

# Formularios

HTML5

# Formularios

Los formularios web son la interface más importante de todas, permiten a los usuarios insertar datos, tomar decisiones, comunicar información y cambiar el comportamiento de una aplicación.

Todo formulario HTML comienza con una etiqueta **form**, dentro de ella se colocan todos los controles del formulario.

Fundamentalmente la etiqueta **form** posee dos atributos:

**action**. Es el atributo que contiene la URL de la página web o servicios que procesará los datos del formulario.

**method**. Permite indicar qué instrucción http utilizaremos para pasar la información al destino de nuestro formulario. Las opciones habituales son **GET** y **POST**. No se distingue entre mayúsculas y minúsculas. La diferencia es que GET genera una cadena de búsqueda en la URL que contiene los datos del formulario; POST, por su parte, pasa los datos pero de forma más oculta.

## El elemento <form>

### Plantilla básica para formularios

```
<!DOCTYPE html>
<html lang="es">
  <head> <title>Formularios</title> </head>
  <body>
    <section>
      <form name="miformulario" id="miformulario" method="get">
        <input type="text" name="nombre" id="nombre">
        <input type="submit" value="Enviar">
      </form>
    </section>
  </body>
</html>
```

## El elemento <input>

El elemento más importante en un formulario es **<input>**. Este elemento puede cambiar sus características gracias al atributo **type** (tipo). Este atributo determina qué clase de entrada es esperada desde el usuario. Los tipos disponibles hasta el momento eran el multipropósitos **text** (para textos en general) y solo unos pocos más específicos como **password** o **submit**. HTML5 ha expandido las opciones incrementando de este modo las posibilidades para este elemento.

En HTML5 estos nuevos tipos no solo están especificando qué clase de entrada es esperada sino también diciéndole al navegador qué debe hacer con la información recibida. El navegador procesará los datos ingresados de acuerdo al valor del atributo **type** y validará la entrada o no.

## Valores del atributo type del elemento input - text

### Cuadros de texto, input type="text"

Los cuadros de texto permiten recoger texto que escriba el usuario

```
<input type="text" name="nombre" />
```

podemos utilizar estos atributos:

- **value**. Da un valor inicial al cuadro, se usa para indicar un texto de ayuda al relleno.
- **id**. Identificador del cuadro. La W3C (organismo que estandariza XML y HTML) recomienda usar **id** en lugar de **name**, pero lo cierto es que PHP recoge los valores gracias a **name**, no funciona con **id**. Por lo que para más seguridad los diseñadores suelen utilizar ambos atributos.

#### Ejemplo:

```
<form action="control_form.php" method="get">  
  Escriba su nombre y apellidos <input type="text" name="nombre" />  
</form>
```

Ese código da como resultado: Escriba su nombre y apellidos

### Atributos de los cuadros de texto

Atributos	Posibles valores	uso
maxlength	Números	Indica el máximo número de caracteres que se le permitirá escribir al usuario.
size	Números	Anchura, en caracteres, del cuadro de texto. No tiene sentido que sea mayor que el anterior (sí que sea menor)
value	Texto	Permite rellenar el cuadro con un texto inicial

### Cuadro de contraseñas, input type="password"

Funcionan como los cuadros de texto, sólo que el texto que se introduce se oculta, mostrando sólo puntos o asteriscos. La sintaxis es:

```
<input type="password" name="nombre" />
```

Usa los mismos atributos que los cuadros de texto. Si usamos método GET, la contraseña es visible en la parte superior del navegador. Con POST esto no ocurre, pero aún así podríamos averiguarla. Por ello lo ideal es pasar cifrada la contraseña.

### Botones

Los botones son controles del formulario en los que no se puede escribir, lo que sí permiten es cargar una imagen mediante su atributo `src` (que funciona como el atributo `src` de las imágenes)

1. **Botón de envío.** Sirve para llevar a cabo la comunicación entre el formulario y la página que recoge sus datos. La sintaxis es:

```
<input type="submit" value="texto del botón" />
```

En cuanto se pulsa este botón, los datos del resto de controles se envían a la página receptora del formulario (que se corresponde a la URL indicada en el apartado **action** de la etiqueta **form**).

2. **Botón de restablecer**

Anula los datos que se han introducidos en los controles del formulario. Deja todos los controles en su estado inicial

```
<input type="reset" value="texto del botón" />
```

3. **Botón genérico**

Un botón genérico se marca indicando **type="button"** en la etiqueta **type**. En los formularios no se usa para enviar o configurar la información, Se utiliza normalmente para capturar su pulsación (mediante **JavaScript** es lo más habitual) y responder en consecuencia

```
<input type="button" value="texto del botón" />
```

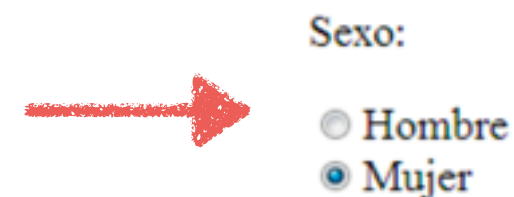
## Valores del atributo type del elemento input - radio y checked

### Botones de radio

Permite elegir una de entre varias opciones. Todas las opciones deben de tener el mismo nombre y sólo cambia el valor de las mismas. El atributo **checked** (que sólo admite el valor **checked**) hace que el botón en el que se usa, aparezca chequeado por defecto.

Ejemplo:

```
<form action="control_form.php" method="get">
  <p>Sexo:</p>
  <input type="radio" name="sexo" value="hombre"/>Hombre<br />
  <input type="radio" name="sexo" value="mujer" checked="checked"/>Mujer<br />
</form>
```

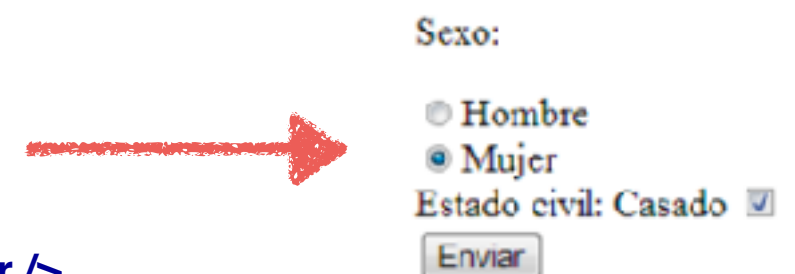


### Casillas de verificación

Se usan igual que los botones de radio, aunque no se recomienda agrupar varios con el mismo nombre; no porque no se pueda, sino porque las casillas se asocian a valores que se activan o desactivan. La diferencia es que el tipo (**type**) es **checkbox** en lugar de **radio**. También posee el atributo **checked** para que inicialmente la casilla aparezca pulsada.

Ejemplo:

```
<form action="control_form.php" method="get">
  <p>Sexo:</p>
  <input type="radio" name="sexo" value="hombre"/>Hombre<br />
  <input type="radio" name="sexo" value="mujer" checked="checked"/>Mujer<br />
  Estado civil: Casado <input type="checkbox" name="estadocivil" value="casado" checked="checked"/> <br />
  <input type="submit" value="Enviar" />
</form>
```

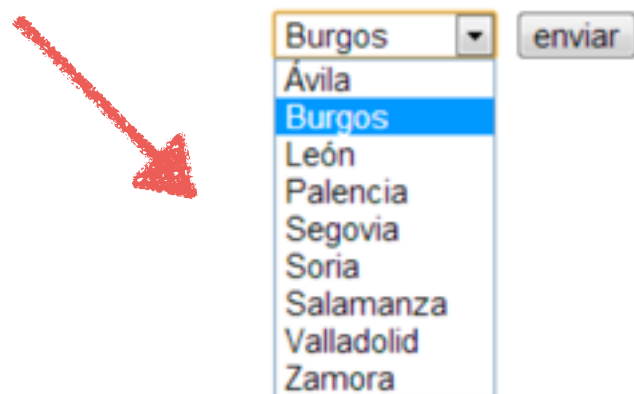


## Valores del atributo type del elemento input

### Cuadros combinados

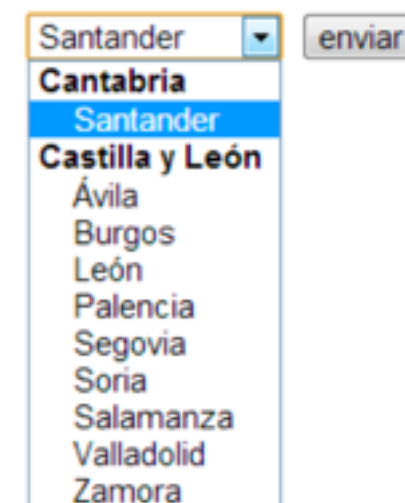
Un cuadro combinado permite el uso de una lista de opciones en la que se puede elegir una de ellas. Todo cuadro comienza con una etiqueta **select** que es la encargada de dar nombre (**name**) al control. Dentro cada opción del cuadro se indica con una etiqueta **option** a la que se da valor mediante el atributo **value**. Dentro de la etiqueta **option** se coloca el texto que verá el usuario.

```
<form action="control_form.php" method="get">
<select name="provincia">
  <option value="av">Ávila</option>
  <option value="bu">Burgos</option>
  <option value="l">León</option>
  <option value="p">Palencia</option>
  <option value="sg">Segovia</option>
  <option value="so">Soria</option>
  <option value="sa">Salamanca</option>
  <option value="va">Valladolid</option>
  <option value="za">Zamora</option>
</select>
<input type="submit" value="enviar"/> </form>
```



Se pueden agrupar opciones dentro del cuadro usando el elemento **optgroup**

```
<form action="control_form.php" method="get">
  <select name="provincia">
    <optgroup label="Cantabria">
      <option value="s">Santander</option>
    </optgroup>
    <optgroup label="Castilla y León">
      <option value="av">Ávila</option>
      <option value="bu">Burgos</option>
      <option value="l">León</option>
      <option value="p">Palencia</option>
      <option value="sg">Segovia</option>
      <option value="so">Soria</option>
      <option value="sa">Salamanca</option>
      <option value="va">Valladolid</option>
      <option value="za">Zamora</option>
    </optgroup>
  </select>
  <input type="submit" value="enviar"/>
</form>
```





## Valores del atributo type del elemento input

### Cuadro de selección de archivo

Cuando se usa **type="file"** en una etiqueta **input**, entonces aparece un botón que al pulsarle hace aparecer un cuadro de selección de archivos mediante el cual podremos elegir un archivo. La ruta local a dicho archivo es lo que se guarda para ser enviado al sitio o página que recibe los valores del formulario

```
<form action="recogida1.php" method="get">
```

```
  Elija el archivo <input type="file" name="archivo" />
```

```
  <input type="submit" value="enviar"/><br />
```

```
</form>
```



### Cuadro de texto multilínea

La etiqueta **textarea** permite colocar un cuadro de texto de varias líneas para introducir un texto largo. El atributo **rows** permite indicar la altura en líneas de texto del cuadro y el atributo **cols**, la anchura en caracteres (los demás atributos son como los de la etiqueta **input type="text"**).

```
<form action="control_form.php" method="get">
```

```
  Escribe un texto descriptivo: <br />
```

```
  <textarea rows="10" cols="40" name="texto"></textarea><br />
```

```
  <input type="submit" value="enviar"/><br />
```

```
</form>
```

Escribe un texto descriptivo:



enviar

### Agrupación de controles, fieldset

La etiqueta **fieldset** permite agrupar controles. Visualmente los controles aparecerán recuadrados y se suele utilizar una etiqueta inmediatamente interior a **fieldset** que es **legend** que contiene el texto que encabezará al grupo de controles.

```
<form action="control_form.php" method="get">
```

```
  <fieldset > <legend>Datos personales </legend>
```

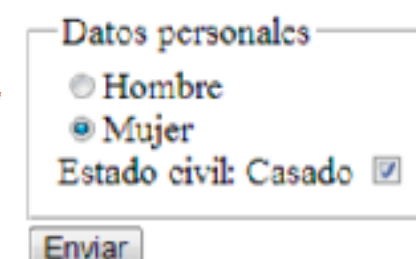
```
    <input type="radio" name="sexo" value="hombre"/> Hombre<br />
```

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

```
    Estado civil: Casado <input type="checkbox" name="estadocivil" value="casado" checked="checked"/>
```

```
  </fieldset>
```

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



# Formularios - HTML5

## Valores del atributo type del elemento input

### Tipo email

```
<input type="email" name="miemail" id="miemail">
```

El texto insertado en el campo generado por el código será validado por el navegador como un email. Si la validación falla, el formulario no será enviado.

### Tipo search (búsqueda)

```
<input type="search" name="busqueda" id="busqueda">
```

Al detectar este tipo de campo algunos navegadores cambiarán el diseño del elemento para ofrecer al usuario un indicio de su propósito

### Tipo url

```
<input type="url" name="miurl" id="miurl">
```

Está destinado a recibir solo URLs absolutas y retornará un error si el valor es inválido.

### Tipo tel

```
<input type="tel" name="telefono" id="telefono">
```

Este tipo de campo es para números telefónicos

### Tipo number

```
<input type="number" name="numero" id="numero" min="0" max="10" step="5">
```

es sólo válido cuando recibe una entrada numérica. Existen algunos atributos ( opcionales) nuevos que pueden ser útiles para este campo:

**min** El valor de este atributo determina el mínimo valor aceptado para el campo.

**max** El valor de este atributo determina el máximo valor aceptado para el campo.

**step** El valor de este atributo determina el tamaño en el que el valor será incrementado o disminuido en cada paso. Por ejemplo, si declara un valor de 5 para step en un campo que tiene un valor mínimo de 0 y máximo de 10, el navegador no le permitirá especificar valores entre 0 y 5 o entre 5 y 10. Por defecto toma el valor 1.



## Valores del atributo type del elemento input

### Tipo range

```
<input type="range" name="numero" id="numero" min="0" max="10" step="5">
```

permite al usuario seleccionar un valor a partir de una serie de valores o rango. Normalmente es mostrado en pantalla como una puntero deslizable o un campo con flechas para seleccionar un valor entre los predeterminados

El tipo **range** usa los atributos **min** y **max** para configurar los límites del rango. También puede utilizar el atributo **step** para establecer el tamaño en el cual el valor del campo será incrementado o disminuido en cada paso. Podemos declarar el valor inicial utilizando el atributo **value**

### Tipo date

```
<input type="date" name="fecha" id="fecha">
```

Gracias al tipo **date** ahora es el navegador el que se encarga de construir las herramientas necesarias para facilitar el ingreso de este tipo de datos

### Tipo week

```
<input type="week" name="semana" id="semana">
```

Este tipo de campo ofrece una interface similar a date, pero solo para seleccionar una semana completa. Por ejemplo 2017-W40 (40 es el número de semana en el año 2017)


### Tipo month

```
<input type="month" name="mes" id="mes">
```

### Tipo time

```
<input type="time" name="hora" id="hora">
```

El tipo de campo time es similar a date, pero solo para la hora



```
<form action="time-input-html5.php" name="formulario">
  <input type="time" name="hora" value="11:45:00" max="22:30:00" min="10:00:00" step="1">
  <input type="submit" value="enviar">
</form>
```

### Tipo datetime /datetime-local (tipo datetime sin la zona horaria )

```
<input type="datetime" name="fechahora" id="fechahora">
```

El tipo de campo **datetime** es para ingresar fecha y hora completa, incluyendo la zona horaria

### Tipo color

```
<input type="color" name="micolor" id="micolor">
```

provee una interface predefinida para seleccionar colores. Normalmente el valor esperado para este campo es un número hexadecimal, como #00FF00.

## Nuevos atributos de input

### Atributo placeholder

```
<input type="search" name="busqueda" id="busqueda" placeholder="escriba su búsqueda">
```

Especialmente en tipos de campo search, pero también en entradas de texto normales, el atributo placeholder representa una sugerencia corta para ayudar al usuario a ingresar la información correcta. El valor de este atributo es presentado en pantalla por los navegadores dentro del campo, como una marca de agua que desaparece cuando el elemento es enfocado.

### Atributo required

```
<input type="email" name="miemail" id="miemail" required>
```

Cuando el atributo required es incluido, la entrada será válida sólo si se cumplen las dos condiciones: que el campo no esté vacío y que el valor ingresado esté de acuerdo con los requisitos del tipo de campo.

### Atributo multiple

```
<input type="email" name="miemail" id="miemail" multiple>
```

Permite la inserción de múltiples valores separados por coma, y cada uno de ellos será validado por el navegador como una dirección de email.

### Atributo autofocus

```
<input type="search" name="busqueda" id="busqueda" autofocus>
```

El atributo autofocus enfocará la página web sobre el elemento seleccionado pero considerando la situación actual. No moverá el foco cuando ya haya sido establecido por el usuario sobre otro elemento.

### Atributo pattern

```
<input pattern="[0-9]{5}" name="codigopostal" id="codigopostal" title="inserte los 5 números de su código postal">
```

El atributo **pattern** es para propósitos de validación. Usa expresiones regulares para personalizar reglas de validación. Puede incluso incluir un atributo **title** para personalizar mensajes de error.

# Formularios - HTML5

## Nuevos atributos de input - novalidate y formnovalidate

**novalidate** para evitar que el formulario completo sea validado o **formnovalidate** para hacer lo mismo con elementos individuales.

Por ejemplo, para el siguiente caso se requiere el ingreso de un número de entre 13 y 16 dígitos:

```
<input type="text" pattern="[0-9]{13,16}" title="A credit card number" />
```

Si queremos pasar las validaciones al lado del servidor, la forma más sencilla de deshabilitar la validación para formularios con HTML5 es añadir el atributo **novalidate** al formulario.

```
<form method="POST" novalidate>
...
</form>
```

O sus alternativas correspondientes:

```
<form method="POST" novalidate="novalidate">
...
</form>

<form method="POST" novalidate="">
...
</form>
```

Otra alternativa que también puede ser útil: usar el atributo **formnovalidate**.

Un botón de **submit** con el atributo **formnovalidate** tiene el mismo efecto que poner **novalidate** en la etiqueta de formulario directamente, pero tiene una ventaja añadida: puede haber múltiples campos **submit** en un formulario lo que permite tener la opción de validar o no el formulario cuando se envía. Esto puede ser muy útil para, por ejemplo, guardar el formulario para continuar más tarde, o procesar algunos valores aunque no se haya completado del todo:

```
<form>
  <table>
    <tr>
      <td>Nombre</td>
      <td> <input type="text" name="nombre" required /> </td> </tr>
    <tr>
      <td>Apellido</td>
      <td> <input type="text" name="apellido" required /> </td> </tr>
    <tr>
      <td></td>
      <td>
        <input type="submit" value="Enviar (con validación)" /><br/>
        <input type="submit" value="Completar más tarde (sin validación)" formnovalidate />
      </td> </tr>
  </table>
</form>
```

# Formularios - HTML5

## Nuevos atributos de input

El atributo **autocomplete**, provee medidas de seguridad adicionales para el formulario completo o elementos individuales

- **on**: Para activar autocomplete
- **off**: Para desactivarlo.

Atributo para autocompletar datos en formularios

Este atributo se puede **aplicar a los input** del formulario, o a **todo el formulario** cuando queremos que todos los elementos compatibles con este elemento tengan autocomplete.

Si el atributo **se aplica a nivel de formulario**, todos los campos que acepten el atributo autocompletarán con contenidos anteriormente enviados. Posteriormente a nivel de input es posible deshabilitar ese atributo.

Lo mismo sucede en el caso opuesto, si ponemos al formulario autocomplete=off, posteriormente podemos activar inputs individualmente

```
<form action="#" autocomplete="on">  
  Nombre: <input type="text" name="nombre" autocomplete="on"/>  
  Apellidos: <input type="text" name="apellidos" autocomplete="off" />  
</form>
```

### Inputs compatibles con el atributo autocomplete HTML5

Este atributo se puede usar en los siguientes inputs:

- text, search, number, url, tel, email, password, range, color, date, datetime, datetime-local, month, time y week

**Ejemplo del atributo autocomplete de HTML5** para que probéis el envío del formulario varias veces, así conseguiremos que los inputs se autocompleten con información previamente enviada.

```
<form action="015-autocomplete-html5.php" name="formulario" method="POST">  
  Introduzca su nombre: <input type="text" name="nombre" autocomplete="on" />  
  Introduzca su edad: <input type="text" name="edad" autocomplete="on" />  
  <input type="submit" value="Enviar"/>  
</form>
```



Introduzca su nombre:

Introduzca su edad:

# Formularios - HTML5

## Nuevos atributos de input

### Atributo form

nos permite declarar elementos para un formulario fuera del ámbito de las etiquetas <form> . Hasta ahora, para construir un formulario teníamos que escribir las etiquetas <form> de apertura y cierre y luego declarar cada elemento del formulario entre ellas. En HTML5 podemos insertar los elementos en cualquier parte del código y luego hacer referencia al formulario que pertenecen usando su nombre y el atributo form

```
<!DOCTYPE html>
<html lang="es">
  <head> <title>Formularios</title> </head>
  <body>
    <nav> <input type="search" name="busqueda" id="busqueda" form="formulario"> </nav>
    <section>
      <form name="formulario" id="formulario" method="get">
        <input type="text" name="nombre" id="nombre">
        <input type="submit" value="Enviar">
      </form>
    </section>
  </body>
</html>
```

## Nuevos ELEMENTOS para formularios

### elemento <datalist>

El elemento <datalist> es un elemento específico de formularios usado para construir una lista de ítems que luego, con la ayuda del atributo list, será usada como sugerencia en un campo del formulario.

```
<datalist id="informacion">
  <option value="123123123" label="Teléfono 1">
  <option value="456456456" label="Teléfono 2">
</datalist>
```

Este elemento utiliza el elemento **<option>** en su interior para crear la lista de datos a sugerir. Con la lista ya declarada, lo único que resta es referenciarla desde un elemento <input> usando el atributo **list**:

```
<input type="tel" name="telefono" id="telefono" list="informacion">
```