

02-09 formularios HTML5

Nombre		Curso	
Apellidos		Fecha	

Introducción.....	3
→ Funcionamiento de los formularios. métodos GET y POST.....	3
→ Funcionamiento de los formularios.....	4
→ Métodos de paso de datos.....	6
paso de datos mediante GET.....	6
paso de parámetros mediante POST.....	6
Resumen final.....	7
→ Elemento form.....	8
→ Funcionamiento.....	8
→ Otros atributos de form.....	8
→ Controles de tipo input.....	10
→ Elemento input.....	10
→ Atributos comunes del elemento input.....	10
→ Cuadros de texto.....	13
→ Cuadro de contraseñas.....	14
→ Botones.....	14
→ Controles ocultos.....	16
→ Botones de radio.....	17
→ Casillas de verificación.....	18
→ Cuadro de selección de archivo.....	19
→ Cuadros numéricos.....	20
→ Cuadros de fecha.....	21
→ Cuadro de selección de hora.....	22
→ Otros cuadros de selección de fechas.....	22
→ Cuadros deslizantes.....	23
→ Otros controles.....	23

Introducción

Los formularios son un elemento presente en muchas páginas web cuya labor es la de recabar información del usuario. La función habitual de un formulario es enviar esa información a un servicio en la red para que la procese adecuadamente.

Por ejemplo podemos pedir al usuario su nombre y contraseña y enviar esa información a una página **PHP**, lenguaje que se interpreta directamente por el servidor que almacena esas páginas, el cual puede validar o no esos datos.

No siempre hay una tecnología del lado del servidor detrás, podemos también, a través de un formulario, indicar que el usuario elija el color de fondo de la página. La labor de recoger el color y modificar el fondo de la página la puede hacer el propio navegador a través del lenguaje **JavaScript**.

Los formularios son fundamentales para las páginas web actuales, ya que es casi imprescindible obtener información del usuario para adaptar los contenidos a sus necesidades e intereses.

En HTML 5 se han mejorado muchísimo las posibilidades de los formularios. Una vez más, el problema es que hay características que no todos los navegadores soportan, solo los más modernos.

→ Funcionamiento de los formularios. métodos GET y POST

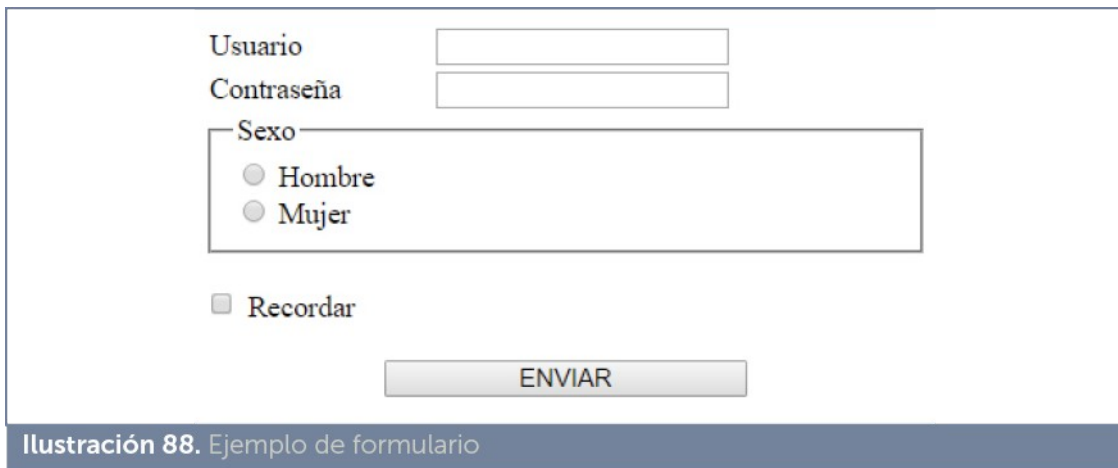


Ilustración 88. Ejemplo de formulario

Formulario de login con los siguientes campos:

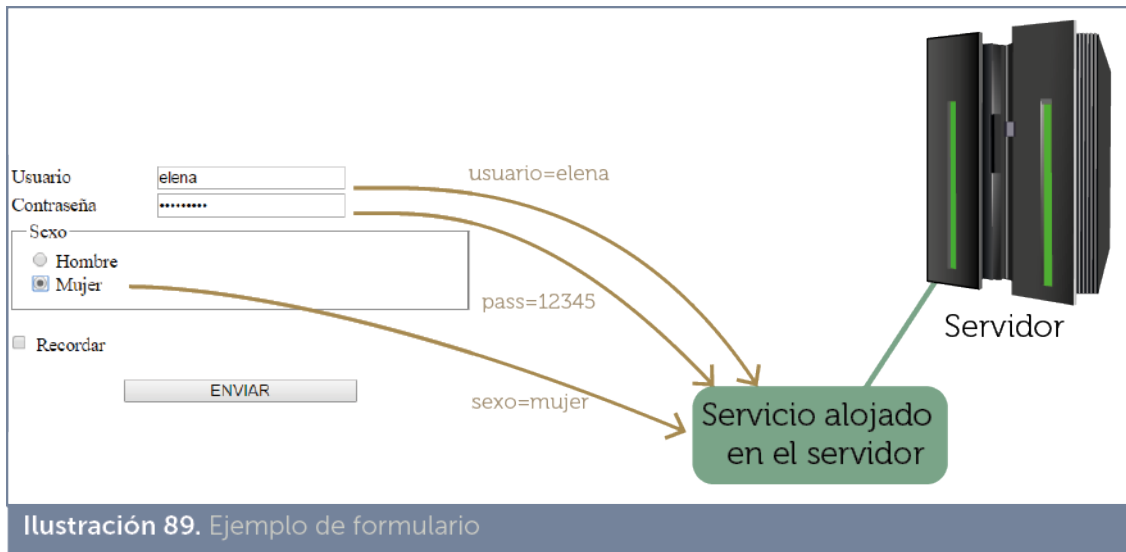
- Usuario:
- Contraseña:
- Sexo:
 - ☐ Hombre
 - ☐ Mujer
- ☐ Recordar
- Botón: ENVIAR

→ Funcionamiento de los formularios

Un formulario es un contenedor de controles que sirven para que el usuario introduzca datos. Un ejemplo de formulario se puede ver en la Ilustración 88.

Cada control posee un **nombre** al cual se le asigna un valor, el cual se corresponde con lo que el usuario introduce en el usuario.

Cuando se envían los datos del formulario (normalmente a través de un botón), se envían todos los nombres y valores de los controles del formulario. En el caso típico, un servicio de Internet recoge esta información y hace algo con ella.



Como se observa en la Ilustración 89, al enviar los datos del formulario, el servidor recibe pares de datos con cada valor asociado a cada nombre de control.

El servidor procesa estos datos, pero de forma opaca para el usuario. No se transmite el código que hace posible el

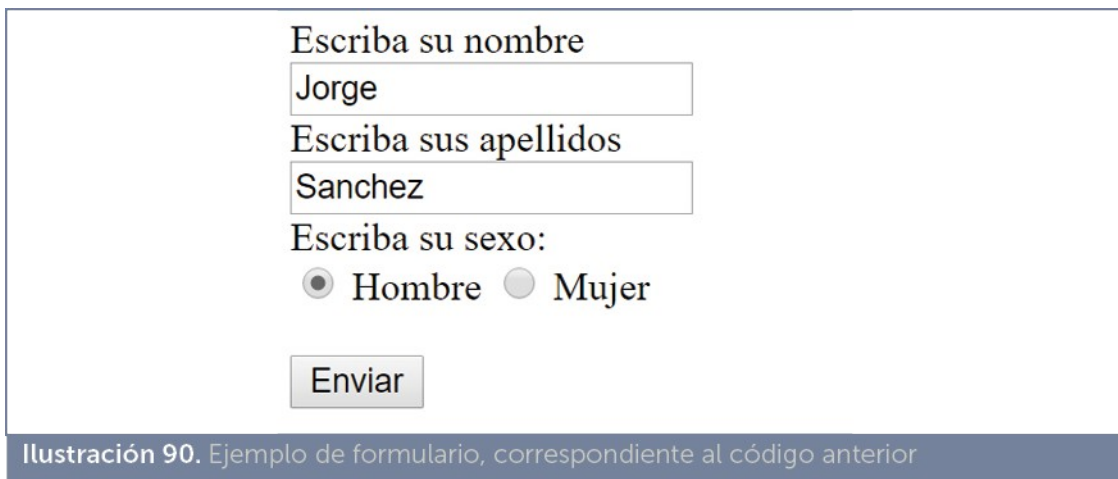


Ilustración 90. Ejemplo de formulario, correspondiente al código anterior

proceso, sino el resultado del mismo.

Este manual no explica como procesar los datos en el lado del servidor, para ello habría que aprender lenguajes como **PHP, ASP, JSP**,.... todos ellos relacionados con tecnologías del lado del servidor.

Al usuario le llegará una respuesta, relacionada con los datos que se han procesado, normalmente en forma de página web.

Supongamos que tenemos este formulario:

```
<form action="servicio.php">
```

```
<label for="nombre">Escriba su nombre</label><br>
```

```
<input type="text" id="nombre" name="nombre"><br>
```



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

```
<label for="apellidos">Escriba sus apellidos</label><br>
<input type="text" id="apellidos" name="apellidos"><br>
<label for="sexo">Escriba su sexo: </label><br>
<input type="radio" id="hombre" name="sexo" value="hombre">
<label for="hombre">Hombre</label>
<input type="radio" id="mujer" name="sexo" value="mujer">
<label for="mujer">Mujer</label><br><br>
<button>Enviar</button>
</form>
```

Como veremos, la etiqueta **form** es la encargada de indicar el destino de los datos. En este caso es la página PHP: *servicio.php*, que deberá estar alojada en el mismo directorio que la página del formulario.

Hay cuatro controles (distinguibles por su atributo **name**): *nombre*, *apellidos* y *sexo*.

Al pulsar enviar se envía el valor *Jorge* para el control llamado *nombre*, *Sánchez* para *apellidos* y *Hombre* para *sexo*.

→ Métodos de paso de datos

paso de datos mediante GET

En el ejemplo anterior, los datos se pasan mediante **GET**. GET es un comando del protocolo http que realiza una petición a un URL.

En este método los datos se pasan en la propia URL. Si el formulario se aloja en la dirección:

<http://jorgesanchez.net/practicas/formulario/> entonces cuando escribamos los datos de la Ilustración 90, se generará esta dirección en el navegador:

```
http://jorgesanchez.net/practicas/formulario/servicio.php?
nombre=Jorge&apellidos=S%C3%A1nchez&sexo=hombre
```

El apellido *Sánchez* se convierte en **S%C3%A1nchez** (debido a que las URL no pueden tener el carácter *á*). Dejando de lado la cuestión de la codificación, la forma de una URL que contiene datos procedentes de un formulario vía GET es:

```
http://urlpágina?var1=valor1&var2=valor2&.....
```

Es decir, se envían pares entre el nombre de la variable (en el formulario asignada a través del atributo **name**) y el valor que se le dio en el formulario. Cuando un control del formulario se queda sin valor, entonces el nombre del control se queda sin definir (el servidor no recibiría ningún dato con ese nombre).

La cuestión en el paso por GET es que los datos que se pasan aparecen en la propia URL; por lo que el paso de datos de un formulario mediante GET tiene estas connotaciones:

- La URL es visible, por lo que todo lo que el usuario ha escrito está a la vista en la URL. Y eso incluye las contraseñas y nombres de usuario que haya utilizado.
- Las páginas generadas con GET pueden añadirse como marcador o favorito a los navegadores y páginas de marcadores.

paso de parámetros mediante POST

El paso por POST funciona exactamente igual, se siguen pasando los pares nombre, valor. Pero los datos se envían dentro del paquete de datos y no por la URL.

Parece más seguro el paso de parámetros por POST, pero lo cierto es que los datos que viajan mediante el método POST, lo hacen en la cabecera del paquete y esta cabecera es visible por cualquier persona que esté “*escuchando*” en la misma red que el cliente que está enviando datos.

Aunque es más difícil obtener los datos enviados vía POST, es posible hacerlo.

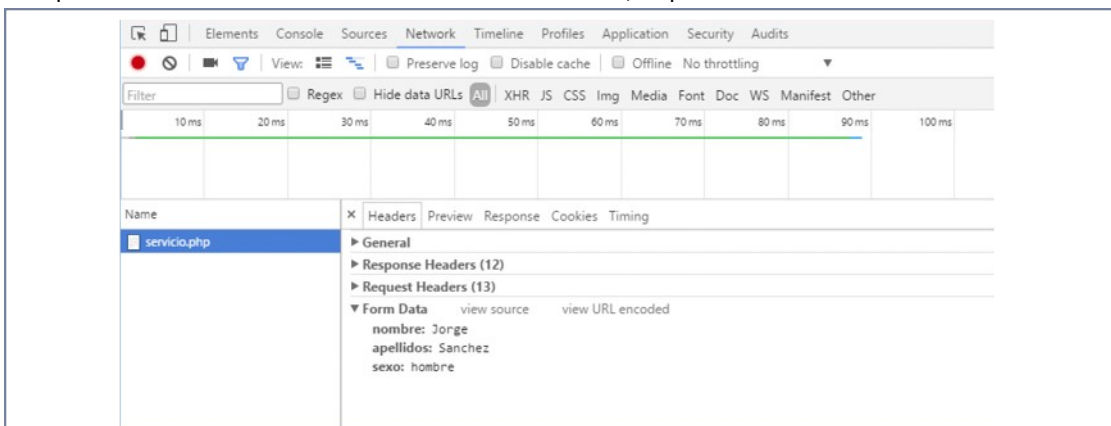


Ilustración 91. Panel de depuración de Google Chrome mostrando datos enviados usando el método POST



Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

Resumen final

GET tiene más mala prensa de POST, por decirlo de esta manera. Pero lo cierto es que el paso por POST no garantiza en absoluto la confidencialidad. GET tiene sus ventajas (podemos dejar guardada la página tras enviar los datos en los marcadores).

Al final, para el diseñador de la página web la realidad es que si creamos simplemente el formulario, deberemos saber qué exige el servicio que recibe los datos, si método GET o método POST. Hay que tener en cuenta que en un sitio web grande el **front-end** de la página (la parte visible para el usuario) lo puede realizar una persona distinta respecto a la que crea el **back-end** (la programación en el lado del servidor, invisible para el usuario).

→ Elemento form

→ Funcionamiento

Todo formulario HTML comienza con una etiqueta **form**, dentro de ella se colocan todos los controles del formulario. Un formulario por lo tanto es un elemento **form**, desde el punto de vista HTML, y dentro habrá diversos elementos que representarán a cada uno de los controles encargados de recabar y presentar los datos del usuario. La idea es por lo tanto esta:

```
<form action="página/servicio" method="GET o POST"> ... </form>
```

Como se observa, hay dos atributos que se encargan de indicar a dónde y de qué forma enviar los datos recogidos del formulario. Eso se hace a través de los atributos **action** y **method**, se explican a continuación:

- **action.** Se trata de la URL de la página o servicio al que se enviarán los datos para su procesamiento. Si no queremos que el formulario envíe datos a otra página (por ejemplo, porque simplemente los queremos manejar internamente con JavaScript), se suele poner `action="#"`.
- **method.** Puede ser GET o POST. Se relacionan con las dos formas de enviar datos del protocolo http. Se explica la diferencia al final de este capítulo.

→ Otros atributos de form

Atributo	Significado
accept-charset	<p>Codificación de caracteres que se utilizará para pasar los contenidos del formulario, por ejemplo <i>ISO-8859-1</i>.</p> <p>Por defecto se usa la codificación de la propia página web.</p>
autocomplete	<p>Puede tomar los valores on u off. Si se activa esta opción (con valor on), el navegador automáticamente completará los datos del formulario basándose en los valores para esos mismos controles que el usuario haya rellenado en otros formularios.</p> <p>Este atributo está disponible también para cada control (así podremos indicar el auto completado en unos controles y en otros no).</p> <p>No funciona en el navegador Opera.</p>
enctype	<p>Solo es válido para el método POST. Indica el formato en el que los datos del formulario son enviados. Nunca se suele usar. Posibles valores:</p> <ul style="list-style-type: none"> • application/x-www-form-urlencoded. Formato predeterminado para el envío de los datos. • multipart/form-data. Formato de dato binario, solo se usa si con el formulario se envía un archivo. • text/plain. Texto plano, compatible con antiguos programas de recepción de datos

Atributo	Significado
	de formulario (por ejemplo viejos programas CGI en el servidor).
novalidate	Los navegadores compatibles con HTML 5 tienen la capacidad de validar algunos datos (por ejemplo en los cuadros numéricos se valida que el usuario realmente ha escrito un número). Si usamos este atributo (que no tiene valor asociado), esas validaciones no se harán
target	Es el mismo atributo que en la etiqueta <i>a</i> . Si se usa con valor _blank , entonces la respuesta al formulario se da en otra ventana.

→ Controles de tipo input

→ Elemento input

La mayoría de controles de un formulario se inserta mediante la etiqueta `input`. Posee un atributo llamado **type** para indicar el tipo de control deseado, por ejemplo `input type="text"` se usa para añadir un cuadro de entrada de texto.

→ Atributos comunes del elemento input

Los atributos comunes de esta etiqueta (independientemente del tipo elegido) se explican a continuación:

Atributo	Significado
type	Indica el tipo de control. Sus posibles valores se explican en las siguientes secciones del manual
name	Atributo fundamental que da un nombre al control. Este nombre es el que se le pasa al servicio receptor de los datos del formulario; realmente se le pasa el nombre y el valor que el usuario ha introducido en el control.
size	Tamaño, en caracteres, que tendrá el cuadro (especialmente útil en los cuadros de texto y numéricos)
value	Permite dar un valor al elemento. Antes de que el usuario rellene datos en él, aparecerá el valor indicado con este atributo.
disabled	El control aparecerá deshabilitado para su edición.
autocomplete	Permite que el navegador rellene automáticamente el contenido del control en base a la información que posee del usuario. Los valores posibles son <code>on</code> u <code>off</code> .
autofocus	Hace que elemento obtenga el foco (el foco hace que el control del formulario que lo tenga sea el que recibe las pulsaciones de teclado del usuario) en cuanto cargue la página.
disabled	El control estará deshabilitado
form	Recibe el identificador del formulario al que pertenece el control. Se usa cuando el elemento está fuera de la etiqueta <code>form</code> . Internet Explorer no es compatible con este atributo. Es un atributo de

Atributo	Significado
	HTML 5
formnovalidated	Hace que el control no sea validado cuando se envíen los datos
maxlength	Permite indicar el máximo número de caracteres que se admitirán al rellenar el control
	Este atributo es el común a todos los elementos HTML. En el caso de los formularios su uso no era habitual ya que necesitan del uso de name que ya se puede entender que identifica a cada control.
	Sin embargo se aconseja usar siempre id en los controles por estas razones:
id	<ul style="list-style-type: none"> El atributo name ya hace bastantes años que se quería eliminar. Aunque esta no es razón para ello ya que, por ahora, el atributo name es la forma de pasar los datos de un control de formulario al servicio que los recibe La nueva etiqueta label (introducida en HTML 5) que sirve como texto de acompañamiento de los controles de un formulario, requiere que el control al que etiqueta esté identificado. La manera más habitual para acceder desde JavaScript a un control del formulario es a través de su identificador. Como es habitual usar JavaScript cuando se usan los formularios, es un excelente hábito indicar siempre valor para el atributo id.
readonly	Indica que el control es de <i>solo lectura</i> ; es decir, no se podrá escribir en él.
required	<p>Atributo de HTML 5 que hace que el elemento sea de obligado rellenado; es decir que no se puede dejar vacío, el usuario tendrá que darle algún valor.</p> <p>No funciona en Safari no en versiones de Internet Explorer anteriores a la 9</p> <p>Permite colocar una expresión regular en un cuadro de texto. De esa forma no se admitirán los valores que no cumplan esa expresión (salvo que indiquemos novalidate)</p> <p>Ejemplo (cuadro de texto que solo acepta introducir 5 letras mayúsculas y tres números):</p> <pre><form action="servicio.php" method="get"> <label for="texto"> Escribe el nº de serie (5 letras y tres números) </label> <input type="text" pattern="[A-Z] {5}[0-9] {3}" id="gustos" name="texto"/> <input type="submit" value="enviar">
 </form></pre>
placeholder	Un placeholder es un texto que ayuda a rellenar un cuadro de un formulario (está especialmente pensado para los cuadros de texto) colocando un texto inicial en el cuadro que se va en cuanto el cuadro obtiene el foco del usuario (el foco se obtiene, por ejemplo, haciendo clic en el control de

Atributo

Significado

texto). Ejemplo:

```
<form action="servicio.php"
method="get">
<label for="texto">
Escribe el nº de serie
</label>
<input type="text" pattern="[A-Z]{5}[0-9]{3}" id="texto" name="texto"
placeholder="5 letras y tres números">

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

Escribe el nº de serie

Ejemplos de patterns

Solo letras Mayusculas pattern="[A-Z]*"

Solo letras pattern="[A-Za-z]*"

Solo números pattern="[0-9]*"

→ Cuadros de texto

Los cuadros de texto usan el atributo **type** con valor **text**. Permiten recoger información escrita por el usuario. Su sintaxis básica es:

```
<input type="text" name="nombre" id="identificador">
```

Además podemos usar los atributos comunes a los elementos **input** comentados anteriormente.

El resultado de un código tipo `input type="text"` es un cuadro en el que el usuario puede introducir texto. Ejemplo:

```
<form action="servicio.php" method="get">  
  <label for="nombre">Escriba su nombre</label>  
  
  <input type="text" name="nombre" id="nombre"  
    size="8" maxlength="20">  
</form>
```

Ese código da como resultado:

Escriba su nombre

Ilustración 92. Cuadro de texto

→ Cuadro de contraseñas

Funcionan como los cuadros de texto, solo que el texto que se introduce se oculta, mostrando solo puntos o asteriscos. Se usan, lógicamente, para introducir contraseñas. La sintaxis es:

```
<input type="password" name="nombre" id="identificador">
```

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 aun así podríamos averiguarla. Por ello lo ideal es pasar cifrada la contraseña a través, por ejemplo, del protocolo **https**.

Escriba su nombre

Ilustración 93. Cuadro de contraseñas

→ Botones

Los botones son controles del formulario en los que no se puede escribir, sino que, simplemente, reciben un click de ratón o un golpe de dedo en un dispositivo táctil. Esa pulsación puede dar lugar a diversos efectos (la mayoría con ayuda de JavaScript). El más común de ellos: enviar los datos del formulario.

Hoy en día los botones se aconsejan utilizarlos mediante el elemento **button**, que se explica más adelante. No obstante muchos diseñadores siguen utilizando esta forma (que por otro lado es más compatible con navegadores antiguos), por ello se explica en este manual

botón de envío

Sirve para llevar a cabo la comunicación entre el formulario y la página que recoge sus datos. La sintaxis básica es:

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

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**).

botón alternativo de tipo submit

Siempre ha sido posible colocar dos botones de tipo submit en el mismo formulario. Pero la cuestión es ¿para qué? ya que ambos harían lo mismo: enviar los datos al servicio indicado en el atributo **action** del elemento **form**.

Con la aparición de HTML 5 se ha dado un sentido más lógico a esto ya que podemos indicar servicios de destino distintos para cada botón **submit**. Eso se consigue con ayuda de estos nuevos atributos:

atributo	significado
	Atributo incorporado en HTML 5 (las versiones de Internet Explorer inferiores a la 10 no le soportan), que asignado a un botón de tipo submit o image , permite indicar el servicio al que se enviarán los datos del formulario.
formaction	Normalmente un botón <i>submit</i> envía los datos al servicio indicado por el atributo action del elemento form , esto permite un destino alternativo, de modo que podríamos tener dos botones cada uno de ellos indicando un destino distinto para los datos.
formmethod	Con las mismas características que el anterior, en este caso permite indicar el método de paso (GET o POST) de los datos
	Con las mismas premisas que los dos atributos anteriores, indica la forma de codificación de los datos (como el atributo type del elemento form), puede ser:
formenctype	<ul style="list-style-type: none">• application/x-www-form-urlencoded• multipart/form-data• text/plain
formnovalidate	Los datos que se envían no se validarán (como el atributo validate del elemento form). No funciona en Safari (además de en las versiones anteriores a la 10 de Internet Explorer).
formtarget	El destino de los datos se mostrarán en una ventana aparte si se le da el valor _blank a este atributo.

botón de restablecer

La sintaxis de este botón es:

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

Este tipo de botones lo que hacen es anular los datos que se han introducidos en los controles del formulario. Deja todos los controles en su estado inicial.

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, sino que se utiliza normalmente para capturar su pulsación y lanzar una acción desde el lenguaje JavaScript.

botón de imagen

Es equivalente al anterior, pero permite colocar un botón con una imagen en lugar de un texto. Este tipo de botón permite usar atributos que controlan las propiedades de la imagen en el botón:

- **alt**. Texto alternativo para la imagen
- **src**. URL de la imagen
- **width**. Anchura de la imagen
- **height**. Altura de la imagen

Hoy en día, prácticamente, no se usan estos botones.

→ **Controles ocultos**

Son controles que no se muestran al usuario y que se usan con la finalidad de hacer trucos. Por ejemplo para poder enviar datos de forma invisible a los servicios y páginas receptoras de los datos del formulario y así el usuario no vea esos datos. Otro es para poder enviar datos entre diferentes páginas web de forma que no se muestren dichos datos por pantalla, ya que sería muy poco estético.

Los controles de este tipo usan el valor **hidden** en el atributo **type**.

→ Botones de radio

Se trata de un control fácil de usar que permite elegir una de entre varias opciones. Todas las opciones deben de tener el mismo nombre y solo cambia el valor (y el identificador) de las mismas.

La razón de tener todos los controles el mismo valor es porque el servicio que recibe los datos, los debe recibir con el mismo nombre (debe almacenar los valores en el mismo, digamos *sitio*) ya que se refieren a la misma cosa. Cambian sus valores en función de la selección que hagamos.

Ejemplo:

```
<form action="servicio.php" method="get">
<p>Sexo:</p>
<input type="radio" name="sexo" id="hombre"

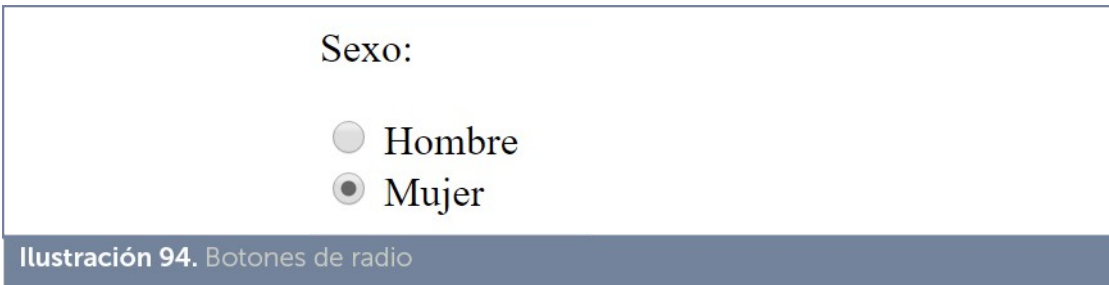
value="h">

<label for="hombre">Hombre</label><br>
<input type="radio" name="sexo" id="mujer"

value="m" checked >

<label for="mujer">Mujer</label> <br>
</form>
```

El resultado es:



Sexo:

☐ Hombre

☒ Mujer

Ilustración 94. Botones de radio

En el ejemplo, los botones de radio tienen el mismo nombre (*sexo*), bajo ese nombre se enviarán los datos a la página *servicio.php* (indicada en el apartado *action*), los valores que se enviarán serán *m* si se elige *mujer* y *h* si se elige *hombre*. El atributo **checked** hace que el botón en el que se usa (el de la mujer), aparezca chequeado por defecto. Además los dos elementos *label* se relacionan con el botón correspondiente, de ese modo haciendo clic en el texto se activa el botón relacionado (prueba de que hemos asociado bien la etiqueta *label* al *input* correspondiente).

→ Casillas de verificación

Se usan igual que los botones de radio, pero en este caso están pensadas para ser independientes unas de otras. Es decir, no se pone el mismo nombre, como ocurría con los botones de radio, a dos casillas diferentes. Son por tanto valores del tipo activar/desactivar

La diferencia es que el tipo (**type**) usa el valor **checkbox** en lugar de **radio**. El atributo **checked** de los botones de radio se puede usar en las casillas de verificación con la misma finalidad: para que inicialmente la casilla aparezca pulsada. Ejemplo:

```
<form action="servicio.php" method="get">
  <p>Descuentos:</p>

  <label for="numerosa">Familia Numerosa</label>

  <input type="checkbox" name="numerosa" id="numerosa"

    value="numerosa"><br>
  <label for="pension">Pensionista</label>

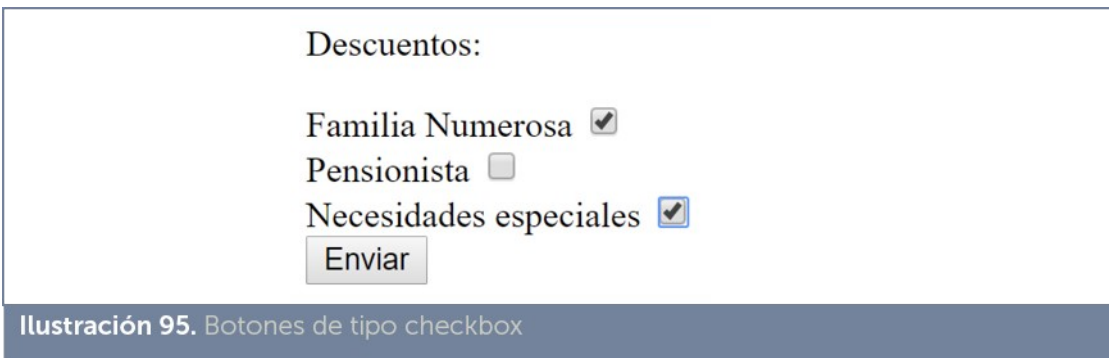
  <input type="checkbox" name="pension" id="pension"

    value="pension"><br>
  <label for="ne">Necesidades especiales</label>

  <input type="checkbox" name="ne" id="ne"

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

El resultado:



Descuentos:

Familia Numerosa ☒

Pensionista ☐

Necesidades especiales ☒

Enviar

Ilustración 95. Botones de tipo checkbox

En este caso se ha diseñado un formulario completo, incluido el botón de enviar.


→ 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.

Ejemplo:

```
<form action="servicio.php" method="get">  
  Elija el archivo  
  <input type="file" name="archivo">  
  <input type="submit" value="enviar"><br>  
</form>
```

El resultado es:



Elija el archivo Seleccionar archivo Ningún archivo seleccionado enviar

Ilustración 96. Botón de selección de archivo

Al pulsar *seleccionar archivo* aparece un cuadro de selección de archivos, cuando elijamos el archivo su ruta será lo que se envíe con el formulario.

atributos relacionados con los cuadros de archivo

atributo	significado
----------	-------------

accept	Permite indicar el tipo MIME de archivos que aceptará el servidor. Por ejemplo image/* indicará que se acepta cualquier formato de imagen. Otras posibilidades son audio/* y video/* o indicar un tipo MIME completo como image/jpeg .
---------------	--

multiple	Si aparece, permite especificar más de un archivo
-----------------	---

→ Cuadros numéricos

Aparecen con *type="number"*. Sirven para introducir números (decimales o no). Los navegadores que reconocen este tipo de control presentan un cuadro de texto con botones para subir y bajar el valor del número (el incremento o decremento que hacen depende del atributo **step**) y además no permiten enviar los datos (salvo que se use el atributo **novalidate**) si se intentan introducir datos no numéricos.

Hay una serie de atributos reservados a este tipo de cuadros:

atributo significado

max Establece el máximo valor que se puede escribir en el cuadro

min Establece el mínimo valor que se puede escribir en el cuadro

step Valor en el que se puede incrementar el número. Por defecto vale **1**, lo que impediría introducir valores decimales. Para decimales por ejemplo habría que indicar un step de **0.01**

¿Cuál es su edad?

Ilustración 97. Cuadro numérico

→ Cuadros de fecha

Se usa con *type="date"*. Permite seleccionar una fecha. Los navegadores que reconocen este tipo de control permiten pulsar en un botón que invoca a un calendario para elegir cómodamente la fecha. Ejemplo:

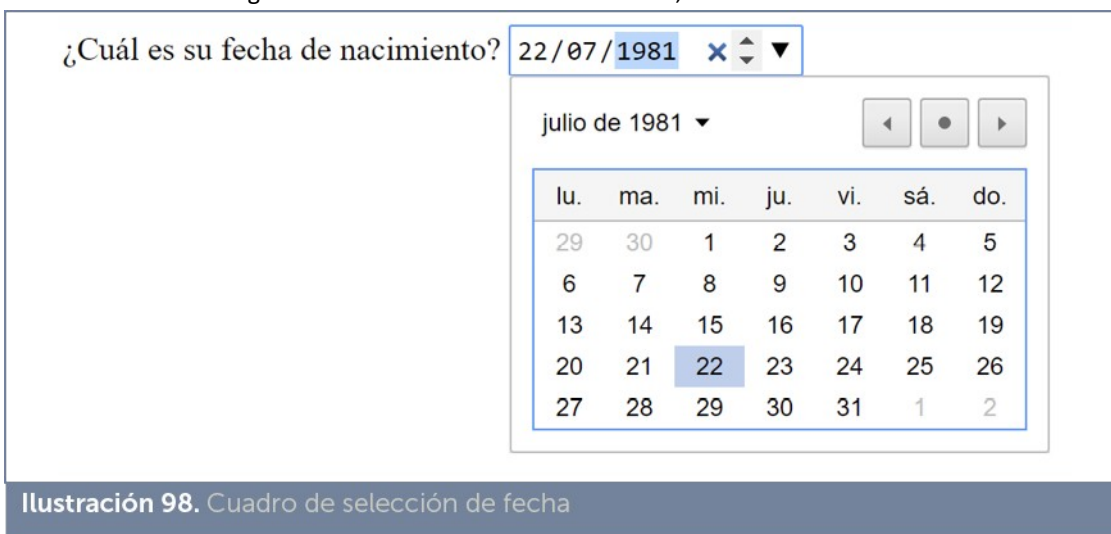
```
<form action="recogida1.php" method="get">
```

```
<label for="fecha_nac">Fecha de nacimiento</label>
```

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

```
</form>
```

El resultado en el navegador **Chrome** del cuadro de calendario, es este:



¿Cuál es su fecha de nacimiento? 22/07/1981

julio de 1981

lu.	ma.	mi.	ju.	vi.	sá.	do.
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

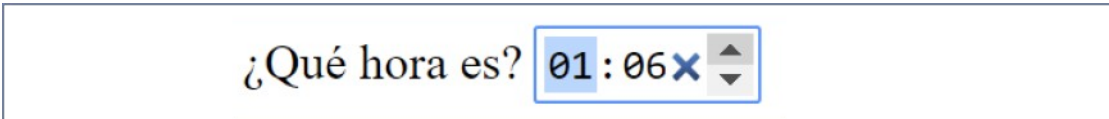
Ilustración 98. Cuadro de selección de fecha

Es pues, un control muy visual y práctico.

Los cuadros de tipo date tienen los atributos **min**, **max** y **step** vistos en los cuadros numéricos.

→ Cuadro de selección de hora

Se usa con **input type="time"**. Permite seleccionar una hora, minutos y segundos



¿Qué hora es? 01:06 x

Ilustración 99. Cuadro de selección de hora

→ Otros cuadros de selección de fechas

Aunque no funcionan en la mayoría de navegadores, son parte ya de html

- **input type="datetime"**. Permite seleccionar fechas con hora
- **input type="month"**. Selecciona meses y años
- **input type="week"**. Selecciona el número de semana y el año de la misma.

→ Cuadros deslizantes

Es el control llamado en inglés *slider*, aunque se usa con el atributo *type="range"*. Presenta un control para elegir datos entre un rango. Los atributos **max** y **min** establecen el rango máximo y mínimo del control. El atributo **step** indica cuánto se mueve el control (si de uno en uno, de dos en dos,...). Ejemplo:

```
<form action="recogida3.php" method="get">  
  
  <label for="rango">Seleccione su edad</label>  
  
  <input type="range" name="range" id="range"  
  
    value="25" min="18" max="65" step="1">  
  
</form>
```

El resultado:



Ilustración 100. Cuadro deslizante de selección de rango

→ Otros controles

- **input type="email"**. Permite indicar direcciones de correo electrónico
- **input type="url"**. Permite indicar direcciones URL.
- **input type="search"**. Presenta un cuadro de texto pensado para hacer búsquedas.
- **input type="tel"**. Permite introducir números de teléfono.
- **input type="color"**. Presenta un control de selección de colores. El color se toma en formato #xxxxxx donde cada x es una cifra hexadecimal. Es decir, el color se obtiene en el formato habitual de colores de HTML.

Práctica Funcionamiento de los formularios. métodos GET y POST e input

1. Escribe en un documento htm con el titulo `<h1> formularios </h1>` y con tus palabras en que consiste el Funcionamiento de los formularios
2. Escribe en un documento htm con el titulo `<h1> GET/POST </h1>` y con tus palabras en que consiste el paso de datos mediante GET y post y las ventajas de cada uno
3. Escribe en un documento htm con el titulo `<h1> Atributos </h1>` una lista de definición que explique cada atributo.
 - accept-charset
 - autocomplete
 - enctype
 - novalidate
 - target
4. Escribe en un documento htm con el caption `<h1> Atributos text</h1>` una tabla de ordenada que explique cada atributo del en 3/4 palabras

type	form
name	formnovalidated
size	maxlength
value	id
disabled	readonly
autocomplete	required
autofocus	pattern
disabled	placeholder

5. Escribe en un documento htm con el titulo <h1> formulario </h1> . Crea un form con get e introduce los siguientes controles. El id debe tener el mismo nombre que el label que contenga
 - Label Nombre. input Cuadros de texto . pattern Solo letras . requerido. autofocus . i
 - Label apellidos . input Cuadros de texto. Pattern letras y espacios. requerido
 - Label edad . input Cuadros de texto. Pattern numero. requerido
 - Label Contraseña. Cuadro de contraseñas tamaño 5
 - Control oculto con el valor ="edad"
 - Inserta cada uno de los tipos de los botones
6. Escribe en un documento htm con el titulo <h1> Atributos button</h1> una lista de ordenada que explique cada atributo.
 - Formaction
 - formmethod
 - formenctype
 - formnovalidate
 - formtarget
7. Crea un formulario get que incluya los siguientes componentes
 - Un input text
 - un componente con 4 Botones de radio
 - Dos Casillas de verificación
 - Cuadro de selección de archivo
8. Crea un formulario como este dentro de una tabla con este estilo
style="width: 80%; margin-left: auto; margin-right: auto; border: red 1px solid;"

Escriba su nombre

Escriba sus apellidos

Escriba su sexo:

☒ Hombre ☐ Mujer

9. Crea un formulario get para realizar una reserva que incluya los siguientes componentes
 - Un input text con los datos de la personas
 - Cuadros numéricos para indicar el numero de personas
 - input type tel para introducir números de teléfono de los usuarios
 - input email para indicar direcciones de correo electrónico
 - Cuadros de fecha
 - Cuadro de selección de hora
10. Crea un formulario get para realizar un ajuste de color que incluya los siguientes componentes
 - Un input text con la temática del formulario
 - **un input type color para tomar el color**
 - dos Cuadros deslizantes para el contraste y el brillo

11. Crea un formulario Post para realizar una submisión de una web a un servicio de búsqueda. debes incluirlo en una tabla

- **Un input text con la temática del formulario**
- **input url para** indicar la dirección de la web a enviar .
- **input tipo search**". Presenta un cuadro de texto pensado para hacer búsquedas.
- **input tipo file** para subir un fichero

Escoge el sistema de envío		
Buscar archivo <input type="text" value="Web a Buscar"/>	Subir fichero <input type="button" value="Seleccionar archivo"/> Ninguno archivo selec.	Escoger web <input type="text" value="Escribe una url correcta"/>
<input type="button" value="Enviar"/>		

para que quede como el documento debes incluir estos estilos en una tabla

para table

style="width: 80%; margin-left: auto; margin-right: auto; border: red 1px solid;"

para td

style="border: COLOR 1px solid; text-align: center;"

COLOR sera red o blue