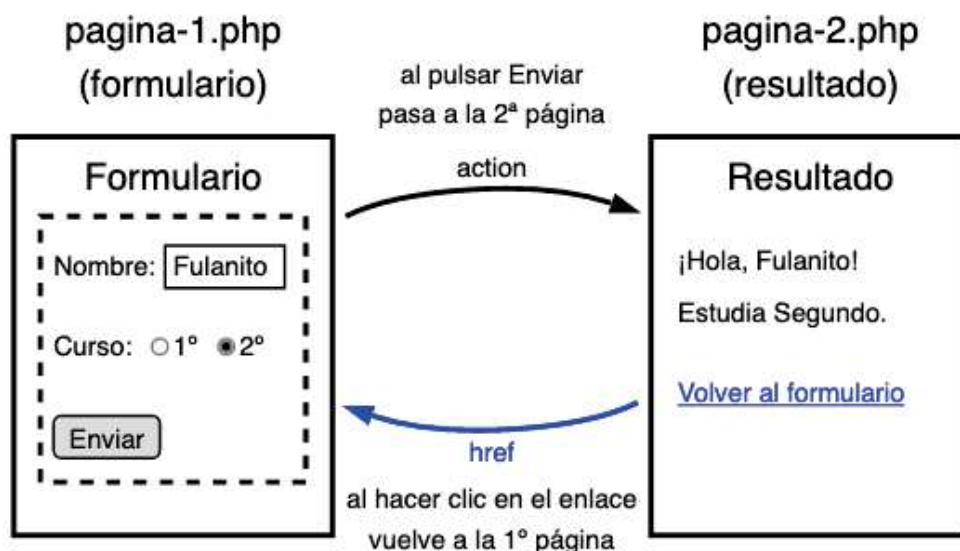


## 6 PHP y Formularios

### 6.1 Introducción a los formularios en PHP

Los formularios son los elementos de las páginas web que permiten a un usuario introducir información en una página web. Así, los formularios nos permiten crear páginas web PHP interactivas en las que el contenido de la página se adapta a la información proporcionada por el usuario.

En el caso más simple, se suelen distribuir las tareas en dos páginas independientes:



- Inicialmente, el navegador muestra la primera página, que contiene el formulario. El usuario completa los controles y pulsa el botón Enviar.
- Al pulsar el botón Enviar, el navegador muestra la segunda página. Al solicitar la segunda página al servidor, el navegador envía también la información introducida por el usuario en el formulario. El programa PHP que genera la segunda página utiliza la información para personalizar el contenido.
- Normalmente, la segunda página incluye un enlace que permite al usuario volver de nuevo a la primera página para empezar de nuevo el proceso.

El ejemplo siguiente corresponde a la imagen anterior (la página con el formulario se llama form-1-1.php y la página que recibe los datos se llama form-1-2.php):

**Formulario**

Nombre:

Curso: ☐ 1° ☒ 2°

Vamos a ver como construirlas y usarlas !!!

### 6.1.1 Creación de formularios

Las etiquetas de HTML que permiten la creación de formularios y de los diferentes tipos de controles (cajas de texto, casillas de verificación, botones radio, etc.) pertenecen a la creación de Páginas web HTML y hojas de estilo CSS que no forma parte de esta formación.

Este curso de PHP se centra en el tratamiento de los datos proporcionados por el usuario mediante el uso de formularios, no en el diseño de formularios. Por ello se recomienda tener un conocimiento general de las etiquetas y atributos relacionados con formularios.

Unas breves notas recordatorias, sobre las etiquetas y atributos más básicos.

#### 6.1.1.1 El formulario: `<form>`

Un formulario es un conjunto de controles (botones, cajas de texto, casillas de verificación, botones radio, etc) que permiten al usuario introducir datos y enviarlos al servidor web para su procesamiento.

```
<form action="ejemplo.php" method="get">
  <p>Nombre: <input type="text" name="nombre" size="40"></p>
  <p>Año de nacimiento: <input type="number" name="nacido" min="1900"></p>
  <p>Sexo:
    <input type="radio" name="hm" value="h"> Hombre
    <input type="radio" name="hm" value="m"> Mujer
  </p>
  <p>
    <input type="submit" value="Enviar">
    <input type="reset" value="Borrar">
  </p>
</form>
```

La etiqueta que delimita un formulario es la etiqueta `<form> ...</form>`.

Los atributos más importantes de la etiqueta `<form>` son:

- `action`: contiene el nombre del agente que procesará los datos remitidos al servidor (**por ejemplo, un script de PHP**)
- `method`: define la manera de enviar los datos al servidor. Los valores posibles son:
  - `get`: los valores enviados se añaden a la dirección indicada en el atributo `action`
  - `post`: los valores se envían de forma separada

Si el atributo `method` no está establecido, **el formulario se comporta como si el valor fuera `get`**.

La etiqueta `<form>` es un elemento de bloque. En su interior puede haber cualquier elemento típico de una página web (párrafos, imágenes, divisiones, listas, tablas, etc.), además de las etiquetas que crean los controles.

Los etiquetas que crean los controles en los formularios son `<input>`, `<button>`, `<select>`, `<optgroup>`, `<option>` y `<textarea>`.

Además, se pueden estructurar los controles con las etiquetas `<fieldset>` y `<legend>`.

Por último, la etiqueta `<label>` permite mejorar la accesibilidad de los controles.

El navegador envía únicamente los datos de los controles contenidos en el formulario. En una misma página puede haber varios formularios que envíen datos al mismo o a diferentes agentes.

Nombre:

Año de nacimiento:

Sexo: ☐ Hombre ☐ Mujer

### 6.1.1.2 Atributos comunes de los controles

Algunos atributos se pueden utilizar en varios tipos de controles.

#### El atributo type

El atributo type de la etiqueta <input> indica el tipo de control de que se trata.

En HTML 4.01 los tipos de <input> eran: botón para enviar submit,

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

botón para reiniciar reset,

```
<input type="reset" value="Borrar">
```

texto text, contraseña password, archivo file, imagen image y oculto hidden.

```
<input type="text" name="texto" placeholder="Escriba algo">
```

En HTML 5 se añadieron los tipos de <input>: botón button, número [number](#), búsqueda [search](#), teléfono [tel](#), dirección web [url](#), dirección de correo [email](#), fecha [date](#), hora [time](#), color [color](#) y rango [range](#).

En HTML 5.1 se añadieron los tipos de <input>: mes [month](#), semana week y fecha y hora local datetime-local.

#### El atributo name

El atributo name identifica al control.

Todos los controles de un formulario deben tener el atributo name porque, en general, sólo se envían los controles que lo tienen (aunque hay excepciones, como los controles de tipo imagen o los botones, que no los necesitan).

En general, los atributos name **no se deben repetir**, porque no se recibiría toda la información al enviarla. La única excepción es el botón radio.

## El atributo value

El atributo value permite establecer el valor inicial de un control, aunque cada control lo utiliza de una forma ligeramente distinta y algunos controles no lo admiten (área de texto, archivo o imagen).

En algunos casos (botones) el valor se muestra al usuario y el usuario no puede modificarlo. En otros casos (cajas de texto, contraseña) el valor se muestra en la página y el usuario puede modificarlo. En otros casos (casillas de verificación, botones radio, opciones de menú, oculto) el valor no se muestra en la página y el usuario no puede modificarlo.

En algunos casos (botones, oculto, botones radio) el atributo value es necesario. En otros casos (casilla de verificación, opciones de menú) es conveniente. En otros casos (cajas de texto, contraseña, imagen) puede omitirse. En el caso del selector de archivo, los navegadores no lo admiten por motivos de seguridad.

## El atributo required

El atributo required permite indicar qué controles es obligatorio rellenar para enviar el formulario.

```
<p>Nombre: <input type="text" name="nombre" required></p>
<p>Sexo:
  <input type="radio" name="hm" value="h" required> Hombre
  <input type="radio" name="hm" value="m" required> Mujer
</p>
```

Nombre:

Sexo: ☐ Hombre ☐ Mujer

## El atributo placeholder

El atributo placeholder permite mostrar en los controles de texto un texto que desaparece al escribir en el control.

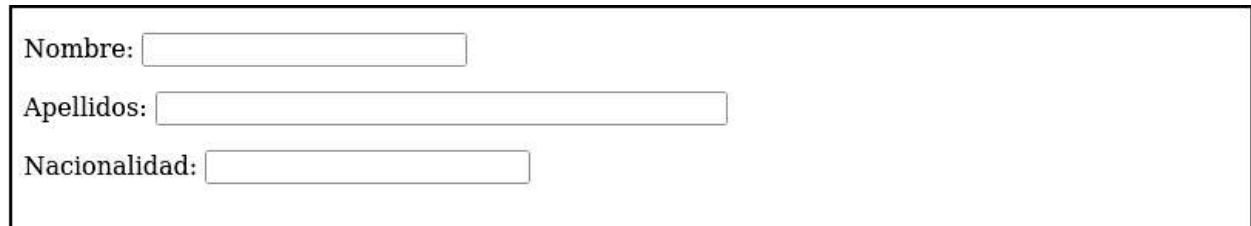
```
<p>Apellidos:
  <input type="text" name="apellidos" size="40" placeholder="Escriba sus dos
  apellidos">
</p>
```

Apellidos:

## El atributo size

El atributo size permite establecer la longitud de los controles de texto ([text](#), [password](#), [search](#), etc.). Por omisión, las cajas suelen tener 20 caracteres de longitud.

```
<p>Nombre: <input type="text" name="nombre"></p>  
<p>Apellidos: <input type="text" name="apellidos" size="40"></p>  
<p>Nacionalidad: <input type="text" name="nacionalidad" size="20"></p>
```



### 6.1.1.3 Los Controles

#### Controles:<input> y otros

Los controles son los elementos que permiten al usuario introducir información.

La mayoría de controles se obtienen mediante mediante la etiqueta vacía [<input>](#).

En HTML 4.01, los controles disponibles eran:

botón para enviar [<input type="submit">](#),

botón para reiniciar [<input type="reset">](#), botón [<button>](#), texto [<input type="text">](#), contraseña [<input type="password">](#), archivo [<input type="file">](#), imagen [<input type="image">](#), oculto [<input type="hidden">](#), área de texto [<textarea>](#) y menú [<select>](#).

En HTML 5 se añadieron los controles:

botón general [<input type="button">](#), número [<input type="number">](#), búsqueda [<input type="search">](#), teléfono [<input type="tel">](#), dirección web [<input type="url">](#), dirección de correo [<input type="email">](#), fecha [<input type="date">](#), hora [<input type="time">](#), color [<input type="color">](#), rango [<input type="range">](#), opciones [<option>](#) (para [<datalist>](#)), lista de valores [<datalist>](#), salida [<output>](#), barra de progreso [<progress>](#) y medidor [<meter>](#).

En HTML 5.1 se añadieron los controles:

mes [<input type="month">](#), semana [<input type="week">](#) y fecha y hora local [<input type="datetime-local">](#).

## Botones: <input> y <button>

Los botones se crean mediante la etiqueta <input> o mediante la etiqueta <button>. La diferencia entre ellos es que <input> sólo puede contener texto, mientras que <button> permite incluir elementos html como imágenes. El contenido del botón <input> se define mediante el atributo **value**, por lo que sólo puede contener texto. El contenido del botón <button> se escribe dentro del elemento, por lo que puede incluir texto e imágenes.

Normalmente los botones no suelen llevar atributo **name**, pero se puede poner si se quiere (tendría sentido si un formulario contuviera dos botones de envío y quisiéramos saber en cuál de ellos ha hecho clic el usuario, pero esto no es muy habitual).

### Botones Submit mediante <input> o <button>

El botón Submit es el que permite al usuario remitir los datos al servidor. Se crea mediante una etiqueta <input> o <button> cuyo atributo type tiene el valor submit.

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

Un formulario web que contiene un campo de texto con el placeholder "Escriba algo:" y un botón rectangular con el texto "Enviar".

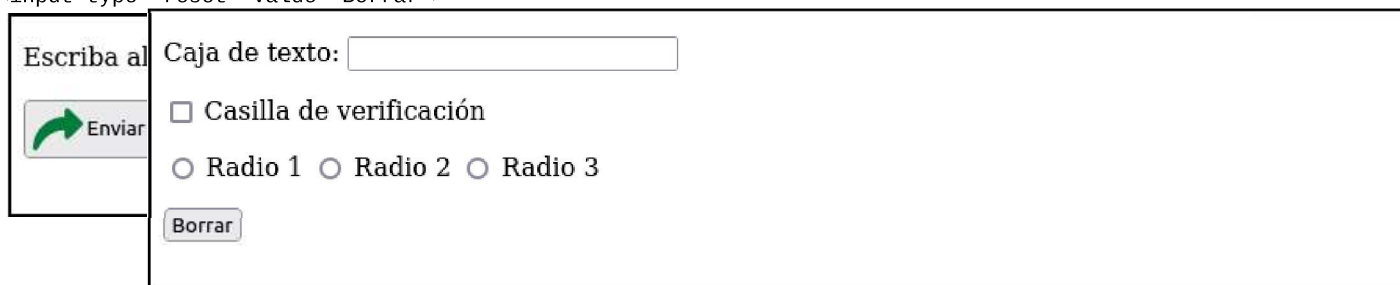
```
<button type="submit">
Enviar
</button>
```

Si un botón <button> no lleva el atributo type, se comporta como un botón de tipo submit.

### Botones Reset mediante <input> o <button>

El botón Reset restablece los valores iniciales del formulario. Se crea mediante una etiqueta <input> o <button> cuyo atributo type tiene el valor reset.

```
<input type="reset" value="Borrar">
```

Un formulario web que contiene un campo de texto con el placeholder "Escriba al", un campo de verificación con el texto "Caja de texto:", un grupo de tres botones de radio con los textos "Radio 1", "Radio 2" y "Radio 3", y un botón rectangular con el texto "Borrar".

```
<button type="reset">
Borrar
</button>
```

Para mas  
informacion  
sobre la  
forma de  
crear un  
formulario

Caja de texto:

☐ Casilla de verificación

☐ Radio 1 ☐ Radio 2 ☐ Radio 3



revisa el Anexo que se da a esta documentacion y que pertenece al curso de ASIR 1 Lenguaje de Marcas

## 6.1.2 Recepción de los datos

Cuando un formulario envía los datos a una página PHP, los datos están automáticamente a disposición del programa PHP en una matriz llamada **`$_REQUEST`**.

Cada dato recibido es un elemento de la matriz `$_REQUEST`: el índice del elemento es el nombre del control (salvo en el control de tipo imagen) y el valor del elemento es el dato introducido o seleccionado por el usuario en el control.

En el siguiente ejemplo, el programa que recibe los datos del formulario simplemente muestra el contenido de la matriz `$_REQUEST` utilizando la función `print_r()` que muestra los índices y valores de la matriz. La página con el formulario se llama `form-2-1.php` y la página que recibe los datos se llama `form-2-2.php`

```
<?php
// form-2-2.php
print "<pre>";
print_r($_REQUEST);
print "</pre>\n";
?>
```

### EJEMPLO DE F

Escriba su nombre:

Que curso Estudias: ☐ Asir 1 ☐ Asir 2

---

Practica Grupo ASIR - IES infanta Elena

```
Array
(
    [nombre] => pedro
    [grupo] => 1
)
```

[Volver al formulario.](#)

Como se puede comprobar en el ejemplo anterior, el control de caja de texto crea siempre un elemento en la matriz `$_REQUEST` (aunque la caja de texto se deje vacía), mientras que el botón radio solamente crea un elemento en la matriz `$_REQUEST` si se elige una de las opciones del botón radioe.

```
Array
(
    [nombre] => sss
)
```

[Volver al formulario.](#)