



La Sénia

Institut d'Educació Secundària
Paiporta

UD4. Formularios

Desarrollo Web en Entorno Servidor

Profesora: Silvia Vilar Pérez

curso 2024-2025

Contenidos

- Formularios: atributos
- Elemento LABEL
- Elemento FIELDSET (agrupa controles)
- Elemento TEXTAREA (caja texto múltiples líneas)
- Elementos de tipo INPUT
- Elemento SELECT (desplegable)
- Servidor interno PHP

Atributos de Form

Los formularios nos permiten recoger datos del usuario para enviarlos al servidor y que éste los procese.

La etiqueta **<form>** tiene los siguientes atributos:

- **action**: Indica el fichero en el servidor al que se enviarán los datos del formulario. Si se omite, es la página actual
- **method**: indica el método HTTP a usar para enviar los datos. Por defecto es GET
- **autocomplete**: Permite (o no) completar con datos introducidos anteriormente por el usuario. Por defecto su valor es “on”
- **name**: especifica el nombre del formulario
- **novalidate**: es un atributo booleano que, si se incluye, indica que los datos del formulario no deben ser validados al enviarse

Elementos de Form

Podemos encontrar varios elementos en un formulario:

- Elemento LABEL
- Elemento FIELDSET (agrupa controles)
- Elemento TEXTAREA (caja texto múltiples líneas)
- Elementos de tipo INPUT
 - ✓ TEXT (caja de texto de una única línea)
 - ✓ PASSWORD
 - ✓ HIDDEN (control oculto para obtener datos)
 - ✓ RADIO (botón selección)
 - ✓ CHECKBOX (check de selección)
 - ✓ BUTTON
 - ✓ SUBMIT
 - ✓ RESET
 - ✓ FILE (para selección de archivos a enviar)
- Elemento SELECT (desplegable)
 - ✓ Simple (sólo un elemento)
 - ✓ Múltiple (varios elementos a la vez)

LABEL

El elemento **<label>** define una etiqueta para otros elementos. Ayuda a describir el el input que debe rellenar el usuario

Con radio button o checkbox, se marca la opción al pichar encima de la etiqueta

Su atributo **for** debe tener el valor del atributo **id** del input al que acompaña para que se puedan correlacionar

Label:

```
<label for="nombre">Nombre:</label>
```

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

FIELDSET

El elemento **<fieldset>** es un contenedor que agrupa diversos controles que tienen datos relacionados

El elemento **<legend>** define la leyenda o texto para el grupo de controles

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

```
<fieldset>
```

```
<legend>Datos Personales:</legend>
```

```
<label for="nombre">Nombre:</label><br>
```

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

```
<label for="apellido">Apellido:</label><br>
```

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

```
<input type="submit" value="Enviar">
```

```
</fieldset>
```

```
</form>
```

A diagram illustrating the visual representation of a fieldset. It consists of a rectangular box with a thin border. The word "Fieldset" is written in a blue, serif font at the top left of the box. A horizontal line extends from the right side of the box, and a vertical line extends from the bottom of the box, suggesting a continuation of the form structure.

TEXTAREA

El elemento **TEXTAREA** muestra un campo de texto multilínea (área de texto)

- Ejemplo

Comentario:

```
<TEXTAREA COLS="50" ROWS="4" NAME="comentario">
```

Este libro me parece ...

```
</TEXTAREA>
```

- Se obtienen los datos del campo en PHP:

```
<?PHP
```

```
$comentario = $_REQUEST['comentario'];
```

```
echo $comentario;
```

```
?>
```

INPUT TEXT

El elemento INPUT de tipo **TEXT** muestra un campo de texto de una sola línea

- Ejemplo:

Introduzca la cadena a buscar:

```
<INPUT TYPE="text" NAME="cadena" VALUE="valor por defecto" SIZE="20">
```



- Se obtienen los datos del campo en PHP:

```
<?php  
$cadena = $_REQUEST['cadena'];  
echo $cadena;  
?>
```


INPUT PASSWORD

El elemento INPUT de tipo **PASSWORD** muestra un campo de texto de una sola línea en el cual los caracteres se ocultan al escribirlos (normalmente puntos)



- Ejemplo

Contraseña:

```
<INPUT TYPE="password" NAME="clave">
```

- Se obtienen los datos del control en PHP:

```
<?PHP
```

```
$clave = $_REQUEST['clave'];
```

```
echo $clave;
```

```
?>
```

INPUT HIDDEN

El elemento INPUT de tipo **HIDDEN** es un control que no se muestra al usuario (oculto) y se usa para incluir datos que no pueden ser vistos ni modificados por el usuario.

- Ejemplo

```
<INPUT TYPE='hidden' NAME='username'  
VALUE="$usuario">
```

- Se obtienen los datos del control en PHP:

```
<?PHP  
$username = $_REQUEST['username'];  
echo $username;  
?>
```

INPUT RADIO

El elemento INPUT de tipo **RADIO** muestra radio button para selección exclusiva de SÓLO UNO de los elementos a elegir, por lo que los controles tienen el mismo nombre y distintos valores. La opción por defecto se indica con **checked**

- Ejemplo:

Radio: ☒ opción 1 ☐ opción 2

Sexo:

```
<INPUT TYPE="radio" NAME="sexo" VALUE="M"
CHECKED>Mujer
```

```
<INPUT TYPE="radio" NAME="sexo" VALUE="H">Hombre
```

- Se obtienen los datos del control en PHP:

```
<?PHP
```

```
$sexo = $_REQUEST['sexo'];
echo ($sexo);
?>
```

INPUT CHECKBOX

El elemento INPUT de tipo **CHECKBOX** muestra casillas de verificación para selección CERO o MÁS elementos a elegir, por lo que los controles tienen el mismo nombre y los valores se almacenan en un array. La/s opción/es marcadas por defecto se indican con **checked**

- Ejemplo

```
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="garaje" CHECKED>Garaje
```

```
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="piscina">Piscina
```

```
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="jardin">Jardín
```

- Se obtienen los datos del control en PHP:

```
<?PHP
```

```
$extras = $_REQUEST['extras'];
```

```
foreach ($extras as $extra)
```

```
echo "$extra<br>\n";
```

```
?>
```

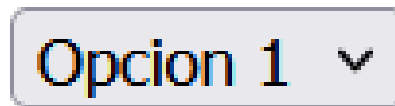
☒ opción 1 ☒ opción 2 ☐ opción 3

SELECT SIMPLE

El elemento **SELECT SIMPLE** muestra un desplegable con valores para poder seleccionar UN ÚNICO valor. Los valores se indican dentro del control y la opción seleccionada por defecto se indica con **selected**

- Ejemplo

Color:



```
<SELECT NAME="color">  
<OPTION VALUE="rojo" SELECTED>Rojo</OPTION>  
<OPTION VALUE="verde">Verde</OPTION>  
<OPTION VALUE="azul">Azul</OPTION>  
</SELECT>
```

- Se obtienen los datos del control en PHP:

```
<?PHP  
$color = $_REQUEST['color'];  
echo $color;  
?>
```

SELECT MÚLTIPLE

El elemento **SELECT MÚLTIPLE** muestra un desplegable con valores para poder seleccionar VARIOS de modo que se almacenan en un array. La/s opción/es seleccionada/s por defecto se indican con **selected**. El atributo **Size** indica el número de valores a mostrar cada vez

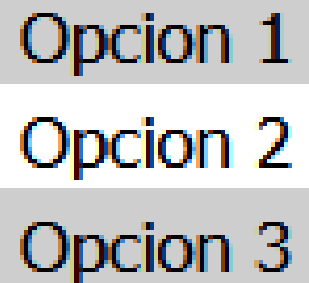
- Ejemplo

Idiomas:

```
<SELECT MULTIPLE SIZE="2" NAME="idiomas[]">
<OPTION VALUE="ingles" SELECTED>Inglés</OPTION>
<OPTION VALUE="frances" SELECTED>Francés</OPTION>
<OPTION VALUE="aleman">Alemán</OPTION>
</SELECT>
```

- Se obtienen los datos del control en PHP:

```
<?PHP
$idiomas = $_REQUEST['idiomas'];
foreach ($idiomas as $idioma)
echo "$idioma<br>\n";?>
```



INPUT BUTTON

El elemento INPUT de tipo **BUTTON** muestra un botón.

- Ejemplo

```
<INPUT TYPE="button" NAME="saludo"
ONCLICK="alert('Hola PHP') VALUE="saludo">
<INPUT TYPE="button" NAME="actualizar"
VALUE="Actualiza datos">
```




- Se obtienen los datos del control en PHP:

```
<?PHP
$saludo = $_REQUEST['saludo'];
if ($saludo) { //Si no es cadena vacía es true
    print ("Se ha visualizado el saludo");}
$actualizar = $_REQUEST['actualizar'];
if ($actualizar=="Actualiza datos") {
    print ("Se han actualizado los datos");}
?>
```

INPUT SUBMIT

El elemento INPUT de tipo **SUBMIT** muestra un botón para enviar los datos del formulario al destino indicado en el action del formulario.



- Ejemplo

```
<INPUT TYPE=submit NAME="enviar" VALUE="Enviar datos">
```

- Se obtienen los datos del control en PHP:

```
<?PHP
```

```
$enviar = $_REQUEST['enviar'];
```

```
if ($enviar=="Enviar datos") { //También funciona if($enviar)
```

```
    echo "Se ha pulsado el botón de enviar";
```

```
}
```

```
?>
```

INPUT SUBMIT - formaction

El elemento INPUT de tipo **SUBMIT** tiene atributos para poder modificar el comportamiento del formulario. Con **formaction** especifica una URL para su acción distinta a la que indica el formulario cuando se procesa, esto es, sobrescribe el atributo **action** del **form**

- Ejemplo

```
<form action="pagina1.php">
```

```
Nombre:<input type="text" name="nombre"><br>
```

```
Apellidos:<input type="text" name="apellidos"><br>
```

```
<input type="submit" value="Enviar">
```

```
<input type="submit" formaction="pagina2.php"
```

```
value="Enviar como Admin">
```

```
</form>
```

INPUT RESET

El elemento INPUT de tipo **RESET** muestra el formulario en blanco “limpiando” los controles.



- Ejemplo

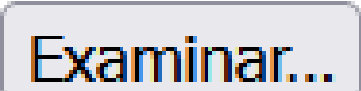
```
<INPUT TYPE=reset NAME="borrar" VALUE="Limpiar datos">
```

- Se obtienen los datos del control en PHP:

```
<?PHP
$borrar = $_REQUEST['borrar'];
if($borrar=="Limpiar datos") { //También if($borrar)
    echo "Se ha pulsado el botón de limpiar datos";
}
?>
```

INPUT FILE

El elemento INPUT de tipo **FILE** muestra un control para anexar un archivo de modo que podemos seleccionarlo y subirlo con el botón Examinar. Para poder subir archivos, el **formulario** debe tener **method="POST"** y **enctype="multipart/form-data"**

- Ejemplo:  No se ha seleccionado ningún archivo.

```
<FORM ACTION="procesa.php" METHOD="post"  
ENCTYPE="multipart/form-data">
```

```
<INPUT TYPE="file" NAME="fichero">
```

```
</FORM>
```

- Este tipo se detalla en la siguiente unidad con la subida de archivos al servidor

Servidor interno PHP

PHP-CLI facilita un servidor interno de PHP con fines de desarrollo y depuración.

Algunos parámetros interesantes son:

- t**; especifica la raíz de los documentos de forma explícita
- s**: Genera código HTML que muestra la sintaxis de ficheros PHP coloreada
- S**: Crea un servidor web que sirve/ejecuta código PHP

Si no se especifica un fichero, PHP busca el fichero `index.php` o `index.html` en su defecto

Servidor interno PHP

Para crear el servidor interno, nos situaremos en la carpeta donde tenemos almacenados los ficheros de PHP

En el terminal ejecutaremos:

php -S localhost:8000 //Puede ser otro puerto

En el navegador, introduciremos la url

127.0.0.1:8000 o bien **localhost:8000**

En el terminal podemos ver la comunicación entre cliente y servidor y posibles datos de depuración de la ejecución del código PHP