

Formularios HTML

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.

[1] Funcionamiento de los formularios.

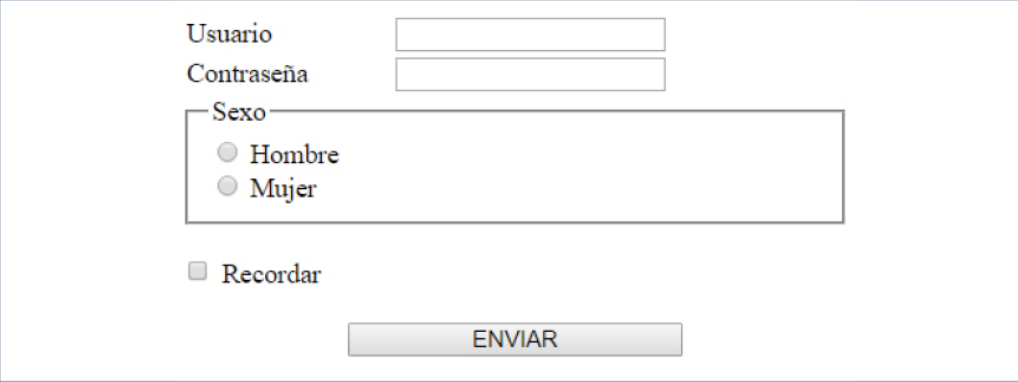


Ilustración 88. Ejemplo de formulario

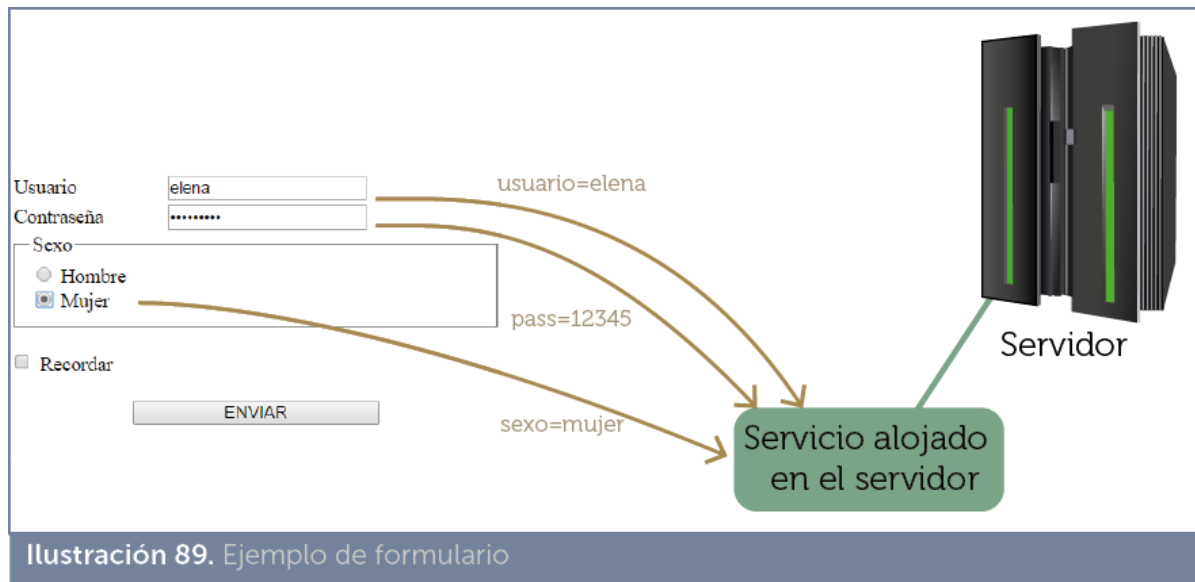
El formulario se encuentra dentro de un recuadro con un borde azul. Contiene los siguientes elementos:

- Un campo de texto etiquetado "Usuario".
- Un campo de texto etiquetado "Contraseña".
- Un grupo de selección etiquetado "Sexo" que contiene dos opciones: "Hombre" y "Mujer", cada una con un botón de radio.
- Un checkbox etiquetado "Recordar".
- Un botón rectangular etiquetado "ENVIAR".

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 proceso, sino el resultado del mismo.

Este manual no explica cómo 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:

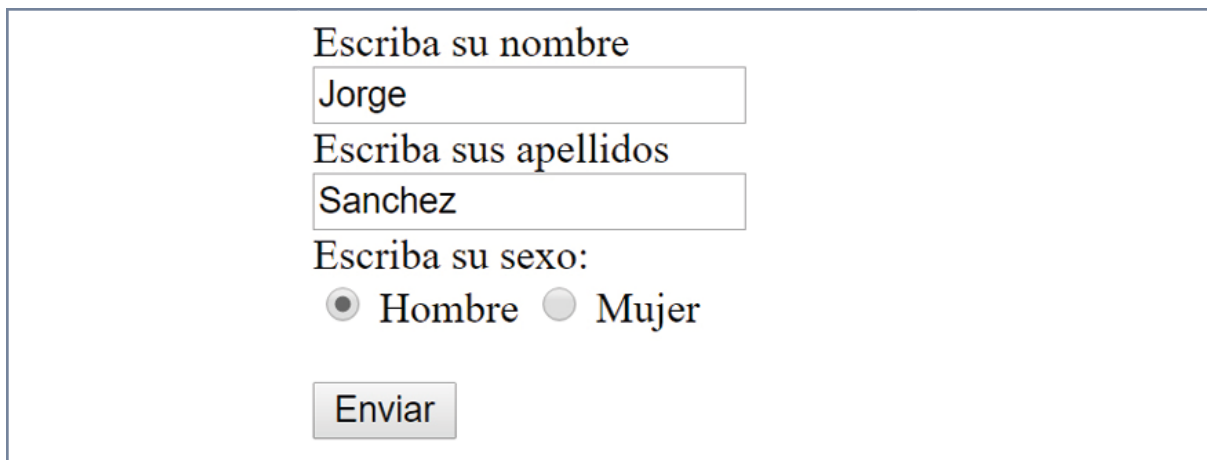
```

<form action="servicio.php">
  <label for="nombre">Escriba su nombre</label><br>
  <input type="text" id="nombre" name="nombre"><br>
  <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*.



The screenshot shows a web form with the following elements:

- A label "Escriba su nombre" followed by a text input field containing the value "Jorge".
- A label "Escriba sus apellidos" followed by a text input field containing the value "Sanchez".
- A label "Escriba su sexo:" followed by two radio buttons. The first radio button is selected and labeled "Hombre". The second radio button is unselected and labeled "Mujer".
- A button labeled "Enviar" at the bottom.

Ilustración 90. Ejemplo de formulario, correspondiente al código anterior

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

[2] Elemento form

[2.1] 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.

[2.2] Otros atributos de form

atributo	significado
autocomplete	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. Este atributo está disponible también para cada control (así podremos indicar el auto completado en unos controles y en otros no).
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

[3] Controles de tipo input y botones

[3.1] 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.

[3.2] 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.
autocomplete	Permite que el navegador rellene automáticamente el contenido del control en base a la información que posee del usuario. Los valores posibles son on u off.
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
maxlength	Permite indicar el máximo número de caracteres que se admitirán al rellenar el control

id	<p>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.</p> <p>Sin embargo se aconseja usar siempre id en los controles por estas razones:</p> <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	<p>Indica que el control es de <i>solo lectura</i>; es decir, no se podrá escribir en él.</p>
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>
pattern	<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 n° de serie (5 letras y tres números) </label> <input type="text" pattern="[A-Z]{5}[0-9]{3}" id="texto" name="texto"/> </pre>

	<pre><input type="submit" value="enviar">
 </form></pre>
placeholder	<p>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 texto). Ejemplo:</p> <pre><form action="servicio.php" method="get"> <label for="texto"> Escribe el n° 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">
 </form></pre> <p>El resultado es:</p> <p>Escribe el n° de serie <input type="text" value="5 letras y tres números"/> <input type="submit" value="enviar"/></p>

[3.3] 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:

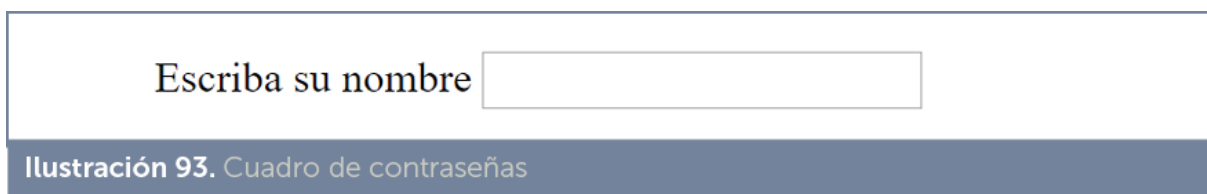
A screenshot of a web browser window showing a simple form. The form has a single text input field. To the left of the input field is the text "Escriba su nombre". The input field is empty and has a light gray border. Below the input field, there is a dark blue horizontal bar containing the text "Ilustración 92. Cuadro de texto" in white.

[3.4] 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**.

A screenshot of a web browser window showing a simple form. The form has a single password input field. To the left of the input field is the text "Escriba su nombre". The input field is empty and has a light gray border. Below the input field, there is a dark blue horizontal bar containing the text "Ilustración 93. Cuadro de contraseñas" in white.

[3.5] 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 aconseja 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 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 borrar los datos que se han introducidos en los controles del formulario. Deja todos los controles en su estado inicial.

botón genérico

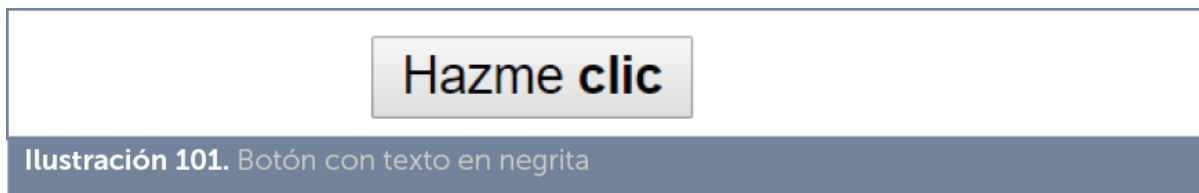
Un botón genérico se marca indicando **type="button"** en la etiqueta **type**. En los formularios no se usa para enviar o configurar la información, sino que se utiliza normalmente para capturar su pulsación y lanzar una acción desde el lenguaje JavaScript.

[3.6] Botones modernos

La recomendación de HTML 5 para los botones es utilizar el elemento **button**. La diferencia práctica está en el funcionamiento, esta etiqueta tiene apertura y cierre y el texto que aparece en el botón está contenido por elemento **button**; de esta forma se permite colocar código HTML en el texto del botón. Ejemplo:

```
<button>Hazme <strong>clic</strong></button>
```

Mostraría el botón:



Con la palabra *clic* en negrita. Podríamos perfectamente colocar imágenes metiendo un elemento **img** dentro del botón.

La única pega es que es una etiqueta más moderna y que no es soportada por algunos navegadores antiguos.

[3.6.1] atributos de la etiqueta button

Son los mismos que los de los botones de tipo input, salvo por algunos detalles (se recomienda ver los atributos del elemento input y de los botones de tipo submit):

atributo	significado
name	Nombre del botón
id	Identificador del botón
value	Permite dar un valor al botón
type	Indica el tipo de botón, puede ser: <ul style="list-style-type: none">■ submit. Envío de datos (valor por defecto)■ reset. Borrado de la información introducida en el formulario■ button. Botón genérico.
disabled	Deshabilita el botón

[3.6] 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**.

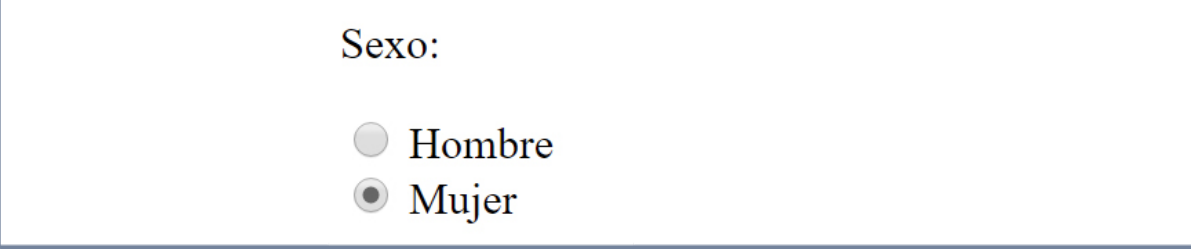
[3.7] Botones de radio(radio button)

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 (**name**) y solo cambia el valor (y el identificador) de las mismas.

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*, los valores que se enviarán serán *m* si se elige *mujer* y *h* si se elige *hombre*. El atributo **checked** hace que el radio button en el que se usa (en este caso 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).

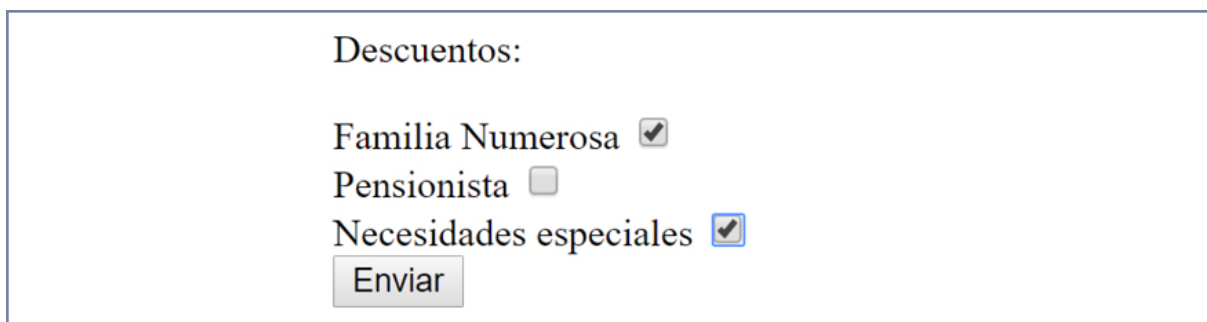
[3.8] Casillas de verificación(checkbox)

Se usan igual que los radio button, pero en este caso están pensadas para ser independientes unas de otras. Es decir, no se utiliza el mismo **atributo name**, como ocurría con los botones de radio. Son por tanto valores que solo pueden encontrarse en 2 estados **activado/desactivado**

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 que en los radio button(para que inicialmente la casilla aparezca seleccionada). 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.

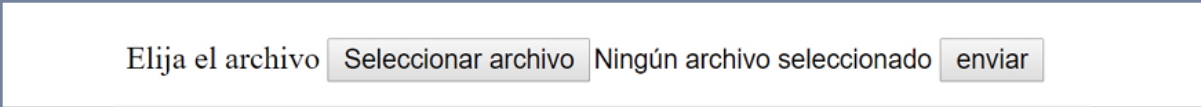
[3.9] Cuadro de selección de archivo

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>
```

Resultado:



Elija el archivo Ningún archivo seleccionado

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 cómo image/jpeg .
multiple	Si aparece, permite especificar más de un archivo

[3.10] 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
<code>max</code>	Establece el máximo valor que se puede escribir en el cuadro
<code>min</code>	Establece el mínimo valor que se puede escribir en el cuadro
<code>step</code>	Valor en el que se puede incrementar el número. Por defecto vale <code>1</code> , lo que impediría introducir valores decimales. Para decimales por ejemplo habría que indicar un step de <code>0.01</code>

¿Cuál es su edad?

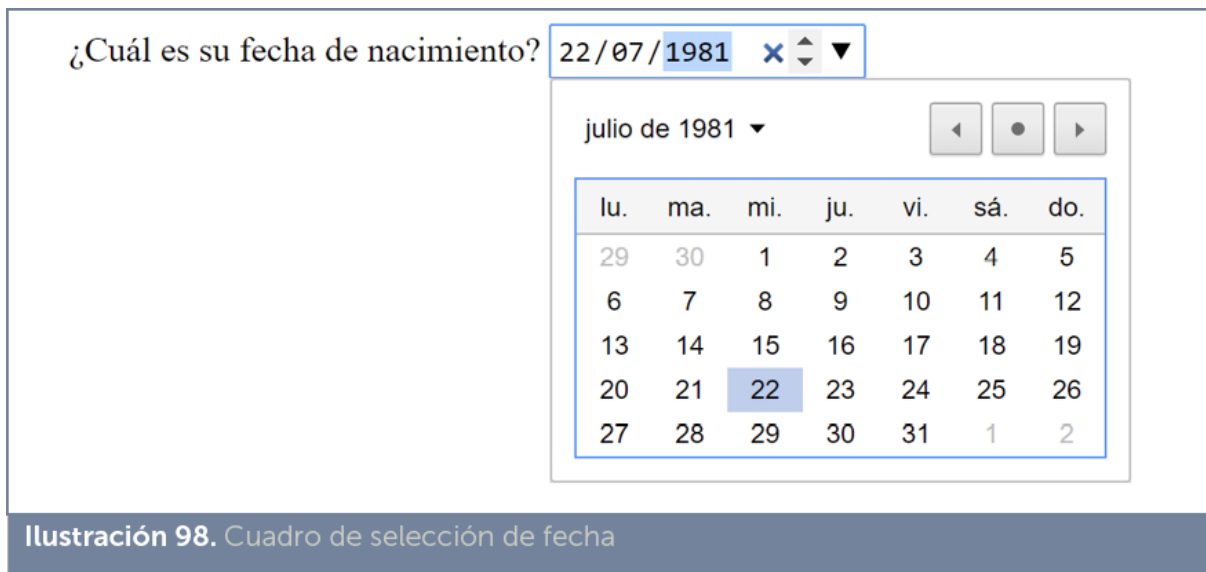
Ilustración 97. Cuadro numérico

[3.11] 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:

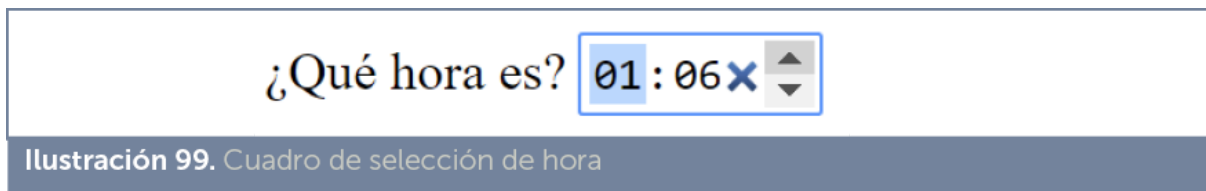


Es pues, un control muy visual y práctico.

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

[3.12] Cuadro de selección de hora

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



[3.13] Otros cuadros de selección de fechas

Aunque en navegadores antiguos no funcionan, son parte ya de html

- **input type="datetime-local"**. 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.

[3.14] 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

[3.15] Otros controles

- **input type="email"**. Permite indicar direcciones de correo electrónico
- **input type="url"**. Permite indicar direcciones URL.
- **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.

[4] Cuadros combinados

Un cuadro combinado permite el uso de una lista de opciones en la que se puede elegir la deseada. 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.

Ejemplo:

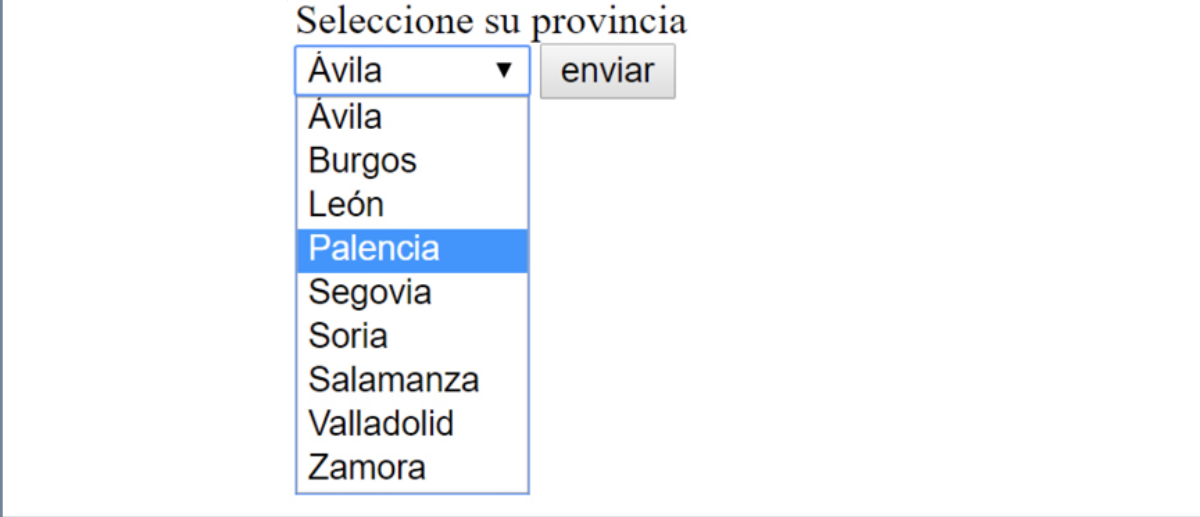

```

<form action="servicio.php" method="get">
  <label for="provincia" id="provincia">
    Seleccione su provincia
  </label><br>
  <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>

```

Resultado:



Seleccione su provincia

Ávila ▼ enviar

Ávila

Burgos

León

Palencia

Segovia

Soria

Salamanca

Valladolid

Zamora

Ilustración 102. Cuadro combinado

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

Ejemplo:

```

<form action="servicio.php" method="get">
  <label for="provincia" id="provincia">
    Seleccione su provincia
  </label><br>
  <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>

```

Da como resultado:

Seleccione su provincia

Santander ▼ enviar

Cantabria

Santander

Castilla y León

Ávila

Burgos

León

Palencia

Segovia

Soria

Salamanca

Valladolid

Zamora

Ilustración 103. Cuadro combinado y agrupado

[4.1] Atributos del elemento select

atributo	significado
name	Nombre del cuadro combinado
id	Identificador del cuadro combinado
autofocus	HTML5. El control adquiere el foco en cuanto carga la página
disabled	El cuadro aparece deshabilitado
multiple	El cuadro se convierte en un cuadro de selección de múltiples valores. Es una opción muy importante ya que este tipo de controles a veces es imprescindible
required	Es obligatorio al menos elegir una opción del cuadro.
size	Número de opciones visibles a la vez. Si elegimos 5, para ver la sexta habrá que usar la barra de desplazamiento del control.

[4.2] Atributos del elemento option

atributo	significado
disabled	La opción aparece deshabilitada en el cuadro combinado
selected	Opción seleccionada por defecto
value	Valor que se envía al servicio que recibe los datos del formulario cuando se elige esta opción

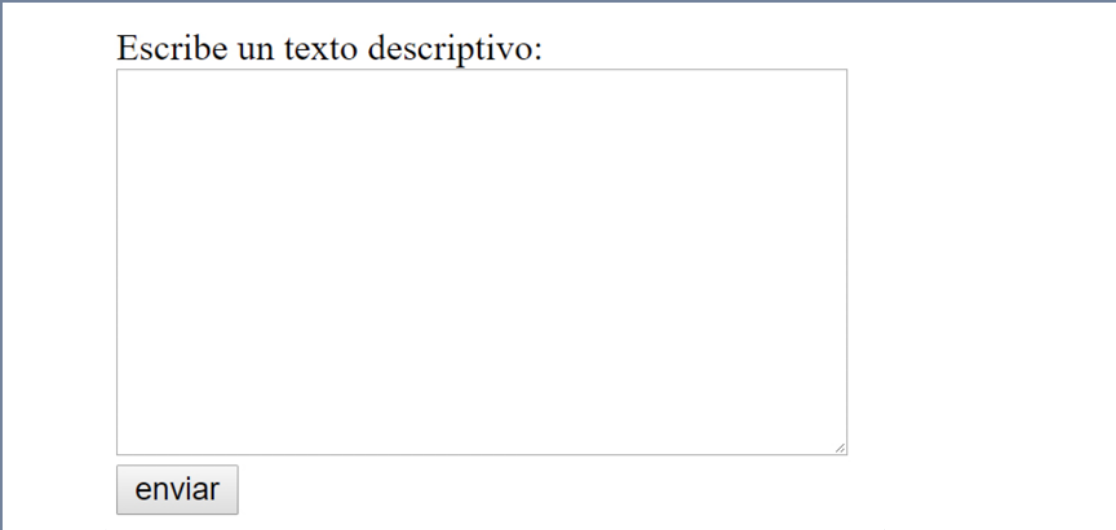
[5] Control de texto multilinea

La etiqueta **textarea** permite colocar un cuadro de texto de varias líneas para que el usuario pueda 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"**).

Ejemplo:

```
<form action="servicio.php" method="get">  
  Escribe un texto descriptivo: <br>  
  <textarea rows="10" cols="40" name="texto"></textarea><br>  
  <input type="submit" value="enviar"><br>  
</form>
```

Resultado:



The screenshot shows a web browser window with a form. The form contains the text "Escribe un texto descriptivo:" followed by a large, empty text area. Below the text area is a button labeled "enviar". The entire form is enclosed in a light blue border.

Ilustración 104. Cuadro multilinea

Entre la etiqueta **textarea** se puede colocar texto que aparecerá inicialmente dentro del cuadro.

[5.1] Atributos del elemento textarea

atributo	significado
name	Nombre del control

id	Identificador del control
value	Especifica un contenido para el control
autofocus	HTML5. El control adquiere el foco en cuanto carga la página
cols	Anchura del cuadro en caracteres
disabled	El cuadro aparece deshabilitado
maxlength	Máximo número de caracteres permitidos en el cuadro
placeholder	HTML 5. Coloca 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 texto). Se usa como texto de ayuda.
readonly	El cuadro de texto es de solo lectura (no se puede cambiar su contenido)
required	Es obligatorio al menos elegir una opción del cuadro.
rows	Número de filas visibles en el cuadro (si se rellenan más, se accederá a ellas con una barra de desplazamiento).

[6] Agrupación de controles, fieldset

La etiqueta **fieldset** permite agrupar controles para que visualmente sea más cómodo el relleno de los controles. Visualmente los controles aparecerán recuadros y se suele utilizar una etiqueta inmediatamente anterior a **fieldset** que es **legend** que contiene el texto que encabezará al grupo de controles.

Ejemplo:

```

<form action="control_form.php" method="get">
  <fieldset>
    <legend>Datos personales</legend>
    <input type="radio" name="sexo" value="hombre"
          id="hombre">
    <label for="hombre"> Hombre</label><br>
    <input type="radio" name="sexo" value="mujer"
          id="mujer" checked="checked">
    <label for="mujer"> Mujer</label><br>

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

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

```

El resultado:




Ilustración 105. Controles agrupados

[7] Etiquetado de controles

En todos los ejemplos de código de este capítulo ya se ha usado el elemento de etiquetado **label**. Su labor es colocar una etiqueta explicativa a cada control.

Los elementos **label** utilizan un atributo **for** que sirve para asociar la etiqueta (que es el texto que aparece en pantalla acompañando al control) al control concreto. El valor de **for** se corresponde con el valor indicado en el atributo **id** del control.

Esto permite una mejor usabilidad (los navegadores reconocen **label** y, por ejemplo, al hacer clic en el texto de la etiqueta, el control asociado recibe el foco) y una mejor semántica del código.

Ejemplo para etiquetar un control de tipo **textarea**:

```
<form action="control_form.php" method="get">  
  <label for="texto">Escribe un texto descriptivo:</label>  
  <textarea rows="10" cols="40" name="texto" ></textarea><br>  
  <input type="submit" value="enviar"><br>  
</form>
```