante un cuadro de texto con un desplegable adjunto.

## Envío del formulario

El **envío del formulario** (*form submit*) es la operación mediante la cual se envía los datos introducidos por el usuario al servidor web para su procesamiento. Este procesamiento puede ser cualquier cosa como, por ejemplo, dar de alta un nuevo producto o realizar un pedido con la información recopilada.

En esta operación, son importantes los atributos `action`, `method` y `enctype` del elemento `<form>`. Y los botones de envío.

### Atributos del formulario relacionados con el envío

El URL al que remitir los datos recopilados por el formulario se indica mediante el atributo `action`. Cuando no se indica, se remite al URL del documento HTML actual.

Mediante el atributo `method` se indica el método HTTP usado por el navegador para remitir los datos al servidor. Los dos posibles valores son `post` y `get`.

Atendiendo al método HTTP elegido para enviar los datos, éstos se transmitirán en el cuerpo del mensaje o bien en la cadena de consulta del URL. Cuando se usa el método `POST`, los datos se envían en el cuerpo de la petición HTTP. En este caso, podemos indicarle al servidor el tipo de contenido mediante el atributo `enctype` del formulario o del atributo `formenctype` del botón de envío.

Veamos un ejemplo ilustrativo:

```
<form action="/form/login" method="post">
  Usuario <input type="text" name="username">
  Contraseña <input type="password" name="password">
  <input type="submit" value="Entrar">
</form>
```

### Botones de envío

Un formulario puede tener cero, uno o más **botones de envío** (*submit buttons*). A través de los cuales, el

usuario puede solicitar al navegador el envío de datos. Este tipo de botones también se define usando el elemento `<input>`, mediante el tipo `submit`. El texto a mostrar en el botón se indica mediante el atributo `value`. He aquí un ejemplo:

```
<input type="submit" value="Enviar">
```

Enviar

## Atributo name de los elementos de entrada

El envío de datos al servidor puede llevarlo a cabo el propio navegador o un pequeño *script* de `JavaScript` escrito para la ocasión. Cuando delegamos el envío en el navegador, es de vital importancia el atributo `name` de los elementos `<input>`. Para entender por qué, lo importante es comprender cómo se envían los datos.

El navegador envía cada elemento de entrada mediante un par `nombre=valor`. Utilizándose el valor del atributo `name` como nombre. Y como valor, lo introducido por el usuario. Veamos un ejemplo:

```
username=yo
```

Si el formulario tiene varios elementos de entrada, éstos se separan mediante un *ampersand* (`&`). Veamos un ejemplo:

```
username=yo&password=alataque
```

Así pues, siempre que deleguemos el envío de los datos en el navegador, debemos asegurarnos que los elementos que contienen datos introducidos por el usuario han definido su atributo `name`.

## Elemento label

El elemento `<label>` define un título que se asocia a un determinado elemento de entrada de datos. Generalmente, se indica antes que el `<input>` con objeto de describir lo que debe introducir el usuario. Veamos un ejemplo:

```
<label for="username">Usuario </label>
```

```
<input type="text" id="username" name="username" placeholder="Tu dirección de correo">
```

Usuario

El objeto `label` se asocia, mediante su atributo `for`, a aquel elemento de entrada cuyo identificador, atributo `id`, es el indicado.

Las etiquetas son de especial importancia con las casillas de confirmación y los botones de opciones. Si delimitamos estos elementos y sus textos asociados mediante un `<label>`, el usuario podrá realizar la selección haciendo clic tanto en la casilla o el botón como en su texto asociado. En otro caso, sólo en la casilla o el botón.

Ejemplo:

```
<label>
```

```
<input type="checkbox" name="newsletter"> ¿Quiere suscribirse a nuestra newsletter?
```

```
</label>
```

```
Idioma de contacto
```

```
<label><input type="radio" name="lang" value="es" checked> Español</label>
```

```
<label><input type="radio" name="lang" value="en"> Inglés</label>
```

```
<label><input type="radio" name="lang" value="it"> Italiano</label>
```

Como se puede observar, si `<label>` delimita al elemento de entrada, no hace falta indicar su atributo `for`.