

Formularios

Table of Contents

1	Introducción.....	3
2	Proceso general.....	4
3	Creación de formularios.....	5
3.1	El formulario: <form>.....	5
3.2	Atributos comunes de los controles.....	6
3.3	Los Controles.....	8
3.4	Recepción de los datos.....	10
3.4.1	Lenguaje de Script del lado Servidor.....	10
3.4.2	Envío de datos a correo.....	11
3.4.3	Visibilidad de los datos enviados: el atributo method.....	11
4	Elementos de los Formularios HTML.....	14
4.1	Elementos Input.....	14
4.1.1	Caja de texto: <input type="text">.....	14
4.1.2	Caja de texto de contraseña: <input type="password">.....	15
4.1.3	Número: <input type="number">.....	15
4.1.4	Fecha: <input type="date">.....	17
4.1.5	Casilla de verificación: <input type="checkbox">.....	17
4.1.6	Botón radio: <input type="radio">.....	19
4.1.7	Selector de color: <input type="color">.....	21
4.1.8	Rango: <input type="range">.....	22
4.1.9	Área de texto: <textarea>.....	22
4.1.10	Menú: <select>.....	23
4.1.11	El atributo disabled.....	27
4.1.12	Botón general mediante <input> o <button>.....	27
4.1.13	Búsqueda: <input type="search">.....	28
4.1.14	Teléfono: <input type="tel">.....	28
4.1.15	URL: <input type="url">.....	29
4.1.16	Dirección de correo electrónico: <input type="email">.....	29
4.1.17	Mes: <input type="month">.....	30
4.1.18	Semana: <input type="week">.....	30
4.1.19	Fecha y hora: <input type="datetime-local">.....	30
4.1.20	Selector de archivo: <input type="file">.....	30
4.1.21	Imagen: <input type="image">.....	31
4.2	Grupos de controles: <fieldset>.....	31
4.3	Accesibilidad: <label>.....	32
4.3.1	Relación implícita.....	32
4.3.2	Relación explícita: el atributo for.....	33

1 Introducción

Los formularios en HTML sirven al propósito de recolectar información proporcionada por los visitantes del sitio, la cual es luego enviada nuevamente al servidor. Para su correcto funcionamiento es importante que el formulario provisto en HTML sea acompañado de un código del lado servidor, al que denominaremos "agente procesador", que se encargará de recibir y procesar la información como el autor vea conveniente. Este procesamiento puede consistir en, por ejemplo, el almacenamiento de la información o su envío mediante correo electrónico.

Esta parte No se realiza en HTML, sino normalmente en Lenguajes del Lado servidor, con acceso a BDD

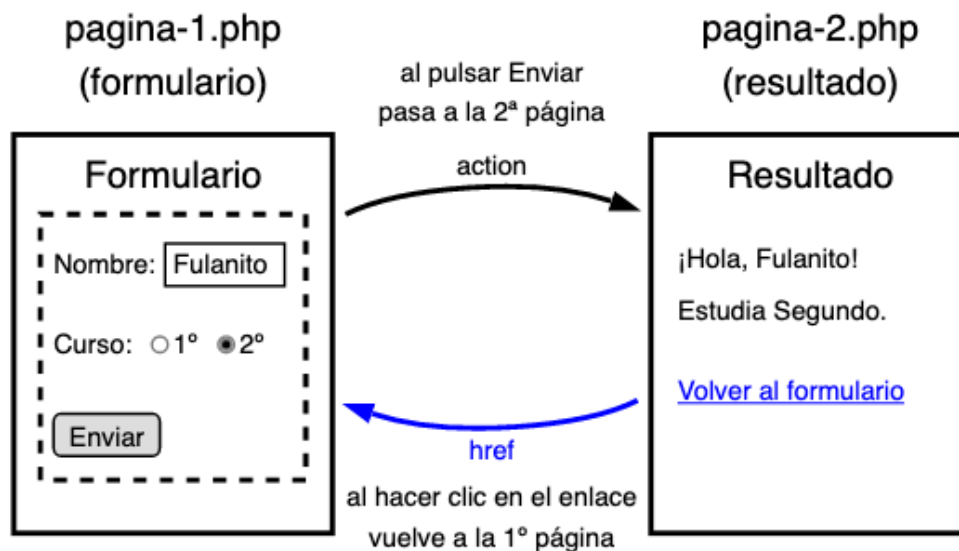
La parte de HTML de un formulario ([form](#)) es básicamente un contenedor para controles. Cada control en un formulario está pensado para recolectar información ingresada por los usuarios, en formas que pueden ir desde líneas de texto a subida de archivos, pasando por opciones, fechas, contraseñas y mucho más. Una vez que los usuarios han rellenado el formulario con los datos, pueden enviarlo al servidor para que el agente procesador administre la información recolectada y la guarde en dichos entornos del lado Servidor.

Veamos una visión general de este proceso con PHP, uno de los lenguajes que se usan de manera habitual para esta finalidad, de hecho WordPress uno de los CMS mas usados del mercado esta programado en PHP:

2 Proceso general

Los formularios son los elementos de las páginas web que permiten a un usuario introducir información en una página web. Así, los formularios nos permiten crear páginas web PHP interactivas en las que el contenido de la página se adapta a la información proporcionada por el usuario.

En el caso más simple, se suelen distribuir las tareas en dos páginas independientes:



- Inicialmente, el navegador muestra la primera página, que contiene el formulario. El usuario completa los controles y pulsa el botón Enviar.
- Al pulsar el botón Enviar, el navegador muestra la segunda página. Al solicitar la segunda página al servidor, el navegador envía también la información introducida por el usuario en el formulario. El programa PHP que genera la segunda página utiliza la información para personalizar el contenido.
- Normalmente, la segunda página incluye un enlace que permite al usuario volver de nuevo a la primera página para empezar de nuevo el proceso.

El ejemplo siguiente corresponde a la imagen anterior (la página con el formulario se llama form-1-1.php o un form-1-1.html según lo construyas y la página que recibe los datos se llama form-1-2.php):

Formulario

Nombre:

Curso: ☐ 1º ☐ 2º

Vamos a ver como construirlas y usarlas !!!

3 Creación de formularios

Las etiquetas de HTML que permiten la creación de formularios y de los diferentes tipos de controles (cajas de texto, casillas de verificación, botones radio, etc.) pertenecen a la creación de Páginas web HTML y hojas de estilo CSS que no forma parte de esta formación.

Comencemos con la información base, sobre las etiquetas y atributos mas usados.

3.1 El formulario: <form>

Un formulario es un conjunto de controles (botones, cajas de texto, casillas de verificación, botones radio, etc) que permiten al usuario introducir datos y enviarlos al servidor web para su procesamiento.

```
<form action="ejemplo.php" method="get">
  <p>Nombre: <input type="text" name="nombre" size="40"></p>
  <p>Año de nacimiento: <input type="number" name="nacido" min="1900"></p>
  <p>Sexo:
    <input type="radio" name="hm" value="h"> Hombre
    <input type="radio" name="hm" value="m"> Mujer
  </p>
  <p>
    <input type="submit" value="Enviar">
    <input type="reset" value="Borrar">
  </p>
</form>
```

La etiqueta que delimita un formulario es la etiqueta <form> ...</form>.

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

- action: contiene el nombre del agente que procesará los datos remitidos al servidor (**por ejemplo, un script de PHP, o el envío a un correo**)
- method: define la manera de enviar los datos al servidor. Los valores posibles son:
 - get: los valores enviados se añaden a la dirección indicada en el atributo action
 - post: los valores se envían de forma separada (también es el metodo cuando mandamos por correo)

Si el atributo method no está establecido, **el formulario se comporta como si el valor fuera get.**

La etiqueta <form> es un elemento de bloque. En su interior puede haber cualquier elemento típico de una página web (párrafos, imágenes, divisiones, listas, tablas, etc.), además de las etiquetas que crean los controles.

Los etiquetas que crean los controles en los formularios son <input>, <button>, <select>, <optgroup>, <option> y <textarea>.

Además, se pueden estructurar los controles con las etiquetas <fieldset> y <legend>.

Por último, la etiqueta <label> permite mejorar la accesibilidad de los controles.

El navegador envía únicamente los datos de los controles contenidos en el formulario. En una misma página puede haber varios formularios que envíen datos al mismo o a diferentes agentes.

Nombre:

Año de nacimiento:

Sexo: ☐ Hombre ☐ Mujer

3.2 Atributos comunes de los controles

Algunos atributos se pueden utilizar en varios tipos de controles.

El atributo type

El atributo type de la etiqueta <input> indica el tipo de control de que se trata.

En HTML 4.01 los tipos de <input> eran: botón para enviar submit,

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

botón para reiniciar reset,

```
<input type="reset" value="Borrar">
```

texto text, contraseña password, archivo file, imagen image y oculto hidden.

```
<input type="text" name="texto" placeholder="Escriba algo">
```

En HTML 5 se añadieron los tipos de <input>: botón button, número [number](#), búsqueda [search](#), teléfono [tel](#), dirección web [url](#), dirección de correo [email](#), fecha [date](#), hora [time](#), color [color](#) y rango [range](#).

En HTML 5.1 se añadieron los tipos de <input>: mes [month](#), semana week y fecha y hora local datetime-local.

El atributo name

El atributo name identifica al control.

Todos los controles de un formulario deben tener el atributo name porque, en general, sólo se envían los controles que lo tienen (aunque hay excepciones, como los controles de tipo imagen o los botones, que no los necesitan).

En general, los atributos name **no se deben repetir**, porque no se recibiría toda la información al enviarla. La única excepción es el botón radio.

El atributo value

El atributo value permite establecer el valor inicial de un control, aunque cada control lo utiliza de una forma ligeramente distinta y algunos controles no lo admiten (área de texto, archivo o imagen).

En algunos casos (botones) el valor se muestra al usuario y el usuario no puede modificarlo. En otros casos (cajas de texto, contraseña) el valor se muestra en la página y el usuario puede modificarlo. En otros casos (casillas de verificación, botones radio, opciones de menú, oculto) el valor no se muestra en la página y el usuario no puede modificarlo.

En algunos casos (botones, oculto, botones radio) el atributo value es necesario. En otros casos (casilla de verificación, opciones de menú) es conveniente. En otros casos (cajas de texto, contraseña, imagen) puede omitirse. En el caso del selector de archivo, los navegadores no lo admiten por motivos de seguridad.

El atributo required

El atributo required permite indicar qué controles es obligatorio rellenar para enviar el formulario.

```
<p>Nombre: <input type="text" name="nombre" required></p>
<p>Sexo:
  <input type="radio" name="hm" value="h" required> Hombre
  <input type="radio" name="hm" value="m" required> Mujer
</p>
```

Nombre:

Sexo: ☐ Hombre ☐ Mujer

El atributo placeholder

El atributo placeholder permite mostrar en los controles de texto un texto que desaparece al escribir en el control.

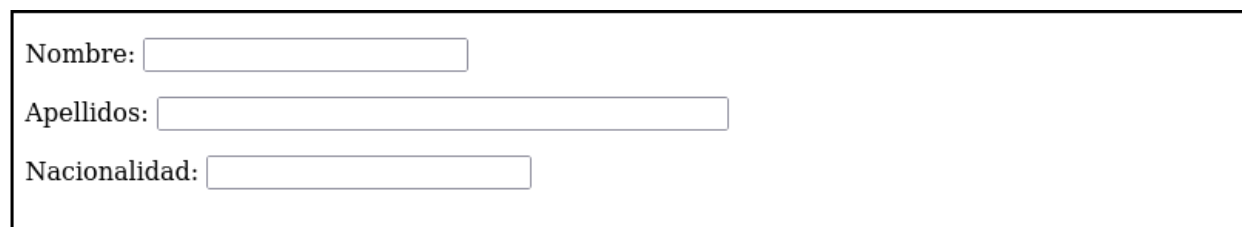
```
<p>Apellidos:
  <input type="text" name="apellidos" size="40" placeholder="Escriba sus dos
  apellidos">
</p>
```

Apellidos:

El atributo size

El atributo size permite establecer la longitud de los controles de texto ([text](#), [password](#), [search](#), etc.). Por omisión, las cajas suelen tener 20 caracteres de longitud.

```
<p>Nombre: <input type="text" name="nombre"></p>  
<p>Apellidos: <input type="text" name="apellidos" size="40"></p>  
<p>Nacionalidad: <input type="text" name="nacionalidad" size="20"></p>
```



3.3 Los Controles

Controles:<input> y otros

Los controles son los elementos que permiten al usuario introducir información.

La mayoría de controles se obtienen mediante mediante la etiqueta vacía `<input>`.

En HTML 4.01, los controles disponibles eran:

botón para enviar `<input type="submit">`,

botón para reiniciar `<input type="reset">`, botón `<button>`, texto `<input type="text">`, contraseña `<input type="password">`, archivo `<input type="file">`, imagen `<input type="image">`, oculto `<input type="hidden">`, área de texto `<textarea>` y menú `<select>`.

En HTML 5 se añadieron los controles:

botón general `<input type="button">`, número `<input type="number">`, búsqueda `<input type="search">`, teléfono `<input type="tel">`, dirección web `<input type="url">`, dirección de correo `<input type="email">`, fecha `<input type="date">`, hora `<input type="time">`, color `<input type="color">`, rango `<input type="range">`, opciones `<option>` (para `<datalist>`), lista de valores `<datalist>`, salida `<output>`, barra de progreso `<progress>` y medidor `<meter>`.

En HTML 5.1 se añadieron los controles:

mes `<input type="month">`, semana `<input type="week">` y fecha y hora local `<input type="datetime-local">`.

Botones: <input> y <button>

Los botones se crean mediante la etiqueta <input> o mediante la etiqueta <button>. La diferencia entre ellos es que <input> sólo puede contener texto, mientras que <button> permite incluir elementos html como imágenes. El contenido del botón <input> se define mediante el atributo **value**, por lo que sólo puede contener texto. El contenido del botón <button> se escribe dentro del elemento, por lo que puede incluir texto e imágenes.

Normalmente los botones no suelen llevar atributo **name**, pero se puede poner si se quiere (tendría sentido si un formulario contuviera dos botones de envío y quisiéramos saber en cuál de ellos ha hecho clic el usuario, pero esto no es muy habitual).

Botones Submit mediante <input> o <button>

El botón Submit es el que permite al usuario remitir los datos al servidor. Se crea mediante una etiqueta <input> o <button> cuyo atributo type tiene el valor submit.

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

Escriba algo:

Enviar

```
<button type="submit">  
Enviar  
</button>
```

Escriba algo:

 Enviar

Si un botón <button> no lleva el atributo type, se comporta como un botón de tipo submit.

Botones Reset mediante <input> o <button>

El botón Reset restablece los valores iniciales del formulario. Se crea mediante una etiqueta <input> o <button> cuyo atributo type tiene el valor reset.

```
<input type="reset" value="Borrar">
```

Caja de texto:

☐ Casilla de verificación

☐ Radio 1 ☐ Radio 2 ☐ Radio 3

Borrar

```
<button type="reset">
  Borrar
</button>
```

Caja de texto:

☐ Casilla de verificación

☐ Radio 1 ☐ Radio 2 ☐ Radio 3

 Borrar

Para mas información sobre la forma de crear un formulario ver el capítulo 4.

3.4 Recepción de los datos

Para la recepción de datos se pueden suar diferentes opciones, vamos a comentar dos de ellas:

3.4.1 Lenguaje de Script del lado Servidor

```
<FORM ACTION="/cgi-bin/registro.php" METHOD=post>
```

Cuando un formulario envía los datos a una página PHP, los datos están automáticamente a disposición del programa PHP en una matriz llamada **`$_REQUEST`**.

Cada dato recibido es un elemento de la matriz `$_REQUEST`: el índice del elemento es el nombre del control (salvo en el control de tipo imagen) y el valor del elemento es el dato introducido o seleccionado por el usuario en el control.

En el siguiente ejemplo, el programa que recibe los datos del formulario simplemente muestra el contenido de la matriz `$_REQUEST` utilizando la función `print_r()` que muestra los índices y valores de la matriz. La página con el formulario se llama `form-2-1.php` y la página que recibe los datos se llama `form-2-2.php`

```
<?php
// form-2-2.php
print "<pre>";
print_r($_REQUEST);
print "</pre>\n";
?>
```

EJEMPLO DE F

Escriba su nombre:

Que curso Estudias: ☐ Asir 1 ☐ Asir 2

Enviar

Borrar

Practica Grupo ASIR - IES infanta Elena

```
Array
(
    [nombre] => pedro
    [grupo] => 1
)
```

[Volver al formulario.](#)

Como se puede comprobar en el ejemplo anterior, el control de caja de texto crea siempre un elemento en la matriz \$_REQUEST (aunque la caja de texto se deje vacía), mientras que el botón radio solamente crea un elemento en la matriz \$_REQUEST si se elige una de las opciones del botón radioe.

```
Array
(
    [nombre] => sss
)
```

[Volver al formulario.](#)

3.4.2 Envió datos a correo

<FORM ACTION="mailto:pcorc@educa.madrid.org" METHOD=post>

La URL puede ser un <mailto:direccion>. De esta forma, la información recogida por el formulario será enviada por correo electrónico a la dirección especificada. Esto es útil si se quiere diseñar un formulario, pero no se dispone de conocimientos suficientes para desarrollar aplicaciones CGI.

Este método impide la validación de datos, y el procesamiento online posterior de los mismos.

3.4.3 Visibilidad de los datos enviados: el atributo method

El atributo method de la etiqueta <form> permite elegir si la información de los controles se incluye en la URL de llamada a la página (method="get") o se incluyen en las cabeceras HTTP (method="post"). **Si no se incluye el atributo, el comportamiento es el mismo que con get.**

La diferencia es que con el valor get se pueden ver en la barra de dirección los nombres de los controles y los valores introducidos por el usuario, mientras que con el valor post no. Los datos recibidos son los mismos y desde el punto de vista de la seguridad ambos métodos son equivalentes (es decir, inseguros si se utiliza http y seguros si se utiliza https).

Normalmente, los formularios disponibles en Internet utilizan el método post, de manera que las URL sean más "limpias" al no contener los datos de los formularios.

Pero en estos apuntes, los formularios utilizan el método get, de manera que se puedan simular fácilmente ataques de inyección, como se comenta en el apartado siguiente.

- Formulario con post

El formulario envía el valor de un control incluyéndolo en una cabecera HTTP, que no es visible para el usuario, como se ve en el ejemplo siguiente:

Formulario:

```
<form action="form-3-2.php" method="post">
  <p>Nombre: <input type="text" name="nombre"></p>

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

</form>

Barra del navegador:

<http://localhost/formulario/form-3-2.php>

Recepción del formulario:

```
<?php
print "<pre>";
print_r($_REQUEST);
print "</pre>\n";
?>
```

Es la misma

- Formulario con get

El formulario envía el valor de un control incluyéndolo al final de la URL, añadiendo un interrogante, el nombre del control, un signo igual y el valor (es decir, ?nombre=valor), como se puede comprobar en el ejemplo siguiente. Para poder ver las URL en su navegador, abra el ejemplo en una pestaña haciendo clic en el icono de enlace al ejemplo situado a la derecha del ejemplo.

Formulario:

```
<form action="form-1-1.php" method="get">
  <p>Nombre: <input type="text" name="nombre"></p>

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

Barra del navegador

(en el ejemplo 1 de antes anterior **<http://localhost/formulario/form-2-1.php?nombre=&grupo=1>**

Recepción del formulario:

```
<?php
print "<pre>";
print_r($_REQUEST);
print "</pre>\n";
?>
```

En caso de que haya varios controles que envíen información en un formulario con get, los nombres y valores aparecen en la barra de dirección separados por el carácter ampersand (&), como **nombre1=valor1&nombre2=valor2&...**, como se puede comprobar en el ejemplo siguiente. Para poder ver las URL en su navegador, abra el ejemplo en una pestaña haciendo clic en el icono de enlace al ejemplo situado a la derecha del ejemplo.

Formulario:

```
<form action="form-5-1.php" method="get">
  <p>Nombre: <input type="text" name="nombre"></p>

  <p>Apellidos: <input type="text" name="apellidos"></p>

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

</form>

Recepción del formulario:

```
<?php
print "<pre>";
print_r($_REQUEST);
print "</pre>\n";
?>
```

PHP incluye en la matriz `$_REQUEST` todos los datos que llegan junto con la petición de la página, tanto si están incluidos en la URL como si están incluidos en cabeceras HTTP.

PHP también crea otras dos matrices, `$_GET` y `$_POST`, que contienen los datos recibidos por uno de los métodos, **`$_GET` los recibidos en la URL (get) y `$_POST` los recibidos en cabeceras (post)**.

En el caso del envío por correo el formato de los datos es similar al explicado anteriormente para HTTP:

Asunto: Envío de formulario desde Firefox

nombre=Juan+Gomez&edad=35

4 Elementos de los Formularios HTML

4.1 Elementos Input

4.1.1 Caja de texto: `<input type="text">`

`<input type="text">`

Las cajas de texto de una sola línea se crean mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `text`.

`<input type="text" name="texto">`

Escriba algo:

Enviar

El atributo `value` (optativo) permite incluir un texto en la caja de texto para que el usuario lo modifique. HTML 5 introdujo el atributo `placeholder`, mucho más cómodo para el usuario, ya que no tiene que borrar el texto de ayuda.

`<input type="text" name="texto" value="Escriba algo">`

`<input type="text" name="texto" placeholder="Escriba algo">`

Lo mas básico:

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

4.1.2 Caja de texto de contraseña: <input type="password">

<input password>

Las cajas de texto de una sola línea específicas para **contraseñas** se crean mediante una etiqueta <input> cuyo atributo type tiene el valor password.

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

Un formulario web que contiene el texto "Contraseña:" seguido de un campo de entrada de tipo password (caja de texto con fondo grisado). Debajo del campo hay un botón rectangular con el texto "Enviar".

Al escribir en una caja de contraseña, en vez de letras aparecen puntos gruesos. Es importante señalar que estas cajas no proporcionan ninguna seguridad en la transmisión, simplemente ocultan al usuario lo que este escribe.

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

4.1.3 Número: <input type="number">

<input number>

Las cajas de texto de una sola línea específicas para **números** se crean mediante una etiqueta <input> cuyo atributo type tiene el valor number.

En principio, los valores admitidos por el control son números enteros.

Escriba un número:

<input type="number" name="numero">

Un formulario web que contiene el texto "Escriba un número:" seguido de un campo de entrada de tipo number (caja de texto con fondo grisado y una flecha de desplazamiento a la derecha). Debajo del campo hay un botón rectangular con el texto "Enviar".

El atributo step permite que el control admita números decimales. Los navegadores alertan al usuario cuando no ha escrito un valor correcto, como mínimo al enviar el formulario.

En el caso de los números decimales:

- Chrome admite la coma (,) y el punto (.) como separadores de la parte decimal.

- Firefox sólo admite la coma (,) como separador de la parte decimal.

Escriba un número cualquiera:

```
<input type="number" name="numero" step="any">
```

Escriba un número cualquiera:

Enviar

El atributo min establece un valor mínimo en la respuesta:

Escriba un número mayor o igual que 20:

```
<input type="number" name="numero" min="20">
```

Escriba un número mayor o igual que 20:

Enviar

El atributo max establece un valor máximo en la respuesta:

Escriba un número menor o igual que 100:

```
<input type="number" name="numero" max="100">
```

Escriba un número menor o igual que 100:

Enviar

El atributo step establece la cantidad, entera o decimal, en la que aumenta o disminuye el valor:

Escriba un múltiplo de 10:

```
<input type="number" name="numero" step="10">
```

Escriba un múltiplo de 10:

Enviar

Escriba un valor de 0,5 en 0,5:
<input type="number" name="numero" step="0.5">

Escriba un valor de 0,5 en 0,5:

Enviar

Para poder escribir cualquier número, el atributo step debe tener el valor any:

Escriba un número cualquiera:
<input type="number" name="numero" step="any">

Escriba un número cualquiera:

Enviar

4.1.4 Fecha: <input type="date">

<input date>

Las cajas de texto de una sola línea específicas para **fechas (días, meses, años)** se crean mediante una etiqueta <input> cuyo atributo type tiene el valor date. El dato se envía con el formato AAAA-MM-DD donde AAAA es el número de año, MM el número de mes y DD el número de día.

Hora: <input type="time" name="hora">

Hora:

Enviar

4.1.5 Casilla de verificación: <input type="checkbox">

<input checkbox>

Las casillas de verificación se crean mediante una etiqueta <input> cuyo atributo type tiene el valor checkbox.

<input type="checkbox" name="casilla">Casilla 1

☐ Casilla 1

Si el atributo checked tiene el valor checked, la casilla aparece marcada.

```
<input type="checkbox" name="casilla" checked>Casilla 1
```

☒ Casilla 1

Las casillas de verificación sólo se envían si se han marcado. El atributo value contiene el valor que envía el formulario si la casilla de verificación está marcada. Si el atributo value no está establecido, el formulario envía el valor on.

```
<input type="checkbox" name="casilla">Casilla 1
```

☐ Casilla de verificación

Enviar

```
<input type="checkbox" name="casilla" value="hola">Casilla 1
```

☐ Casilla de verificación

Enviar

resultado de los anteriores

El dato recibido es **on**.

[Volver al formulario.](#)

El dato recibido es **hola**.

[Volver al formulario.](#)

4.1.6 Botón radio: <input type="radio">

<input radio>

Los botones radio se crean mediante una etiqueta <input> cuyo atributo type tiene el valor radio.

```
<input type="radio" name="boton" value="1">Opción 1
```

☒ Opción 1

Los botones radio que tienen el mismo atributo name forman un grupo, es decir, que si se marca uno de ellos se desmarca automáticamente el resto.

```
<input type="radio" name="boton" value="1">Opción 1<br>
<input type="radio" name="boton" value="2">Opción 2<br>
<input type="radio" name="boton" value="3">Opción 3
```

☐ Opción 1
☐ Opción 2
☒ Opción 3

Si los atributos name no tienen el mismo valor, los botones radio son independientes (al marcar uno no se desmarca el otro):

```
<input type="radio" name="boton1" value="1">Opción 1<br>
<input type="radio" name="boton2" value="2">Opción 2
```

☒ Opción 1
☒ Opción 2

Los botones radio sólo se envían si se han marcado. El atributo value contiene el valor que envía el formulario si el botón radio está marcado. Si el atributo value no está establecido, el formulario envía el valor on, así que para poder saber cuál ha sido la opción elegida por el usuario es necesario establecer con valores distintos los atributos value de todos los elementos de un botón radio.

```
<input type="radio" name="boton"> Hombre<br>
<input type="radio" name="boton"> Mujer
```

```
<input type="radio" name="boton" value="hombre"> Hombre<br>
<input type="radio" name="boton" value="mujer"> Mujer
```

El dato recibido es **on**.

[Volver al formulario.](#)

El dato recibido es **mujer**.

[Volver al formulario.](#)

Si uno de los botones tiene el atributo checked con el valor checked, el botón aparece marcado de forma predeterminada.

☐ Opción 1
☒ Opción 2

Borrar

Una vez se ha hecho clic en uno de los botones radio, siempre habrá una opción seleccionada. Para desactivar un botón radio, se necesita un botón Reset.

```
<input type="radio" name="boton" value="1">Opción 1<br>
<input type="radio" name="boton" value="2">Opción 2<br>
<input type="reset" value="Borrar">
```

Nombre:

☐ Opción 1
☒ Opción 2

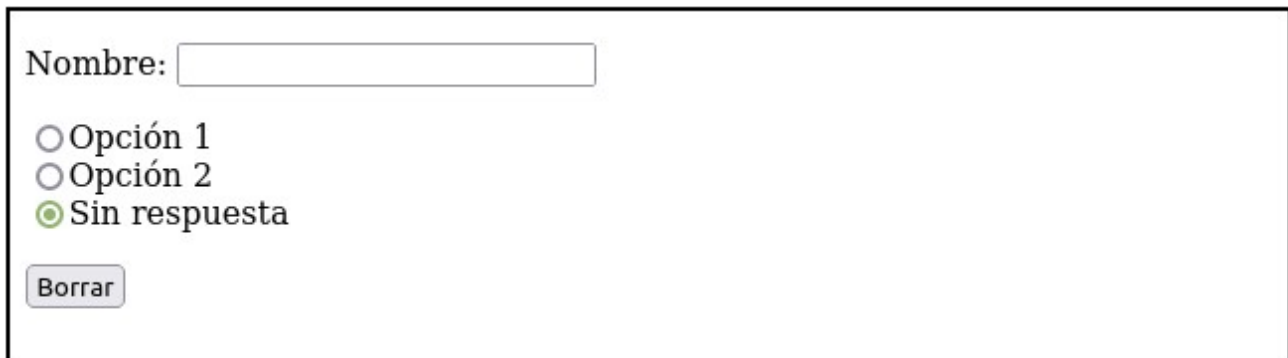
Borrar

El inconveniente del botón Reset es que reinicia todos los controles del formulario. Este problema no tiene una solución sencilla. Algunos recomiendan incluir **en los botones radio una opción neutra**, de manera que el usuario pueda cambiar su respuesta sin tener que reiniciar todo el formulario. El problema es que el usuario puede malinterpretar esa opción como una opción más y elegirla a propósito, por lo que se recomienda valorar en cada caso la mejor solución. Quizás en formularios con pocos controles no merezca

la pena incluir esa opción neutra, y hacerlo sólo en formularios con muchos controles en los que el usuario tendría que volver a completar todo el formulario si utiliza el botón Reset

```
<p>Nombre: <input type="text" name="nombre"></p>
```

```
<p>  
  <input type="radio" name="boton" value="1">Opción 1<br>  
  <input type="radio" name="boton" value="2">Opción 2<br>  
  <input type="radio" name="boton" value="0" checked>Sin respuesta  
</p>
```



Nombre:

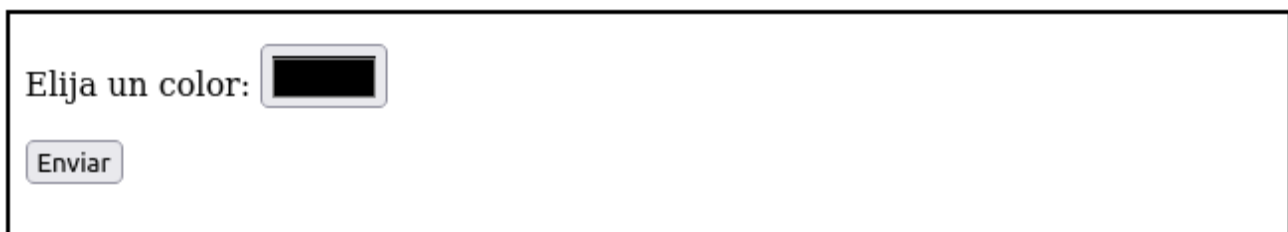
☐ Opción 1
☐ Opción 2
☒ Sin respuesta

4.1.7 Selector de color: `<input type="color">`

```
<input color>
```

El control de selector de color se crea mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `color`. Permite elegir un color mediante el selector de color proporcionado por el sistema operativo.

```
<input type="color" name="colorElegido">
```



Elija un color:

El atributo `value` permite especificar un color inicial. El color se debe indicar en formato hexadecimal (`#RRGGBB`).

```
<input type="color" name="colorElegido" value="#ff0000">
```

Elija un color:

4.1.8 Rango: `<input type="range">`

`<input range>`

El control de rango se crea mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `range`. El rango de valores predeterminado es de 0 a 100 (valores enteros).

Indique su nivel:

```
<input type="range" name="rango">
```

Indique su nivel:

Los atributos `min`, `max` y `step` permiten elegir respectivamente el valor mínimo, el máximo y el incremento.

Indique su nivel (de 1 a 10, de 3 en 3):

```
<input type="range" name="rango"
  min="1" max="10" step="3">
```

Indique su nivel (de 1 a 10, de 3 en 3):

4.1.9 Área de texto: `<textarea>`

`<textarea>`

Las cajas de texto de varias líneas se crean mediante la etiqueta `<textarea>`. Los atributos obligatorios `rows` y `cols` establecen el número de filas y columnas iniciales de la caja, aunque los navegadores permiten modificarlo arrastrando la esquina inferior derecha.

```
<textarea name="texto" rows="4" cols="20"></textarea>
```




```
<textarea name="texto" rows="3" cols="30"></textarea>
```



Si se escribe texto en el interior de la etiqueta, este se muestra en la caja de texto. HTML 5 introdujo el atributo placeholder que puede ser más cómodo para el usuario si tiene que borrar todo el texto incluido.

```
<textarea name="texto" rows="4" cols="40">Escriba algo</textarea>
```



```
<textarea name="texto" rows="4" cols="40" placeholder="Escriba algo"></textarea>
```



4.1.10 Menú: <select>

```
<select>
```

<option>

<optgroup>

Los menús se crean mediante la etiqueta <select>. Cada opción del menú se define mediante la etiqueta <option>.

```
<select name="menu">
  <option>Uno</option>
  <option>Dos</option>
  <option>Tres</option>
</select>
```

Elija una opción: Uno ▾

Enviar

El valor que se envía es el texto que aparece en el menú, salvo si el elemento <option> contiene el atributo value.

```
<select name="menu">
  <option>Uno</option>
  <option value="2">Dos</option>
  <option value="three">Tres</option>
</select>
```

La opción elegida es **2**.

[Volver al formulario.](#)

El atributo selected indica la opción por omisión.

```
<select name="menu">
  <option value="1">Uno</option>
  <option value="2">Dos</option>
  <option value="3" selected>Tres</option>
</select>
```


Elija una opción: Tres ▼

Enviar

Si ningún elemento posee el atributo selected, los navegadores muestran la primera opción del menú.

Elija una opción: Uno ▼

Enviar

El problema de mostrar una de las opciones de forma predeterminada es que al recibir el formulario no se puede saber si el usuario ha elegido realmente esa opción o simplemente no ha modificado el control. Por ello, se aconseja incluir una opción en blanco al principio de los menús

```
<select name="menu">
  <option value="0">...</option>
  <option value="1">Uno</option>
  <option value="2">Dos</option>
  <option value="3">Tres</option>
</select>
```

Elija una opción: ... ▼

Enviar

El atributo size permite definir la altura del control.

```
<select name="menu" size="3">
  <option value="1">Uno</option>
  <option value="2">Dos</option>
  <option value="3">Tres</option>
  <option value="4">Cuatro</option>
  <option value="5">Cinco</option>
</select>
```

Elija una opción:

Uno
Dos
Tres

Enviar

El atributo `multiple` permite elegir varias opciones simultáneamente (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.

```
<select name="menu[]" size="5" multiple>
  <option value="1">Uno</option>
  <option value="2">Dos</option>
  <option value="3">Tres</option>
  <option value="4">Cuatro</option>
  <option value="5">Cinco</option>
</select>
```

Elija una o varias opciones:

Uno
Dos
Tres
Cuatro
Cinco

Enviar

Se pueden agrupar opciones utilizando la etiqueta `<optgroup>`.

```
<select name="menu">
  <option selected>...</option>
  <optgroup label="Grupo1">
    <option value="1">Opción uno</option>
    <option value="2">Opción dos</option>
    <option value="3">Opción tres</option>
  </optgroup>
  <optgroup label="Grupo2">
    <option value="4">Opción cuatro</option>
    <option value="5">Opción cinco</option>
    <option value="6">Opción seis</option>
  </optgroup>
</select>
```

4.1.11 El atributo disabled

El atributo disabled permite deshabilitar el control. Una vez deshabilitado, el control ni siquiera puede coger el foco.

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



```
<input type="text" name="texto" disabled>
```

4.1.12 Botón general mediante <input> o <button>

```
<input button>
```

```
<button>
```

Los botones de tipo button se crean mediante la etiqueta <input> o mediante la etiqueta <button>. La diferencia entre ellos es que <input> sólo puede contener texto, mientras que <button> permite incluir elementos html como imágenes.

El contenido del botón <input> se define mediante el atributo value, por lo que sólo puede contener texto.

```
<input type="button" value="Este botón no hace nada">
```



El contenido del botón <button> se escribe dentro del elemento, por lo que puede incluir texto e imágenes.

```
<button type="button">
  
  Este botón no hace nada
</button>
```



Un botón de tipo button se suele asociar a un script.

```
<input type="button" onclick="mensaje('Ha pulsado el botón')" value="Botón">
```

```
<script>
  function mensaje(texto) {
    alert(texto);
  }
</script>
```

```
<button type="button" onclick="mensaje('Ha pulsado el botón')">
  
  Botón
</button>
```

```
<script>
  function mensaje(texto) {
    alert(texto);
  }
</script>
```

4.1.13 Búsqueda: <input type="search">

```
<input search>
```

Las cajas de texto de una sola línea específicas para **búsquedas** se crean mediante una etiqueta <input> cuyo atributo type tiene el valor search.

Búsqueda: <input type="search" name="busqueda" size="40">

Búsqueda:

La recomendación HTML 5 aclara que la diferencia entre text y search es únicamente de estilo y que el cuadro de búsqueda debería tener el estilo de los cuadros de búsqueda del sistema utilizado.

4.1.14 Teléfono: <input type="tel">

```
<input tel>
```

Las cajas de texto de una sola línea específicas para **números de teléfonos** se crean mediante una etiqueta <input> cuyo atributo type tiene el valor tel.

Teléfono: <input type="tel" name="telefono">

Teléfono:

La recomendación HTML 5 aclara que los formatos de números de teléfono siguen formatos tan variados que este control no obliga a utilizar ninguna sintaxis en particular.

En los teléfonos móviles, al escribir en una caja de texto de tipo tel se muestra un teclado numérico.

4.1.15 URL: `<input type="url">`

`<input url>`

Las cajas de texto de una sola línea específicas para **direcciones web (URL)** se crean mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `url`.

URL: `<input type="url" name="direccion" size="40">`

URL:

Actualmente los navegadores sólo comprueban que el texto escrito empiece por `http:` o `https:`, pero no obligan a que las URL sean sintácticamente válidas.

En los teléfonos móviles, al escribir en una caja de texto de tipo `url` se adapta ligeramente el teclado de texto, mostrándose el carácter `.`

4.1.16 Dirección de correo electrónico: `<input type="email">`

`<input email>`

Las cajas de texto de una sola línea específicas para **direcciones de correo electrónico** se crean mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `email`.

Correo: `<input type="email" name="correo" size="40">`

Correo:

Actualmente los navegadores sólo comprueban que el texto escrito contenga una arroba `@` y no contengan caracteres no permitidos, pero no obliga a que las direcciones sean válidas.

En los teléfonos móviles, al escribir en una caja de texto de tipo `email` se adapta ligeramente el teclado de texto, mostrándose el carácter `@`.

4.1.17 Mes: `<input type="month">`

`<input month>`

Las cajas de texto de una sola línea específicas para **meses (y años)** se crean mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `month`. El dato se envía con el formato AAAA-MM donde AAAA es el número de año y MM el número de mes.

Mes: `<input type="month" name="mes">`



The image shows a web form with a label "Mes:" followed by a text input field. Below the input field is a button labeled "Enviar". The input field is empty, representing the `<input type="month">` element.

Nota: Firefox no muestra un control específico para seleccionar el mes y el año

4.1.18 Semana: `<input type="week">`

`<input week>`

Las cajas de texto de una sola línea específicas para **semanas (y años)** se crean mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `week`. El dato se envía con el formato AAAA-WSS donde AAAA es el número de año y SS el número de semana.

Semana: `<input type="week" name="semana">`

Nota: Firefox no muestra un control específico para seleccionar la semana y el año,

4.1.19 Fecha y hora: `<input type="datetime-local">`

`<input datetime-local>`

Las cajas de texto de una sola línea específicas para **fechas (días, meses, años, horas y minutos)** se crean mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `datetime-local`. El dato se envía con el formato AAAA-MM-DDTHH-MM donde AAAA es el número de año, MM el número de mes, DD el número de día, HH son las horas y MM los minutos.

Fecha y hora: `<input type="datetime-local" name="fechaHora">`



The image shows a web form with a label "Fecha y hora local:" followed by a text input field. The input field contains the placeholder text "dd / mm / aaaa , -- : --" and a calendar icon. Below the input field is a button labeled "Enviar". The input field represents the `<input type="datetime-local">` element.

4.1.20 Selector de archivo: `<input type="file">`

`<input file>`

El selector de archivo se crea mediante una etiqueta `<input>` cuyo atributo `type` tiene el valor `file`.

```
<input type="file" name="archivo">
```



Tradicionalmente, los navegadores no han admitido el atributo `value` en el selector de archivo por motivos de seguridad, ya que podría utilizarse para **"robar" ficheros al usuario sin su autorización**. La recomendación HTML 5 indica que el atributo `value` se puede utilizar para seleccionar selectores de archivos distintos al general, pero los navegadores no parecen utilizar este atributo.

```
<input type="file" name="archivo" value="image/*">
```

4.1.21 Imagen: `<input type="image">`

```
<input image>
```

El control de tipo imagen inserta una imagen que funciona como un botón (aunque los navegadores no le dan relieve como a los botones). Al hacer clic en un punto de la imagen se envía el formulario (como si se hubiera pulsado un botón submit) y se envían las coordenadas del punto en el que se ha hecho clic (junto con los valores de los otros controles del formulario).

Este control debe incluir siempre el atributo `alt` (con un valor no vacío), para indicar el texto que deben mostrar los navegadores si la imagen no está disponible.

```
<input type="image" name="diana" src="diana.svg" alt="Diana">
```

4.2 Grupos de controles: `<fieldset>`

```
<fieldset>
```

```
<legend>
```

La etiqueta `<fieldset>` permite agrupar un conjunto de controles. Los navegadores muestran una caja alrededor de cada grupo de controles.

La etiqueta `<legend>` permite añadir una leyenda al `<fieldset>`. Los navegadores muestran la leyenda sobre el borde que rodea el grupo de controles.

```
<fieldset>
  <legend>Datos personales</legend>
  <p>Nombre: <input type="text" name="nombre" size="30"></p>
  <p>Edad: <input type="number" name="edad"></p>
</fieldset>
```

```
<fieldset>
  <legend>Cultura general</legend>
```

```

<p>Capital de Noruega:
  <input type="radio" name="p1" value="1"> Copenhague
  <input type="radio" name="p1" value="2"> Helsinki
  <input type="radio" name="p1" value="3"> Oslo
</p>
<p>Fecha de la Revolución francesa:
  <input type="radio" name="p2" value="1"> 1492
  <input type="radio" name="p2" value="2"> 1789
  <input type="radio" name="p2" value="3"> 1917
</p>
</fieldset>

```

Datos personales

Nombre:
Edad:

Cultura general

Capital de Noruega: ☐ Copenhague ☐ Helsinki ☐ Oslo
Fecha de la Revolución francesa: ☐ 1492 ☐ 1789 ☐ 1917

Sólo puede haber una etiqueta <legend>.

4.3 Accesibilidad: <label>

<label>

La etiqueta <label> permite asociar un control con un texto, de manera que mejore la accesibilidad de los formularios.

La asociación entre el control y la etiqueta <label> puede ser implícita o explícita.

4.3.1 Relación implícita

Se dice que la relación es implícita cuando el control se encuentra en el interior de la etiqueta.

Por ejemplo, en el caso de una casilla de verificación, la etiqueta <label> permite que la casilla se marque o desmarque haciendo clic en el texto, como se muestra en los ejemplos siguientes:

```
<input type="checkbox" name="casilla">Casilla 1
```



```
<label><input type="checkbox" name="casilla">Casilla 1</label>
```

En el caso de la cajas de texto `<input type="text">`, la etiqueta `<label>` permite que el cursor se sitúe en la caja de texto haciendo clic en el texto, como se muestra en los ejemplos siguientes:

Nombre: `<input type="text" name="casilla">`

```
<label>Nombre: <input type="text" name="casilla"></label>
```

4.3.2 Relación explícita: el atributo for

Se dice que la relación es explícita cuando la etiqueta `<label>` contiene el atributo `for`, que indica el control afectado (el control tiene entonces que tener establecido el atributo `id`).

Por ejemplo, en el caso de una casilla de verificación, la etiqueta `<label>` permite que la casilla se marque o desmarque haciendo clic en el texto, como se muestra en los ejemplos siguientes:

```
<input type="checkbox" name="casilla">Casilla 1
```

```
<input type="checkbox" name="casilla" id="casilla1">  
<label for="casilla1">Casilla 1</label>
```

En el caso de la cajas de texto `<input type="text">`, la etiqueta `<label>` permite que el cursor se sitúe en la caja de texto haciendo clic en el texto, como se muestra en los ejemplos siguientes:

Nombre: `<input type="text" name="casilla">`

```
<label for="text1">Nombre:</label>  
<input type="text" name="casilla" id="text1">
```

La forma explícita se necesita cuando el control y el texto asociado se encuentran en elementos distintos (por ejemplo en una tabla) y la etiqueta `<label>` no puede incluir ambos, como muestra el ejemplo siguiente:

```
<table>  
  <tr>  
    <th>Sexo</th>  
    <th></th>  
  </tr>  
  <tr>  
    <td><input type="radio" name="hm" value="h" id="h"></td>  
    <td><label for="h">Hombre</label></td>  
  </tr>  
  <tr>  
    <td><input type="radio" name="hm" value="m" id="m"></td>  
    <td><label for="m">Mujer</label></td>  
  </tr>  
</table>
```

Sexo	
<input type="radio"/>	Hombre
<input type="radio"/>	Mujer