

# Formularios HTML

Gabriel Chaves Benítez (gchaves@us.es)

Departamento de Ciencias de la Computación e Inteligencia Artificial  
Universidad de Sevilla

26-01-2026, Bases de Datos

Los formularios son fundamentales para pasar de un sitio web estático a uno **dinámico**, a nivel de **comunicación con el servidor**.

- Permiten **introducir datos** por parte del usuario de la web.
- Estos datos pueden **enviarse al servidor** por peticiones HTTP.
- Hay **varias opciones** de manejo de esa comunicación:
  - Comunicar directamente HTML con PHP (queda ya fuera de la asignatura).

Los formularios son fundamentales para pasar de un sitio web estático a uno **dinámico**, a nivel de **comunicación con el servidor**.

- Permiten **introducir datos** por parte del usuario de la web.
- Estos datos pueden **enviarse al servidor** por peticiones HTTP.
- Hay **varias opciones** de manejo de esa comunicación:
  - Comunicar directamente HTML con PHP (queda ya fuera de la asignatura).
  - Llamar a un servidor (e.g. NodeJS) mandándole los datos.

# Formularios HTML

`<form action="ruta_servidor"> Contenido </form>`

- Un formulario es una **zona** del cuerpo de un documento HTML **con la que el usuario puede interactuar**, realizando selecciones o rellenando campos de texto.
- *Contenido* puede incluir cualquier contenido HTML con **controles de formulario**.
- `<form>` no muestra nada visual, pero sí los controles que contiene, con los que el usuario puede interactuar.
- `action` indica dónde se envían los datos del usuario, en caso de gestionarlo desde HTML (**no vamos a emplearlo nosotros**).
- Un documento puede tener varios formularios.

# Controles de formulario

- Control de propósito múltiple: `<input>`, con atributos:
  - Nombre del control: `name="nombre"`
  - Tipo del control: `type="tipo"`
    - Campos de texto, selección de opciones, ficheros, botones, campos ocultos.
  - Valor del control: `value="valor"`
  - Tamaño del control: `size="medida"`
  - Otros atributos según el tipo
- El `name` del control es necesario cuando empleamos la acción del formulario para enviarlo, para identificar los datos en el servidor.
- Es habitual tener además un `id` por cada control, cuando enviamos el formulario con técnicas más actuales basadas en Javascript.

# Campos de texto

- Hay varios tipos de campos de texto, entre otros:
  - Campo de texto de una línea: `<input type="text">`
  - Campo de contraseña: `<input type="password">`
- El `value` de estos controles es el texto escrito.
- Las contraseñas no se muestran al usuario.
- Atributos para estos tipos:
  - Tamaño del control en nº caracteres: `size="numero"`
  - Tamaño máximo del valor: `maxlength="numero"`
  - Texto por defecto: `value="texto"`
  - Info. en el propio control: `placeholder="texto"`

# Áreas de texto

- `<textarea>` *Contenido* `</textarea>`
- Atributos para esta etiqueta:
  - Nombre del control: `name="nombre"`
  - Número de filas visibles: `rows="numero"`
  - Número de columnas visibles: `cols="numero"`
  - El valor del control es su contenido.

```
<form>
```

```
  <textarea name="comentario" rows="5" cols="20">
```

```
    Escribe aquí tus comentarios
```

```
  </textarea>
```

```
</form>
```

# Selección de opciones

- Hay dos tipos de controles de selección de opciones:
  - Selección de una opción entre varias:  
`<input type="radio">`
  - Selección de una opción aislada:  
`<input type="checkbox">`
- El valor de las opciones seleccionadas es "on", salvo que se haya indicado otro distinto con el atributo value
- Las opciones disponibles de un mismo control de tipo "radio" se identifican con el mismo name (y  $\neq$  value)
- Atributos para estos tipos:
  - Opción seleccionada por defecto: checked

Para ver más detalles y ejemplos podemos acceder [aquí](#).

# Subida de ficheros

- Hay un único tipo de control de selección de fichero:
  - `<input type="file">`
- El navegador muestra un campo de texto donde se ha de indicar la localización absoluta del fichero que se quiere enviar, y un botón que da paso a una ventana de selección de fichero a través de un navegador de directorios.
- El valor de este control es el contenido del fichero indicado.

Podemos encontrar más detalles sobre la etiqueta en [este enlace](#).

# Controles input de tipo botón

- Hay cuatro tipos de botones<sup>1</sup>:
  - Botón de envío de datos: `<input type="submit">`
  - Botón gráfico de envío: `<input type="image">`
  - Botón de borrado: `<input type="reset">`
  - Botón genérico: `<input type="button">`
- El atributo `value` sirve para indicar el texto del botón
- Los botones tienen una acción asociada según su `type`:
  - `submit` e `image` envían los datos al servidor
  - `image` admite `src="URI"` para asociar la imagen
  - `reset` pone los controles en su estado inicial
  - `button` depende de la acción que le asociemos

---

<sup>1</sup>Más detalles [aquí](#)

# Botones con button

- En lugar de inputs de tipo button, tenemos la etiqueta:
  - `<button> Contenido </button>`<sup>2</sup>
  - Dentro de *Contenido* podemos encontrar texto, imágenes, etc.
- Atributos de esta etiqueta
  - Nombre del control: `name="nombre"`
  - Valor del control: `value="valor"`
  - Tipo del control: `type="tipo"`
    - Botón de envío: `type="submit"`
    - Botón de borrado: `type="reset"`
    - Botón genérico: `type="button"`

---

<sup>2</sup>Más información [aquí](#).

# Campos ocultos

- Campo oculto:
  - `<input type="hidden">`
- No se visualiza nada en el navegador.
- Se puede ver a través de la opción del navegador: “ver el código fuente de la página”.
- Son accesibles para el código dinámico de cliente (Javascript) y pasarlo al servidor.
- El valor del control es el indicado con el atributo `value`

# Menús desplegables con select

- Menú desplegable/de selección:
  - `<select> opciones </select>`
- Atributos de esta etiqueta
  - Nombre del control: `name="nombre"`
  - Número de filas visibles: `size="numero"`
  - Selección múltipe: `multiple`
  - Opciones a seleccionar:
    - `<option> texto </option>`
    - Valor por defecto: `value="valor"`
    - Opción seleccionada por defecto: `selected`

# Menús desplegables con select

## Ejemplo

```
<form>
  Aficiones:
  <select align="top" name="aficiones" size="3" multiple>
    <option value="v1">Música</option>
    <option value="v2" selected>Cine</option>
    <option value="v3" selected>Lectura</option>
    <option value="v4">Maquetas</option>
    <option value="v5">Fotografía</option>
    <option value="v6">Informática</option>
  </select>
</form>
```

# Menús desplegables con datalist

- Menú desplegable con input y list:
  - `<input list="datalistId">`
- Datalist:
  - Agrupación de opciones para otros controles, como el input anterior.
  - Opciones a seleccionar:
    - `<option value="texto">`

# Menús desplegables con datalist

## Ejemplo

```
<label for="aficiones2">Afición principal</label>
<input list="affs" name="aficiones2" />
<datalist id="affs">
  <option value="Música">
  <option value="Cine">
  <option value="Lectura">
  <option value="Maquetas">
  <option value="Fotografía">
  <option value="Informática">
</datalist>
```

- Además de controles, hemos visto en los formularios las etiquetas que proporcionan información sobre los controles: `<label>Texto</label>`.
- Todos los controles vistos admiten el atributo booleano `disabled` que los desactiva, de modo que no se envíen al servidor.

# Envío de datos del formulario

- La información introducida por el usuario en los controles, debe enviarse al servidor para su procesamiento (en PHP, NodeJS, Python, Java, Ruby o algún otro lenguaje de servidor). Existen varias vías para ello:
  - Emplear el atributo `action` del `<form>`, de modo que pasa los valores de los controles, cada uno asociado a su `name`. Se activa a través del `submit` del formulario.
  - Manejar la acción mediante JavaScript, y desde ella lanzar la petición al servidor. Se puede activar mediante un botón genérico o cualquier otro evento (***esta será nuestra opción***).

# Envío de datos del formulario

Atributos de `<form>` sobre todo para la opción directa, sin pasar por JS (**no la emplearemos**):

- Ruta al *endpoint* del servidor: `action="URI"`
- Forma en que se pasan los datos a dicho elemento:  
`method="post"|"get"`
  - El método `get` (método por defecto), habitual en formularios que envían poca información, pasa los datos explícitos a la URI.
  - El método `post`, generalmente usado con formularios que envían gran cantidad de información, manda los datos en el cuerpo de la petición. Si se envían ficheros completos, entonces es necesario indicar la siguiente codificación del contenido: `enctype="multipart/form-data"`.

Al margen del contenido enlazado en la web de la asignatura, podemos encontrar más información a través de los siguientes enlaces:

- [Formularios en lenguajehtml.com](#), de Manz.dev, con muy buenas explicaciones y ejemplos.
- [Formularios en W3Schools](#), para poder interactuar con los ejemplos de la página, modificarlos y probarlos.