

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from the bar, containing the text 'LMSGI'.

LMSGI

Manual de HTML5

U10. Formularios
Lenguajes de Marcas

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

IES ANTONIO SEQUEROS
DPTO INFORMÁTICA

Contenido

[10] Formularios.....	3
[10.1] introducción.....	3
[10.2] funcionamiento de los formularios. métodos GET y POST.....	4
[10.2.1]funcionamiento de los formularios.....	4
[10.2.2]métodos de paso de datos.....	6
[10.3] elemento form.....	8
[10.3.1]funcionamiento.....	8
[10.4] controles de tipo input.....	9
[10.4.1]elemento input.....	9
[10.4.2]atributos comunes del elemento input.....	9
[10.4.3]cuadros de texto.....	11
[10.4.4]cuadro de contraseñas.....	12
[10.4.5]botones.....	12
[10.4.6]controles ocultos.....	13
[10.4.7]botones de radio.....	13
[10.4.8]casillas de verificación.....	14
[10.4.9]cuadro de selección de archivo.....	15
[10.4.10]cuadros numéricos.....	16
[10.4.11]cuadros de fecha.....	17
[10.4.12]cuadro de selección de hora.....	17
[10.4.13]otros cuadros de selección de fechas.....	18
[10.4.14]cuadros deslizantes.....	18
[10.4.15]otros controles.....	18
[10.5] botones modernos.....	19
[10.5.1]atributos de la etiqueta button.....	19
[10.6] cuadros combinados.....	20
[10.6.1]atributos del elemento select.....	22
[10.6.2]atributos del elemento option.....	23
[10.7] cuadro de texto multilínea.....	23
[10.7.1]atributos del elemento textarea.....	24
[10.8] agrupación de controles, fieldset.....	25
[10.9] etiquetado de controles.....	26
[10.10] datalist.....	26

[10] Formularios

[10.1] 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.

La página <http://www.wufoo.com.mx/html5/> presenta una tabla (en inglés) muy interesante para saber, sobre cada nueva característica de los formularios HTML 5 qué navegadores la soportan (e incluso desde qué versión).

[10.2] funcionamiento de los formularios. métodos GET y POST

Formulario de login:

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

Ilustración 88. Ejemplo de formulario

[10.2.1]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 anterior](#).

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.

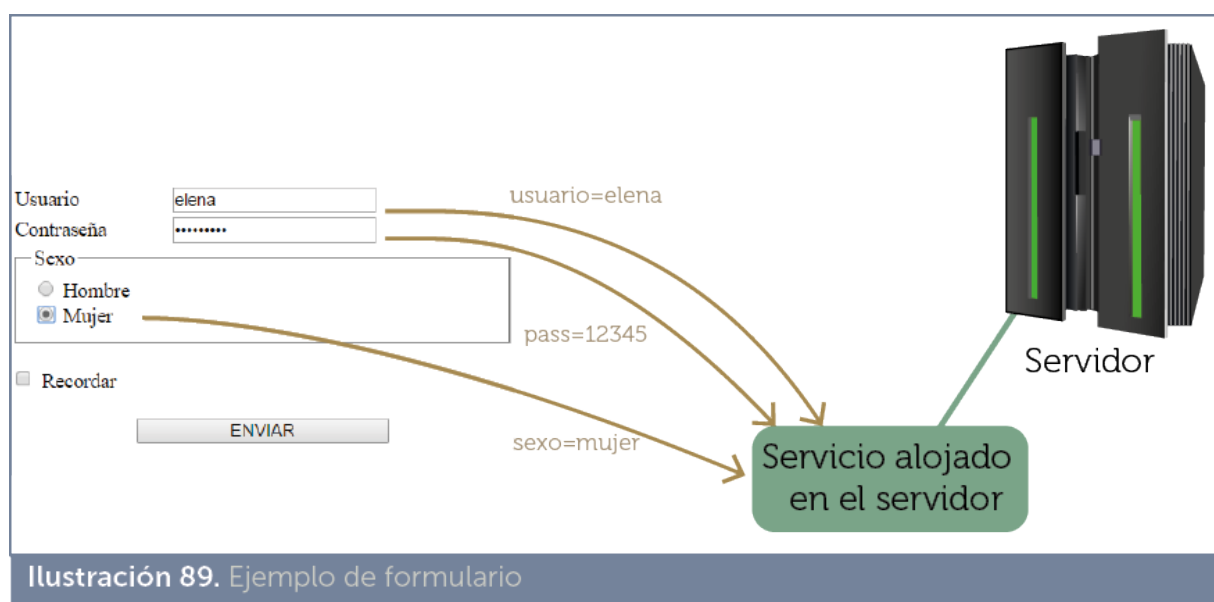


Ilustración 89. Ejemplo de formulario

Como se observa en la [ilustración anterior](#), 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 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>

  <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">

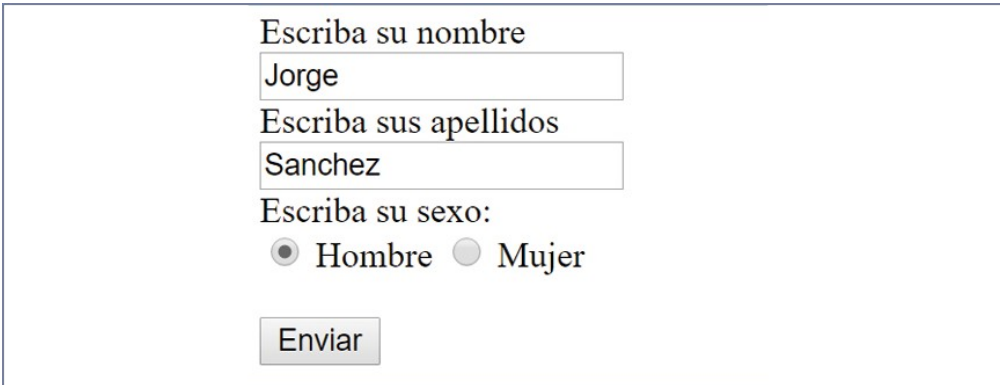
  <label for="mujer">Mujer</label><br><br>

  <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*.



Escriba su nombre
Jorge

Escriba sus apellidos
Sanchez

Escriba su sexo:
☒ Hombre ☐ Mujer

Enviar

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*.

[10.2.2] 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://urlpagina?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.

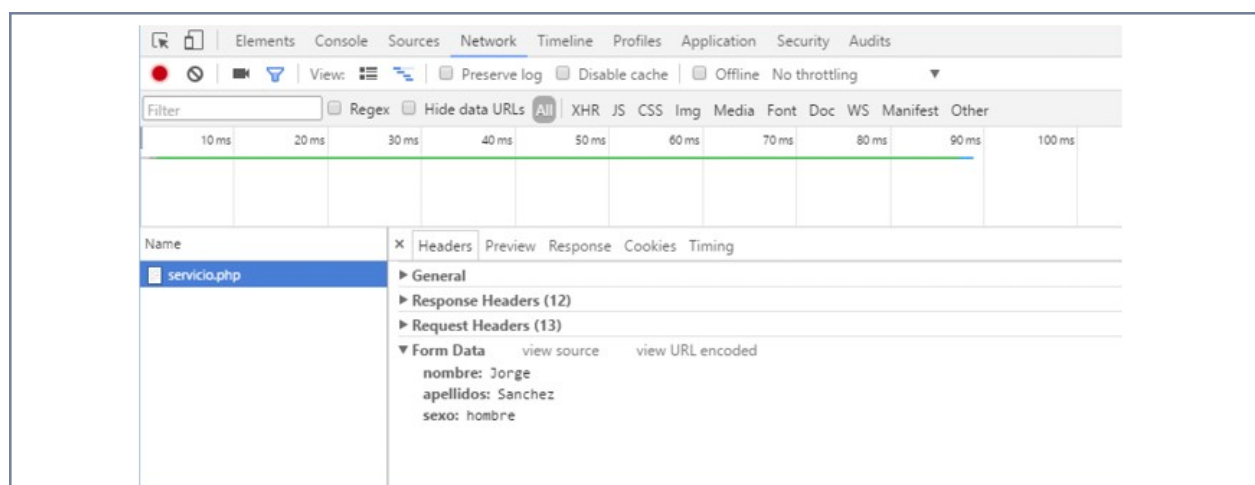


Ilustración 91. Panel de depuración de Google Chrome mostrando datos enviados usando el método POST

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).

[10.3] elemento form

[10.3.1]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. Se explica la diferencia al final de este capítulo.

[10.4] controles de tipo input

[10.4.1]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.

[10.4.2]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.
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 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
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
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	Indica que el control es de <i>solo lectura</i> ; es decir, no se podrá escribir en él.
required	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.
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 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	<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 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">
 </form></pre> <p>El resultado es:</p>

	Escribe el nº de serie <input type="text" value="5 letras y tres números"/> <input type="button" value="enviar"/>
--	---

[10.4.3]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:

Escriba su nombre <input type="text"/>
--

Ilustración 92. Cuadro de texto

[10.4.4]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**.

Escriba su nombre

Ilustración 93. Cuadro de contraseñas

[10.4.5]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 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 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.

[10.4.6]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**.

[10.4.7]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).

[10.4.8]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>

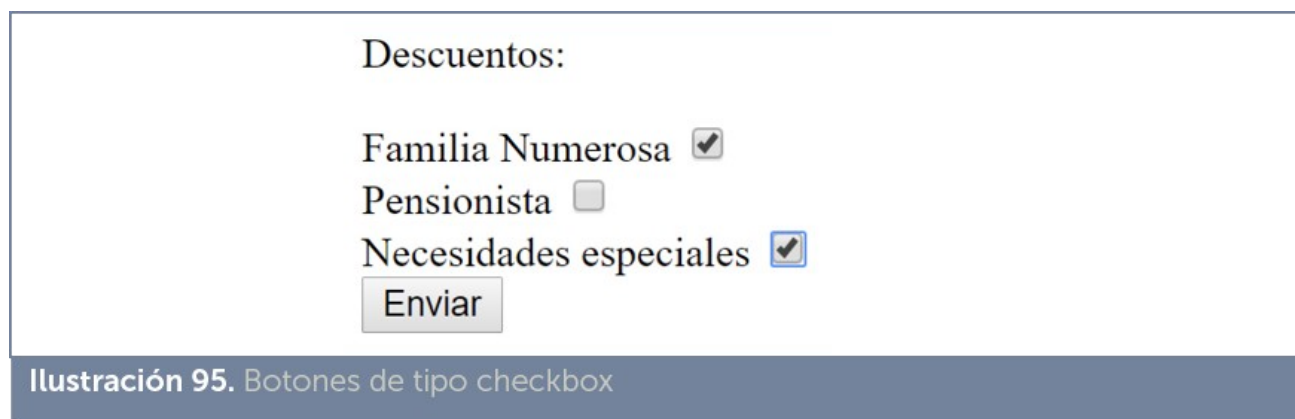
  <input type="checkbox" name="pension" id="pension"      value="pension"><br>

  <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.

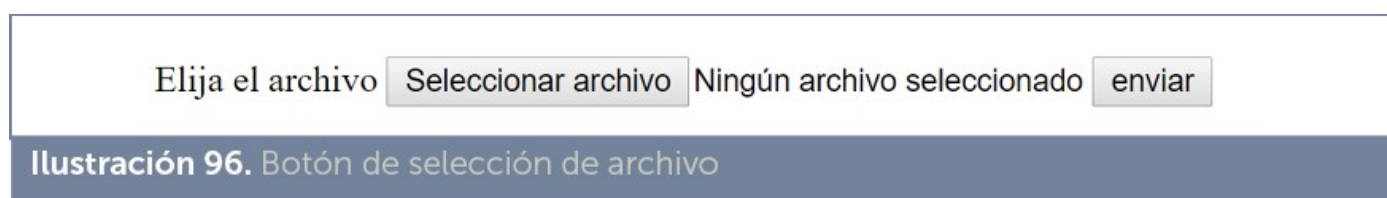
[10.4.9]cuadro de selección de archivo

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.

[10.4.10]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
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

[10.4.11]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:

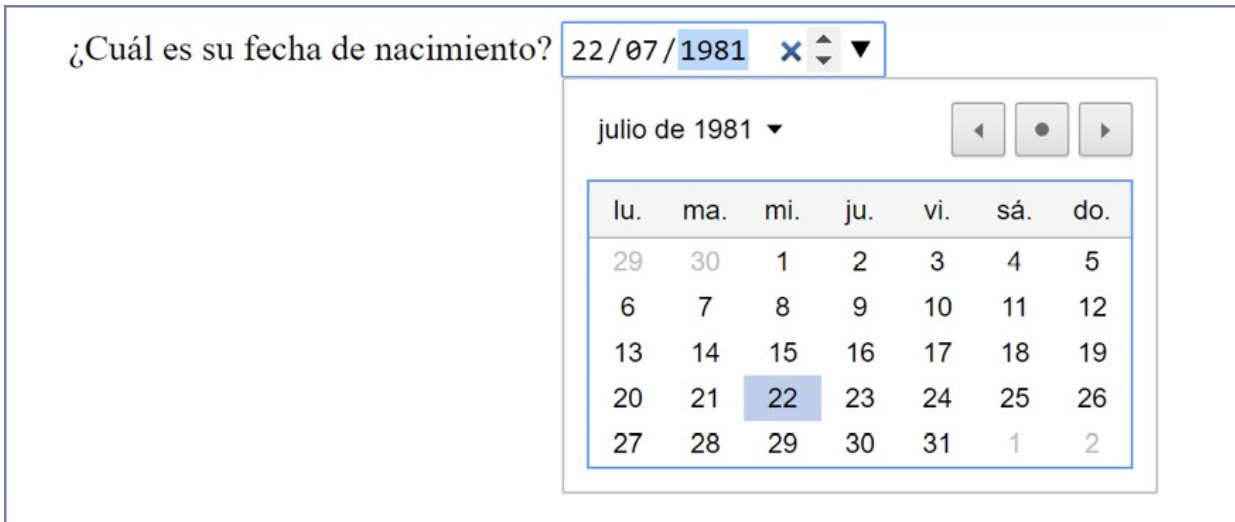
```
<form action="recogida1.php" method="get">

    <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:



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

julio de 1981 ◀ ● ▶

lu.	ma.	mi.	ju.	vi.	sáb.	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

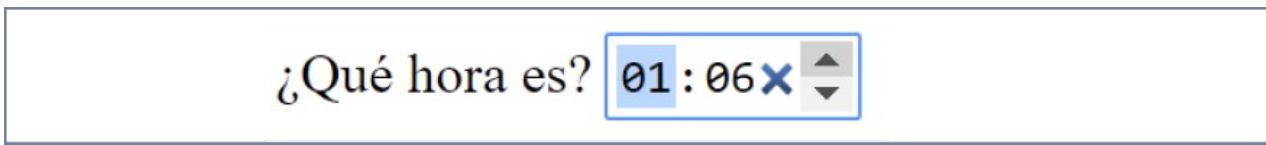
Ilustración 98. Cuadro de selección de fecha

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.

[10.4.12]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

[10.4.13]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.

[10.4.14]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:



[10.4.15]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.

[10.5] 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>clíc</strong></button>
```

Mostraría el botón:



Ilustración 101. Botón con texto en negrita

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.

[10.5.1] 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
form	Recibe el identificador del formulario al que pertenece el botón

[10.6] 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:

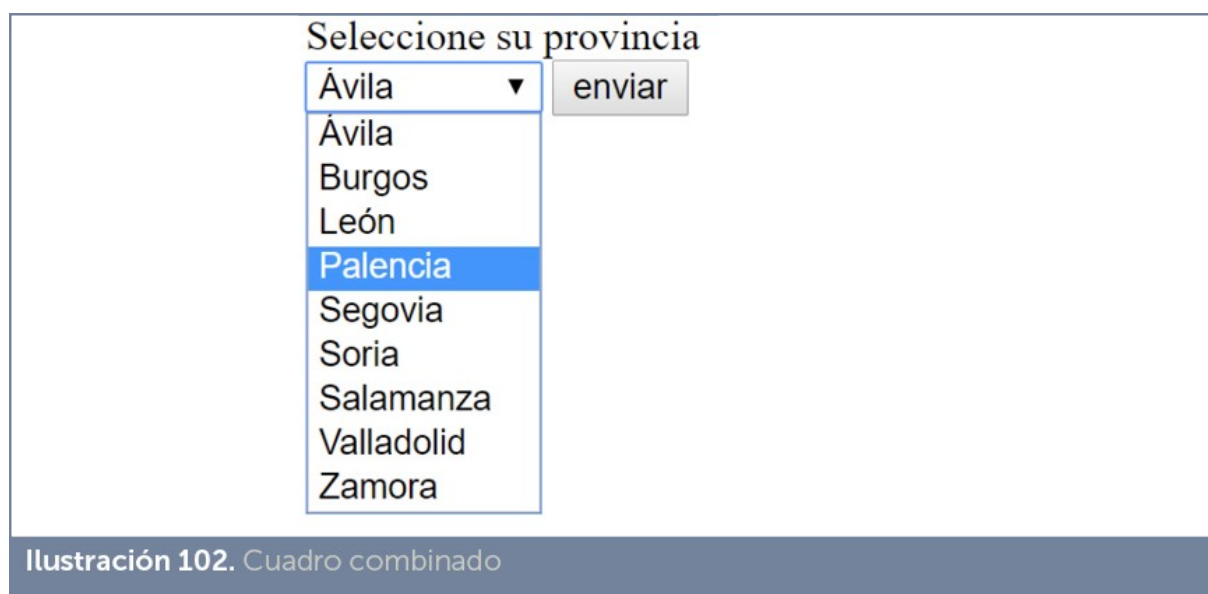


Ilustración 102. Cuadro combinado

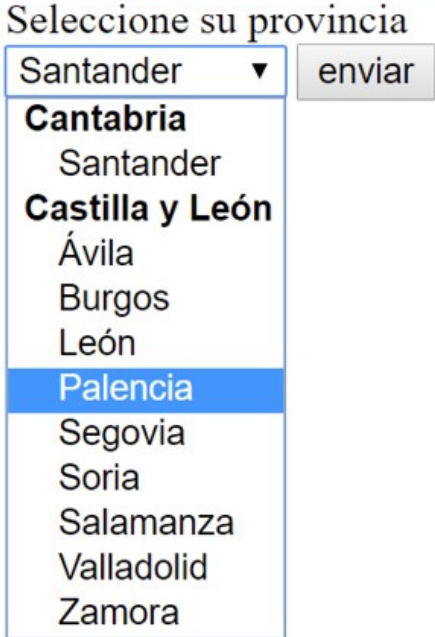
Se pueden agrupar opciones dentro del cuadro usando el elemento **optgroup**. Ejemplo:

```
<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

[10.6.1] 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
form	Recibe el identificador del formulario al que pertenece el botón
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	HTML 5. 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 contro..

[10.6.2] atributos del elemento option

atributo significado

disabled La opción aparece deshabilitada en el cuadro combinado

label Versión de texto de la opción más corta. No funciona en Firefox ni en Internet Explorer anterior a la versión 7

selected Opción seleccionada por defecto

value Valor que se envía al servicio que recibe los datos del formulario cuando se elige esta opción

[10.7] cuadro de texto multilinea

La etiqueta **textarea** permite colocar un cuadro de texto de varias líneas para que el usuario puede 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:

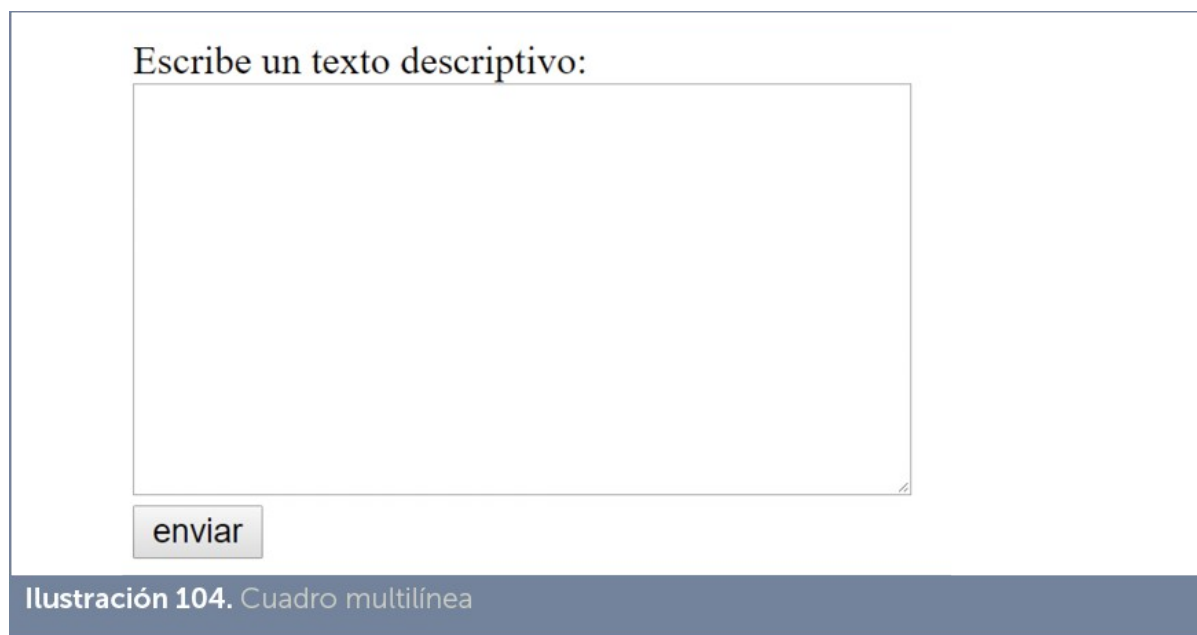
The screenshot shows a web form within a blue-bordered container. At the top, the text "Escribe un texto descriptivo:" is displayed. Below it is a large, empty text area with a thin gray border. At the bottom left of the text area, there is a gray button with the text "enviar" in black. The entire form is set against a white background.

Ilustración 104. Cuadro multilinea

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

[10.7.1] 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
form	Recibe el identificador del formulario al que pertenece el botón
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	HTML 5. 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).
wrap	Forma de encapsular el texto cuando se envían los datos al servicio de destino. Puede ser soft o hard .

[10.8] 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 recuadrados y se suele utilizar una etiqueta inmediatamente interior 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:



Datos personales

☐ Hombre

☒ Mujer

Estado civil: Casado ☒

Ilustración 105. Controles agrupados

[10.9] 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>
```

[10.10] datalist

Se trata de un elemento HTML 5 muy potente. Permite añadir entradas a un control de cuadro de texto (y también a cuadros especializados como los de email, url,...). La forma de utilizarlo consiste en usar el atributo HTML5 **list** existente en la etiqueta **input**. Ese atributo asociará el cuadro de texto a la lista de valores a través del identificador del **datalist**. Luego dentro de **datalist** se colocan etiquetas **option** para cada opción en la lista (al estilo de los cuadros de tipo select).

Ejemplo:


```
<form action="recogida.php" method="get">
    <label for="gustos">
        Escribe en qué te gusta pasar tu tiempo de ocio
    </label>
    <input type="text" id="gustos" name="gustos" list="listaGustos" >

    <datalist id="listaGustos">

        <option label="deportes" value="Deportes" >
        <option label="teatro" value="Teatro">
        <option label="cine" value="Cine">
        <option label="leer" value="Leer">
    </datalist>
    <input type="submit" value="enviar"><br>
</form>
```

No aparece un cuadro combinado, será un cuadro de texto en el que al arrimar el ratón aparecen las entradas indicadas (como sugerencias de escritura), pero podremos escribir lo que queramos (sin elegir ninguna opción de la lista).

El código anterior permitiría esta situación:



Escribe en qué te gusta pasar tu tiempo de ocio

Deportes
Teatro
Cine
Leer

enviar

Ilustración 106. Datalist

El usuario puede elegir cualquiera de las opciones de la lista de sugerencias, pero también puede escribir lo que quiera en el cuadro.

Sin embargo esta forma no funciona en algunos navegadores (solo en los modernos y Safari no ha incorporado esta etiqueta), por ello se suele usar un truco que consiste en meter un elemento **select** dentro del elemento **datalist**. Los navegadores modernos ignorarán la etiqueta **select** y los viejos el **datalist**.

```
<form action="recogida.php" method="get">
  <label for="gustos">
    Escribe en qué te gusta pasar tu tiempo de ocio
  </label>
  <datalist id="listaGustos">

    <select name="listaGustos">

      <option value="deportes">Deportes</option>
      <option value="teatro">Teatro</option>
      <option value="cine">Cine</option>
      <option value="leer">Leer</option>

    </select>

  </datalist>

  <input id="gustos" name="gustos" list="listaGustos">
  <input type="submit" value="enviar"><br>
</form>
```