

UD-02e: Introducción a PHP

Desarrollo WEB en Entorno Servidor

Curso 2020/2021

Características básicas del lenguaje

Formularios web

- Los formularios HTML son la forma natural de hacer llegar al servidor web los datos del usuario desde un navegador.
- Van encerrados siempre entre las etiquetas `<form>` y `</form>`. Dentro de un formulario están los elementos sobre los que puede actuar el usuario, con etiquetas como `<input>`, `<select>`, `<textarea>` y `<button>`.
- El atributo `action` del elemento form indica la página a la que se le enviarán los datos del formulario. En nuestro caso, una página `.php`.

Características básicas del lenguaje

- Por su parte, el atributo **method** especifica el método usado para enviar la información. Puede tener dos valores:

- **get**: los datos del formulario se agregan al URL utilizando un signo de interrogación "?" como separador.

```
localhost/ejercicioCine2/cinePagina.php?Fila=3&Silla=5&Titulo=Alien&pele=Alien
```

- **post**: con este método los datos se incluyen en el cuerpo del formulario y se envían utilizando el protocolo HTTP.

```
localhost/ejercicioCine2/cinePagina.php
```

- Los datos se recogerán de distinta forma dependiendo de cómo se envíen.

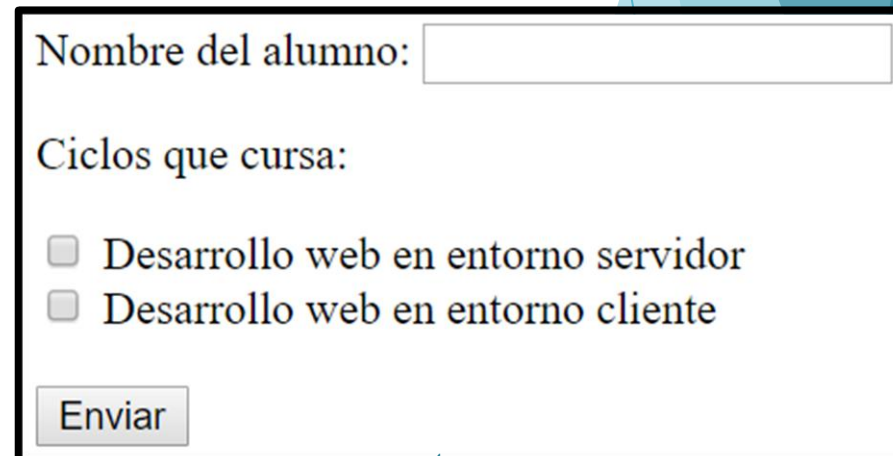
Ejemplo

```
<html>
  <head>
    <title>Formularios PHP</title>
  </head>
  <body>
    <form name="formulario" action="procesa.php" method="post">
      Nombre del alumno: <input type="text" name="nombre" /><br />
      <p>Módulos que cursa:</p>
      <input type="checkbox" name="modulos[]" value="DWES" />
        Desarrollo web en entorno servidor <br />
      <input type="checkbox" name="modulos[]" value="DWECE" />
        Desarrollo web en entorno cliente <br />
      <br />
      <input type="submit" value="Enviar" />
    </form>

  </body>
</html>
```

Usamos el method="post"

* Para enviar alguna variable en la que sea posible almacenar más de un valor, como es el caso de las casillas de verificación en este ejemplo (se pueden marcar varias a la vez), usa corchetes en el nombre de la variable para indicar que se trata de un array.



Nombre del alumno:

Ciclos que cursa:

☐ Desarrollo web en entorno servidor

☐ Desarrollo web en entorno cliente

Procesamiento de la información devuelta por un formulario

Archivo *procesa.php*

```
<html>
  <head>
    <title>Formularios PHP</title>
  </head>
  <body>
    <?php
      $nombre = $_POST['nombre'];
      $modulos = $_POST['modulos'];
      echo "Nombre: ".$nombre."<br />";
      foreach ($modulos as $mod)
      {
        echo "Modulo: ".$mod."<br />";
      }
    ?>
  </body>
</html>
```

Nombre del alumno:

Módulos que cursa:

- ☒ Desarrollo web en entorno servidor
- ☒ Desarrollo web en entorno cliente

Nombre: Bartolo
Modulo: DWES
Modulo: DWEC

Si hubiéramos usado GET en el method (en vez de POST):

```
<html>
  <head>
    <title>Formularios PHP</title>
  </head>
  <body>
    <?php
      $nombre = $_GET['nombre'];
      $modulos = $_GET['modulos'];
      echo "Nombre: ".$nombre."<br />";
      foreach ($modulos as $modulo)
      {
        echo "Modulo: ".$modulo."<br />";
      }
    ?>
  </body>
</html>
```

*Se podría haber usado `$_REQUEST`, que vale para ambas situaciones. No obstante, usaremos siempre `$_GET` y `$_POST`

Buenas prácticas del programador de PHP:

- ▶ Siempre que sea posible, es preferible validar los datos que se introducen en el navegador antes de enviarlos. Para ello deberás usar código en Javascript.
- ▶ En cualquier caso, tenemos que realizar las validaciones también en el servidor. Si no, por ejemplo, alguien puede crear un formulario propio idéntico pero que no realice las mismas validaciones en el navegador.
- ▶ Es muy habitual usar la misma página que muestra el formulario como destino de los datos. Si tras comprobar los datos éstos son correctos, se reenvía a otra página. Si son incorrectos, se rellenan los datos correctos en el formulario y se indican cuáles son incorrectos y el motivo.
- ❑ Para hacerlo de este modo, hay que comprobar si la página recibe datos o no (hay que mostrar el formulario con datos o no). Se puede comprobar usando la función isset con una variable que debe recibir (**por ejemplo, poniéndole un nombre al botón de enviar** y comprobando si esa variable tiene valor).
- ❑ En el siguiente código de ejemplo se muestra cómo hacerlo:

```

<body>
<?php
    if (isset($_POST['enviar']))
    {
        $nombre = $_POST['nombre'];
        $modulos = $_POST['modulos'];
        echo "Nombre: ".$nombre."<br />";
        foreach ($modulos as $modulo)
        {
            echo "Modulo: ".$modulo."<br />";
        }
    }
    else
    {
        ?>

```

Comprueba si se ha pulsado el botón **enviar** (si está definida la variable `$_POST['enviar']`, que ocurre si se envía el formulario)

Con cadena vacía, vuelve la página actual.

```

<form name="input" action="" method="post">
    Nombre del alumno: <input type="text" name="nombre" /><br />
    <p>Módulos que cursa:</p>
    <input type="checkbox" name="modulos[]" value="DWES" />
    Desarrollo web en entorno servidor <br />
    <input type="checkbox" name="modulos[]" value="DWECE" />
    Desarrollo web en entorno cliente <br />
    <br />
    <input type="submit" value="Enviar" name="enviar"/>
</form>

<?php
}
?>
</body>

```

El body de la página tiene “2 versiones”, según en qué situación esté.

Si el formulario no se ha enviado (el usuario acaba de llegar a la página), lo renderizamos. [código de abajo]

Si el formulario ya se ha enviado antes, haremos lo que corresponda.

En este caso, sólo echos [código de arriba]

(Observa que está todo dentro de `<?php` y `?>` menos el formulario, que es html)

Características básicas del lenguaje

Acceso desde *php* a los diferentes elementos de entrada

INPUT TEXT

HTML → nombre: <input type="text" name="cadena" value="valor por defecto" size="20">

nombre: ;

PHP →

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

(Si el formulario fuera GET, igual pero con **\$_GET**.
Y lo mismo para todos los tipos de elementos)

Características básicas del lenguaje

Acceso desde php a los diferentes elementos de entrada

INPUT RADIO

HTML → matricula:
`<input type="radio" name="matricula" value="or" checked>ordinaria`
`<input type="radio" name="matricula" value="li">libre`

matricula: ☒ ordinaria ☐ libre

PHP → `$matricula = $_POST['matricula'];`
`echo $matricula; // visualizará or o li`

Características básicas del lenguaje

Acceso desde php a los diferentes elementos de entrada

INPUT CHECKBOX

HTML → `<input type="checkbox" name="optativas[]" value="log" checked>lógica`
`<input type="checkbox" name="optativas[]" value="fra">francés`
`<input type="checkbox" name="optativas[]" value="inf">informática`

☒ lógica ☐ francés ☐ informática

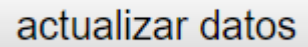
PHP → `$optativas = $_POST['optativas']; // devolverá un array.`
`foreach ($optativas as $opt)`
`echo $opt.'
';`

Características básicas del lenguaje

Acceso desde php a los diferentes elementos de entrada

INPUT BUTTON

HTML → `<input type="button" name="actualizar" value="actualizar datos">`



PHP →

```
<?php
    if (isset($_POST['actualizar']))
        echo 'se ha pulsado el botón de actualizar';
?>
```

Características básicas del lenguaje

Acceso desde php a los diferentes elementos de entrada

INPUT PASSWORD

HTML → contraseña: <input type="password" name="clave">

contraseña:

PHP →

```
<?php
    $clave = $_POST['clave'];
    echo $clave;
?>
```

Características básicas del lenguaje

Acceso desde php a los diferentes elementos de entrada

INPUT SUBMIT

HTML → `<input type="submit" name="enviar" value="enviar datos">`

enviar datos

PHP →

```
<?php
    if (isset($_POST['enviar']))
        echo 'se ha pulsado el botón de enviar';
?>
```

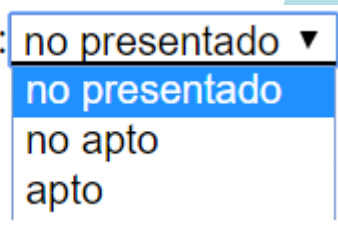
Características básicas del lenguaje

Acceso desde php a los diferentes elementos de entrada

SELECT SIMPLE

HTML → calificación: `<select name="calificacion">`
 `<option value="nopresentado" selected>no presentado</option>`
 `<option value="noapto">no apto</option>`
 `<option value="apto">apto</option>`
 `</select>`

PHP → `<?php`
 `$calificacion = $_POST['calificacion'];`
 `echo $calificacion;`
 `?>`

calificación: 

Características básicas del lenguaje

Acceso desde php a los diferentes elementos de entrada

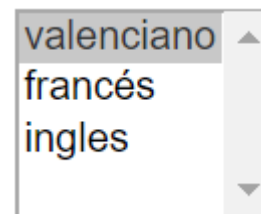
SELECT MÚLTIPLE

HTML →

```
<select multiple size="4" name="idiomas[]">
  <option value="valenciano" selected>valenciano</option>
  <option value="frances">francés</option>
  <option value="ingles">ingles</option>
</select>
```

PHP →

```
<?php
  $idiomas = $_POST['idiomas'];
  foreach ($idiomas as $idioma)
    echo '$idioma<br>';
?>
```



(Se pueden marcar varios con Ctrl)

Características básicas del lenguaje

Acceso desde php a los diferentes elementos de entrada

TEXTAREA

HTML → comentario: `<textarea cols="30" rows="3" name="comentario">`
este libro me parece ...
`</textarea>`

comentario:

PHP → `<?php`
 `$comentario = $_POST['comentario'];`
 `echo $comentario;`
`?>`

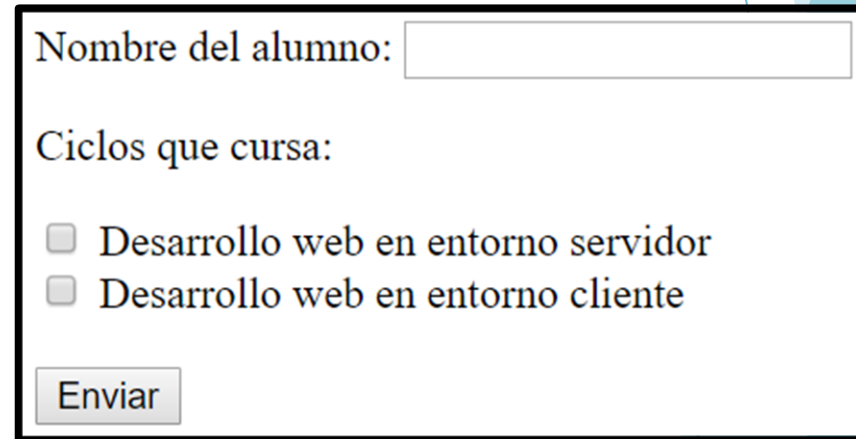
Características básicas del lenguaje

Procesamiento de formularios en PHP

Ejemplo con **restricciones**:

Tenemos que comprobar que el **nombre no esté vacío** y que se haya seleccionado **como mínimo uno de los módulos**.

Además, si falta algún dato, debemos generar el formulario rellenando aquellos datos que el usuario haya introducido correctamente.



Nombre del alumno:

Ciclos que cursa:

☐ Desarrollo web en entorno servidor

☐ Desarrollo web en entorno cliente

```
if (!empty($_POST['modulos']) && !empty($_POST['nombre']))  
{  
    // Aquí iría el código a ejecutar cuando los datos son correctos  
}  
  
else  
{  
    // Aquí generamos el formulario, indicando los datos incorrectos  
    // y rellenando los valores correctamente introducidos  
}
```

Si los datos NO son correctos:

Mostramos sólo los datos correctamente introducidos en el formulario.

Para ello se utiliza el atributo **value** en las entradas de texto...

```
Nombre del alumno:<input type="text" name="nombre"
    value="<?php if(isset($_POST['nombre'])) echo $_POST['nombre']; ?>" />
```

... y el atributo **checked** en las casillas de verificación:

```
<input type="checkbox" name="modulos[]" value="DWES"
<?php if(in_array("DWES", $_POST['modulos'])) echo "checked"; ?> />
```

Para indicar al usuario los datos que no ha rellenado (o que ha rellenado mal), se debe comprobar si es la primera vez que se visualiza el formulario o no.

```
Nombre del alumno: <input type="text" name="nombre"
value="<?php if(isset($_POST['nombre'])) echo $_POST['nombre']; ?>" />
<?php
if (isset($_POST['enviar']) && empty($_POST['nombre']))
    echo "<span style='{color:red}'> Debe introducir un nombre </span>";
?>
<br />
```

```
<body>
```

```
<?php
if (!empty($_POST['modulos']) && !empty($_POST['nombre']))
{
    $nombre = $_POST['nombre'];
    $modulos = $_POST['modulos'];
    echo "Nombre: ".$nombre."<br />";
    foreach ($modulos as $modulo)
    {
        echo "Modulo: ".$modulo."<br />";
    }
}
else
{
```

```
?>
<form name="input" action="" method="post">
Nombre del alumno: <input type="text" name="nombre" value="<?php if(isset($_POST['nombre'])) echo $_POST['nombre'];?>" />
<?php
if (isset($_POST['enviar']) && empty($_POST['nombre']))
echo "<span style='color:red'> Debe introducir un nombre!!</span>" ?> <br />
<p>Módulos que cursa:
<?php
if (isset($_POST['enviar']) && empty($_POST['modulos']))
echo "<span style='color:red'> Debe escoger al menos uno!!</span>"
?>
</p>
<input type="checkbox" name="modulos[]" value="DWES"
<?php
if(isset($_POST['modulos']) && in_array("DWES",$_POST['modulos']))
echo 'checked="checked"';
?> />
Desarrollo web en entorno servidor <br />
<input type="checkbox" name="modulos[]" value="DWECE"
<?php
if(isset($_POST['modulos']) && in_array("DWECE",$_POST['modulos']))
echo 'checked="checked"';
?> />
Desarrollo web en entorno cliente<br /><br />
<input type="submit" value="Enviar" name="enviar"/>
</form>
<?php
```

```
}
?>
```

```
</body>
```

Aquí también podría ponerse
`if (isset($_POST['modulos']) && isset($_POST['nombre']))`

Esta parte de arriba comprueba si se ha enviado ya el formulario con ambos campos rellenos. No consultamos si se ha pulsado el botón de enviar. Sólo comprobamos los POST de los campos (eso sí, no se generarán hasta que no pulsamos el botón de Enviar). Si ambos POST no están vacíos, que es lo que queremos, acabamos con un echo. Si no, iríamos al *else*. Así que ojo, aunque este fragmento de código esté situado arriba, es “lo último” en ocurrir.

Las líneas moradas sirven para mostrar un mensaje avisando de que un campo del formulario se ha dejado en blanco. Lo que comprueba es: ¿se ha pulsado el botón de enviar y está vacío el POST de dicho campo? Si la respuesta es sí → Añadimos un texto en rojo con el aviso.

Las líneas azules sirven para conservar lo que había escrito en un campo en caso de que otro campo del formulario haya fallado. *Por ejemplo: si escribo el nombre Bartolo, pero no marco asignatura, al enviar se me avisa de que elija asignatura, pero Bartolo se conserva.* Lo que comprueba es: ¿existe el POST de nombre? Si la respuesta es sí, lo pongo como value y ya saldrá escrito en el input.

En general, este bloque de líneas de abajo se encarga de mostrar el formulario, tanto la 1ª vez como todas las que hagan falta después si se dejó algún campo vacío. Las líneas moradas muestran avisos, y las azules claro conservan la parte que está bien.

Nombre del alumno:

Módulos que cursa:

- ☐ Desarrollo web en entorno servidor
- ☐ Desarrollo web en entorno cliente

Enviar

Nombre del alumno: <-- Debe introducir un nombre!!

Módulos que cursa: <-- Debe escoger al menos uno!!

- ☐ Desarrollo web en entorno servidor
- ☐ Desarrollo web en entorno cliente

Enviar

Nombre del alumno:

Módulos que cursa:

- ☐ Desarrollo web en entorno servidor
- ☐ Desarrollo web en entorno cliente

Enviar

Nombre del alumno:

Módulos que cursa: <-- Debe escoger al menos uno!!

- ☐ Desarrollo web en entorno servidor
- ☐ Desarrollo web en entorno cliente

Enviar

Nombre del alumno:

Módulos que cursa:

- ☒ Desarrollo web en entorno servidor
- ☒ Desarrollo web en entorno cliente

Enviar

Nombre: Anacleto
Modulo: DWES
Modulo: DWEC



EJERCICIOS

❖ Crea una página llamada ***ejer11.php*** cuyo título sea ***Ejercicio11***

Invéntate un formulario (que tenga sentido) en el que haya:

- 1 input text
- 1 select simple con 3 opciones
- 2 input radio
- 1 input submit

Sin comprobar nada, cuando se pulse el botón del formulario éste **se enviará por GET** a otra página diferente llamada **mostrar.php**

En ella se hará un echo de toda la información recibida y de forma organizada.
(como no comprobamos los campos, podría dar error en algún caso)

FORMULARIO INVENTADO

Nombre:

Elige tu kebab: ▼

¿Salsa? ☐ con ☐ sin

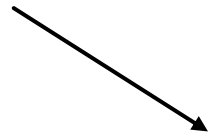


FORMULARIO INVENTADO

Nombre:

Elige tu kebab: ▼

¿Salsa? ☒ con ☐ sin



Nombre: Pascu
Tu kebab elegido: mixto
¿Salsa? con

EJERCICIOS



❖ Crea una página llamada ***ejer12.php*** cuyo título sea *Ejercicio12*

Invéntate ahora un formulario que contenga:

- 1 input text (inicialmente vacío)
- Una pregunta y 3 input checkbox para poder marcar
(inicialmente todos desmarcados, y se pueden marcar varios)
- 1 text area (inicialmente vacío)
- 1 input submit

Esta vez formulario se envía **a la misma página** por POST, y si algún campo está vacío (en el caso de los checkbox, si no se ha marcado ninguno de los 3) se deben mostrar **mensajes de error** (**conservando su valor** los que sí estaban rellenos).

Si todos los campos se han relleno, se deben mostrar por pantalla.

OTRO FORMULARIO INVENTADO

Nombre:

¿A qué rey o reyes va dirigida?

- ☐ Melchor
- ☐ Gaspar
- ☐ Baltasar

Escribe tu carta para los Reyes Magos

Enviar

OTRO FORMULARIO INVENTADO

Nombre: ¡Debes introducir tu nombre!

¿A qué rey o reyes va dirigida?

- ☐ Melchor
- ☐ Gaspar
- ☐ Baltasar ¡Marca al menos un rey!

Escribe tu carta para los Reyes Magos

¡Debes escribir algo!

Enviar

OTRO FORMULARIO INVENTADO

Nombre:

¿A qué rey o reyes va dirigida?

- ☐ Melchor
- ☐ Gaspar
- ☐ Baltasar

Escribe tu carta para los Reyes Magos

Quiero que me toque la lotería

Enviar

OTRO FORMULARIO INVENTADO

Nombre:

¿A qué rey o reyes va dirigida?

- ☐ Melchor
- ☐ Gaspar
- ☐ Baltasar ¡Marca al menos un rey!

Escribe tu carta para los Reyes Magos

Quiero que me toque la lotería

Enviar

OTRO FORMULARIO INVENTADO

Nombre:

¿A qué rey o reyes va dirigida?

- ☒ Melchor
- ☐ Gaspar
- ☒ Baltasar

Escribe tu carta para los Reyes Magos

Quiero que me toque la lotería

Enviar



Nombre: Pascu

Reyes a los que va dirigida:

- melchor
- baltasar

Tu carta:

Quiero que me toque la lotería

EJERCICIOS



- ❖ Realiza la gestión de datos de un formulario simple mediante un script auto-referenciado con nombre ***ejer13.php*** cuyo título sea ***Ejercicio13***, que muestre el formulario de la página siguiente y su respuesta después de recoger datos.
- Un campo del texto para introducir el Nombre (*Obligatorio, ponle asterisco).
- Una lista multi-selección para tipos de música preferidos. Los valores disponibles son: ***Acústica, BSO, R&B, Electrónica, Folk, Jazz, Pop, Rock***. (Obligatorio al menos uno).
- Debe mostrarse el nombre de la máquina cliente y su dirección IP (*debes usar la ayuda en pág.30*), además de la fecha y hora en formato ***dd/mm/aaaa hh:mm:ss***
- Después de mostrar los datos introducidos, se nos mostrará también un enlace para que el usuario pueda volver al formulario original.

Ayuda para obtener la IP del cliente:

- Hagas lo que hagas, asegúrate de no confiar en los datos enviados por el cliente.
- `$_SERVER['REMOTE_ADDR']` contiene la dirección IP real de la parte que se conecta.
- Sin embargo, pueden estar detrás de un servidor proxy, en cuyo caso el proxy puede haber configurado `$_SERVER['HTTP_X_FORWARDED_FOR']`, pero este valor se falsifica fácilmente. Por ejemplo, puede ser configurado por alguien sin un proxy, o la IP puede ser una IP interna de la LAN detrás del proxy.
- Esto significa que es muy recomendable guardar ambas informaciones.
- Aquí se proporciona una función que puedes usar y que te ayudará a resolver este asunto.

```
function getIPCliente () {  
    if (isset($_SERVER['HTTP_CLIENT_IP']))  
        $ip = $_SERVER['HTTP_CLIENT_IP'];  
    elseif (isset($_SERVER['HTTP_X_FORWARDED_FOR']))  
        $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];  
    elseif (isset($_SERVER['HTTP_X_FORWARDED']))  
        $ip = $_SERVER['HTTP_X_FORWARDED'];  
    elseif (isset($_SERVER['HTTP_FORWARDED_FOR']))  
        $ip = $_SERVER['HTTP_FORWARDED_FOR'];  
    elseif (isset($_SERVER['HTTP_FORWARDED']))  
        $ip = $_SERVER['HTTP_FORWARDED'];  
    elseif (isset($_SERVER['REMOTE_ADDR']))  
        $ip = $_SERVER['REMOTE_ADDR'];  
    else  
        $ip = 'Origen desconocido.';  
    return $ip;  
}
```