

Lenguajes de marcas y sistemas de gestión de la información



UT02 – HTML

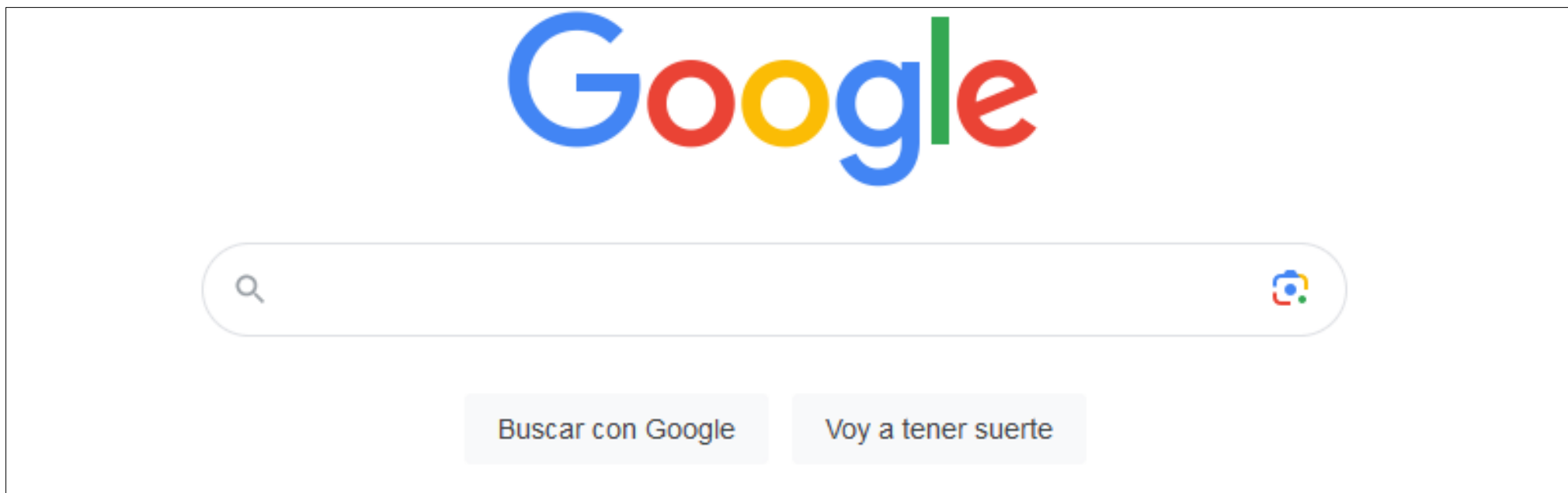
8 – Formularios – Introducción – Campos básicos – Botones

¿Qué es un formulario?

Un formulario es una de las formas que tiene un sitio web para interactuar con sus usuarios.

Permite la recogida de datos y su envío al servidor para que sean procesados.

Ejemplo de formulario: la página inicial de Google.es:



Componentes de un formulario

Un formulario está compuesto de:

- Uno o más controles de formulario, como, por ejemplo:
 - Cajas de texto (de una o de varias líneas)
 - Selectores desplegables (select)
 - Casillas de verificación (checkbox)
 - Botones de opción (radiobutton)
- Contenidos dentro de un elemento <form>
- Con un botón para hacer el envío de los datos
- Opcionalmente, con unas reglas de validación de los datos

El elemento <form>

Elemento de bloque que delimita y contiene los campos del formulario.

Principales atributos específicos para configurar cómo se comporta:

- **action:** URL a la que se envían los datos en los campos del formulario.
- **method:** método HTTP que se debe usar para enviar los datos.
Normalmente POST, pero se puede usar GET.

Todos los atributos de form (estos dos y los demás que veremos más adelante) son opcionales, pero se considera buena práctica poner al menos estos dos. Si no se ponen:

- **action** será la URL de la página del formulario. Por así decirlo, el formulario se envía a sí mismo.
- **method** será GET.

El elemento `<input>`

Se utiliza para definir la mayoría de los campos de formulario. Es un elemento en línea. Con el atributo “type” se indica qué tipo de campo de formulario se debe mostrar.

En general, debe ir acompañado de una etiqueta, que se crea con el elemento “<label>” que se asocia al input.

Para asociar la etiqueta al input se usa el atributo “for” de label, con el valor del atributo “id” del input, o se coloca el input dentro del tag label.

Esta asociación de los inputs con sus label es fundamental para conseguir formularios accesibles.

El elemento `<button>`

Se utilizan para realizar acciones en el formulario. Se pueden activar con un clic de ratón, una pulsación de teclado, con el dedo (táctil), con un comando de voz (asistencia a personas con movilidad reducida), etc.

Se representa normalmente como un botón, pero su aspecto puede cambiarse con CSS.

Al activarlo, se realiza una acción. La acción a realizar dependerá del tipo de botón, que se especifica con el atributo `type`. Entre otros tenemos:

- `type="submit"`. Este botón envía el formulario.
- `type="reset"`. Este botón devuelve el formulario al estado inicial. No se recomienda su uso porque los pueden confundir con el botón para enviar los datos y provoca la pérdida de los cambios realizados en el resto de los campos.

Ejemplo de formulario simple

```
<form action="recogida-datos.html" method="post">
  <p>
    <label for="input-nombre">Nombre: </label>
    <input type="text" name="nombre" id="input-nombre">
  </p>
  <p>
    <label>Apellidos:
      <input type="text" name="apellidos">
    </label>
  </p>
  <p><button type="submit">Enviar</button></p>
</form>
```

Experimenta un poco...

Crea una página HTML e incluye un formulario.

Incluye distintos tipos de campos, usando input con distintos valores para el atributo "type".

Prueba también el elemento <textarea>.

Y prueba el elemento <select>. Este último permite elegir entre múltiples valores, y cada uno de ellos se tiene que definir con el elemento <option>.

El método get

Cuando un formulario usa method="get", los datos del formulario aparecen en la URL, en la barra de direcciones del navegador.

```
<form method="get" action="http://localhost:8080/view-form/get">  
  <label>Usuario: <input type="text" name="usr"></label>  
  <label>Contraseña: <input type="password" name="pwd"></label>  
  <button type="submit">Enviar</button>  
</form>
```

En la URL del navegador podremos ver los valores de los campos:

```
http://localhost:8080/view-form/get?usr=&pwd=
```

El método post

Cuando un formulario usa `method="post"`, los datos del formulario aparecen no aparecen en la URL, en la barra de direcciones del navegador. Viajan como parte del cuerpo de la petición.

```
<form method="post" action="http://localhost:8080/view-form/post">  
  <label>Usuario: <input type="text" name="usr"></label>
```

En la URL del navegador no veremos los datos. Podrían verse, por ejemplo, con las herramientas de desarrollo de Chrome.

<http://localhost:8080/view-form/post>

▼ **Form Data** view source

usr: jllopezalvarez

pwd: pwdfacildeadivina

Controles básicos – Elemento input

Una gran parte de los controles se crean con el elemento “input”. El atributo “type” de este elemento determina el tipo de campo.

Atributo type	Tipo de control
text	Texto (1 línea)
password	Contraseña (no se ve el texto escrito)
radio	Botones de selección
checkbox	Casilla de verificación
file	Selector de ficheros

Hay muchos tipos más avanzados, definidos por HTML5, que veremos más adelante.

Controles básicos – Botones

Para enviar un formulario es necesario un elemento que realice el envío.

Este elemento puede ser:

- Elemento button. Esta es la forma preferida en HTML 5.

```
<button type="submit">Texto del botón</button>
```

- Elemento input de tipo submit. Esta forma es la original, pero se recomienda usar button en su lugar. El texto del botón se escribe en el atributo value.

```
<input type="submit" value="Texto del botón">
```

- Elemento input de tipo image. Esta forma envía también las coordenadas de la imagen en las que se hizo clic.

```
<input type="image" src="..." alt="...">
```

Elemento label

Cada campo del formulario debería tener un elemento label asociado. Este elemento es la “etiqueta” que el usuario ve, la que indica la utilidad del campo.

```
<form method="post" action="http://localhost:8080/view-form/post">  
  <label>Usuario: <input type="text" name="usr"></label>  
  <label>Contraseña: <input type="password" name="pwd"></label>  
  <button type="submit">Enviar</button>  
</form>
```

La etiqueta describe el campo de formulario, y, cuando se hace clic en ella, el navegador hace foco del teclado en el campo asociado.

Elemento label – Asociación implícita

Cuando un control de formulario se encuentra dentro de una etiqueta label, el navegador, automáticamente, asocia la etiqueta al campo.

```
<form method="post" action="http://localhost:8080/view-form/post">  
  <label>Usuario: <input type="text" name="usr"></label>  
  <label>Contraseña: <input type="password" name="pwd"></label>  
  <label>  
    <input type="checkbox" name="remember" value="true">  
    Recordarme</label>  
  <button type="submit">Enviar</button>  
</form>
```

Elemento label – Asociación explícita

Cuando no se puede anidar el control dentro de un label, se usa el id del control y el atributo “for” del label:

```
<form method="post" action="http://localhost:8080/view-form/post">
  <p><label for="usr">Usuario:</label></p>
  <p><input type="text" name="usr" id="usr"></p>
  <p><label for="pwd">Contraseña:</label></p>
  <p><input type="password" name="pwd" id="pwd"></p>
  <p>
    <input type="checkbox" name="remember"
              id="remember" value="true">
    <label for="remember">Recordarme</label>
  </p>
  <button type="submit">Enviar</button>
</form>
```

Atributos name / id

El atributo “name” de los controles se usa para enviar la información al servidor.

El atributo id se usa para asociar el control con una etiqueta (label) o para realizar acciones con JavaScript.

El atributo id es un atributo global (lo puede tener cualquier elemento HTML), pero es importante recordar que no puede repetirse en dos elementos, por lo tanto, no puede haber dos controles con el mismo id.

El atributo name también se usa para agrupar distintos controles “radio” para que sean excluyentes (sólo pueda seleccionarse uno)

Ejemplo radio buttons no excluyentes

```
<form method="post" action="http://localhost:8080/view-form/post">
  <p><label>
    <input type="radio" name="opcion1"
      value="1">Opcion 1</label></p>
  <p><label>
    <input type="radio" name="opcion2"
      value="2">Opcion 2</label></p>
  <p><label>
    <input type="radio" name="opcion3"
      value="3">Opcion 3</label></p>
  <button type="submit">Enviar</button>
</form>
```

Los radio buttons no son excluyentes (se pueden seleccionar varios a la vez) porque no tienen todos el mismo valor del atributo "name".

Ejemplo radio buttons excluyentes

```
<form method="post" action="http://localhost:8080/view-form/post">
  <p><label>
    <input type="radio" name="opcion"
      value="1">Opcion 1</label></p>
  <p><label>
    <input type="radio" name="opcion"
      value="2">Opcion 2</label></p>
  <p><label>
    <input type="radio" name="opcion"
      value="3">Opcion 3</label></p>
  <button type="submit">Enviar</button>
</form>
```

Los radio buttons ahora sí son excluyentes (se puede seleccionar sólo uno de ellos) porque tienen todos el mismo valor del atributo "name".

Múltiples valores para una misma variable

Si hay varios campos con un mismo atributo “name”, se enviará al servidor múltiples valores para la misma variable.

Por ejemplo, si tenemos estos campos:

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

Se recibirá en el servidor cuatro valores, uno por cada campo, todos asociados a “color”. En la petición HTTP se puede ver lo que se envía.

Y sin interpretar:

```
color=Rojo&color=Azul&color=Amarillo&color=Negro
```

Datos de formulario

▼ color: [...]

0: "Rojo"

1: "Azul"

2: "Amarillo"

3: "Negro"

Múltiples valores para una misma variable

Algunos lenguajes de servidor como PHP, necesitan, para que se interprete correctamente este tipo de envíos, que se use un valor especial para el atributo "name".

En estos casos (sólo si vamos a usar PHP en el servidor, u otro lenguaje que lo necesite), se deben añadir corchetes al nombre del campo, para indicar que se debe interpretar en el servidor como un array de valores:

```
<input type="text" name="color[]">  
<input type="text" name="color[]">  
<input type="text" name="color[]">  
<input type="text" name="color">
```

Más información: [PHP Tutorial - PHP Form Multi-Value Fields](#)

Textarea

Control para que el usuario pueda introducir textos de más de una línea.

Introduce la descripción (varias líneas)



```
<label for="textarea01">
  Introduce la descripción (varias líneas)
</label>
<textarea id="textarea01"></textarea>
```

Select simple

Control para seleccionar una opción entre varias.

```
<label for="selectorsimple">Seleccione una opción</label>
<select name="selectorsimple" id="selectorsimple">
  <option value="LLMM">Lenguajes de marcas</option>
  <option value="FOL">FOL</option>
  <option value="PROG">Programación</option>
  <option value="SI">Sistemas informáticos</option>
  <option value="BBDD">Bases de datos</option>
  <option value="EEDD">Entornos de desarrollo</option>
</select>
```

Seleccione una opción

Lenguajes de marcas ▼

Lenguajes de marcas

FOL

Programación

Sistemas informáticos

Bases de datos

Entornos de desarrollo

Select simple en modo lista

Control para seleccionar una opción entre varias, pero se muestran todos los elementos (no tiene flecha para desplegar las opciones). Se hace con el atributo "size".

```
<label for="selectorsimple2">Seleccione una opción</label>
<select name="selectorsimple2" id="selectorsimple2" size="4">
  <option value="LLMM">Lenguajes de marcas</option>
  <option value="FOL">FOL</option>
  <option value="PROG">Programación</option>
  <option value="SI">Sistemas informáticos</option>
  <option value="BBDD">Bases de datos</option>
  <option value="EEDD">Entornos de desarrollo</option>
</select>
```

Seleccione una opción

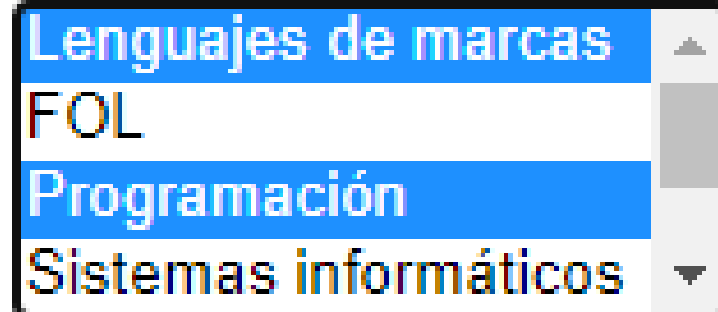
Lenguajes de marcas
FOL
Programación
Sistemas informáticos

Select múltiple

Control para seleccionar una o varias opciones entre varias. Se usa el atributo "multiple". Para seleccionar varias opciones se usa la tecla "control".

```
<label for="selectormultiple">Seleccione una o  
varias opciones</label>  
<select name="selectormultiple" id="selectormultiple"  
multiple size="4">  
  <option value="LLMM">Lenguajes de marcas</option>  
  <option value="FOL">FOL</option>  
  <option value="PROG">Programación</option>  
  <option value="SI">Sistemas informáticos</option>  
</select>
```

Seleccione una o varias opciones



Selector de ficheros – Input type="file"

Control para seleccionar un fichero y enviarlo al servidor. El formulario debe usar el método POST, y usar la codificación enctype="multipart/form-data".

```
<form action="recogerFichero.php"
      method="post" enctype="multipart/form-data">
  <label for="fichero">Seleccione un fichero</label>
  <input type="file" name="fichero" id="fichero">
</form>
```

Seleccione un fichero

Seleccionar archivo

Ningún archivo seleccionado

Enviar información oculta – Type="hidden"

A veces las aplicaciones web tienen que enviar información al servidor que no es necesario que el usuario visualice, y que no tiene que editar.

Para estos casos se puede usar el elemento input, con type="hidden".

Este elemento se envía con el resto del formulario, pero no es visible de ninguna forma.

```
<input type="hidden" name="personId" value="1232">
```

Se suele utilizar en las aplicaciones que tienen que editar registros de una base de datos, para enviar de vuelta al servidor el identificador del registro que está siendo modificado, junto a los nuevos datos.