

Bibliografía:

Implantación de Aplicaciones Web, Jorge Sánchez Asenjo
PHP con acceso a bases de datos, Félix Mateos.

TEMA 4. GESTIÓN DE FORMULARIOS HTML CON PHP

Un formulario HTML se define con el elemento `<form>`, cuyos atributos principales son:

- **action**: Dónde se van a enviar los datos recogidos por el formulario
- **method**: Para indicar si los datos se van a enviar por el método **POST** (en la cabecera del paquete http) o **GET** (en la url). No tenemos que pasar por GET los datos confidenciales. Por defecto el método es GET.
- **enctype**: No se suele poner. Indica cómo se van a codificar los datos recogidos por el formulario cuando el método de envío elegido es POST:
 - `application/x-www-form-urlencoded`: Es el valor predeterminado
 - `multipart/form-data`: Se utiliza cuando el formulario contiene archivos
 - `text/plain`: Se utilizaba sobre todo cuando el destino del formulario era una dirección de correo electrónico (`action=mailto:correoejemplo@gmail.com`), aunque ahora está en desuso porque hay otros métodos para enviar los datos de un formulario a un correo.

Dentro del formulario podemos crear diferentes controles:

- `<input type='text'>`: Campos de texto
- `<input type='password'>`: Campos de contraseña
- `<input type='radio'>`: Botones de opción
- `<input type='checkbox'>`: Casillas de activación
- `<input type='hidden'>`: Campos de texto ocultos
- `<input type='file'>`: Para adjuntar archivos
- `<input type='submit'>`: Botón para enviar el formulario
- `<input type='reset'>`: Borra los datos del formulario
- `<select><option>...</option></select>`: Cuadros de lista
- `<textarea>`: Cuadros de texto

El atributo **name** de cada uno de estos tipos de controles es el que envía el formulario junto con el dato recogido por el campo.

Estos datos pueden recogerse en PHP mediante las variables superglobales **\$_POST** o **\$_GET**, según cuál haya sido el método de envío elegido. Con la variable **\$_REQUEST** se pueden recoger los datos sea cual sea el método de envío elegido.

Por ejemplo, el siguiente formulario:

```

001 <html>
002   <head>
003     <title>Formulario</title>
004   </head>
005   <body>
006     <form action='procesar.php' method='post'>
007       <label for='id_nombre'>Nombre:</label>
008       <input type='text' name='nombre' id='id_nombre' /><br /><br />
009       <label for='id_clave'>Clave:</label>
010       <input type='password' name='clave' id='id_clave' /><br /><br />
011       Sexo:<br />
012       <label for='id_hombre'>Hombre:</label>
013       <input type='radio' name='sexo' id='id_hombre' value='hombre' /><br />
014       <label for='id_mujer'>Mujer:</label>
015       <input type='radio' name='sexo' id='id_mujer' value='mujer' /><br /><br />
016       Aficiones:<br />
017       <label for='id_musica'>Música</label>
018       <input type='checkbox' name='aficiones[]' id='id_musica'
value='musica' /><br />
019       <label for='id_deporte'>Deporte</label>
020       <input type='checkbox' name='aficiones[]' id='id_deporte'
value='deporte' /><br />
021       <input type='submit' value='Enviar' />
022     </form>
023   </body>
024 </html>
025 <html>

```

Podría procesarse con este sencillo script PHP:

```

001 <?php
002   echo $_POST['nombre'].'<br />';
003   echo $_POST['clave'].'<br />';
004   echo $_POST['sexo'].'<br />';
005   print_r( $_POST['aficiones']);
006 ?>

```

Adquisición de datos enviados mediante formularios HTML

Para acceder desde PHP a los datos recibidos mediante un formulario tendremos que recurrir a las superglobales `$_POST`, `$_GET` o `$_REQUEST`. Estas superglobales son arrays asociativos cuyos índices coinciden con los valores de los atributos name de los controles de los formularios.

El tamaño de los datos que se pueden enviar a través de peticiones GET suele menor que el del método POST. La limitación para el método GET puede encontrarse en el navegador del cliente, que trunque la URL a 255 caracteres, o en el lado del servidor Apache, donde el tamaño máximo está controlado por la directiva LimitRequestLine del `httpd.conf`.

En el caso del método POST, la directiva post_max_size del `php.ini` determina el tamaño máximo de los datos que pueden enviarse. Pero esta directiva está supeditada al valor establecido en la directiva LimitRequestBody del `httpd.conf` de Apache.

Si un script PHP recibe variables con el mismo nombre a través de varias vías simultáneamente (GET, POST o Cookies) se procesarán según el orden establecido en la directiva `request_order` del `php.ini`, de modo que la última que aparezca en esta directiva será la que prevalecerá. Por ejemplo, si un script recibe una variable llamada clave a través de GET y otra con el mismo nombre a través del método POST, y la directiva está configurada como `request_order='PG'`, prevalecerá la recibida a través del método GET.

Los controles que admiten varios valores, como un conjunto de casillas de activación debe utilizar un name en el HTML terminado en corchetes [] para que PHP los reciba como un array.

```
formulario.html
001  <html>
002    <head>
003      <title>Formulario</title>
004    </head>
005    <body>
006      <form action='procesar.php' method='post'>
007        <select name='aficiones[]' size='5' multiple='multiple'>
008          <option value='lectura'>Lectura</option>
009          <option value='musica'>Música</option>
010          <option value='deporte'>Deporte</option>
011          <option value='informatica'>Informática</option>
012          <option value='electronica'>Electrónica</option>
013        </select>
014        <br />
015        <input type='checkbox' name='idiomas[]' value='ingles' />Inglés
016        <br />
017        <input type='checkbox' name='idiomas[]' value='frances' />Francés
018        <br />
019        <input type='checkbox' name='idiomas[]' value='italiano' />Italiano
020        <br />
021        <input type='submit' />
022      </form>
023    </body>
024  </html>

procesar.php
001  <?php
002    var_dump($_POST);
003  ?>
```

Comprobar que la variable está asignada antes de utilizarla

Un error frecuente es dar por hecho que vamos a recibir en nuestro script PHP todas las variables que esperamos y empezar a operar con ellas directamente. Esto producirá un error si alguna de esas variables no estuviera definida.

Para evitarlo, es conveniente recoger los valores de `$_POST` y `$_GET` en variables globales del sistema, comprobando previamente que realmente están asignadas. Por ejemplo:

```
001 <?php
002     if(isset($_GET['identificativo'])) {
003         $identificativo=$_GET['identificativo'];
004     } else {
005         die('Credenciales incompletas');
006     }
007     if(isset($_GET['clave'])) {
008         $clave=$_GET['clave'];
009     } else {
010         die('Credenciales incompletas');
011     }
012     if($identificativo=='juanfe' && $clave=='mateos') {
013         echo 'Bienvenido superusuario';
014     } else {
015         echo 'Acceso no autorizado';
016     }
017 ?>
```

Redirigir al cliente a otra dirección

Para redirigir al cliente a otra dirección utilizamos la cabecera Location.

```
header('Location: página.php');
```

Se pueden pasar parámetros

```
header('Location: errores.php?tipo=1');
```

Mostrar un mensaje y redirigir al cliente a otra dirección

La cabecera Refresh, nos permite establecer un plazo en segundos antes de redirigir al navegador a otra dirección. Por ejemplo:

```
header('Refresh: 5; url=http://www.google.es');
echo 'Lo que busca no existe, le redirigiremos a Google en 5 segundos'
```

Autollamada de páginas

Se pueden enviar y recibir datos de un formulario con una única página. A este efecto podemos llamarle "autollamada de páginas", también se le suele llamar como "Formularios reentrantes".

En ambos casos, para formularios o envío de datos por la URL, se debe seguir el siguiente esquema:

- Comprobar si recibo datos por URL o por formulario
- Si no recibo datos → Muestro el formulario o los enlaces que pasan variables.
- Si recibo datos → Tengo que procesar los datos del formulario.

Veamos a continuación como sería el código de un formulario reentrante llamado autollamada.php.

```
<html>
<head>
    <title>Me llamo a mi mismo...</title>
</head>

<body>
<?
if (!isset($_POST['enviar'])){
?><form action="autollamada.php" method="post">
    Nombre:<input type="text" name="nombre" size="30"> <br>
    Empresa:<input type="text" name="empresa" size="30"> <br>
    Telefono:<input type="text" name="telefono" size=14 value="+34">
<br>
    <input type="submit" name="enviar" value="Enviar">
</form>

<?
}else{
    echo "<br>Su nombre: " . $_POST["nombre"];
    echo "<br>Su empresa: " . $_POST["empresa"];
    echo "<br>Su Teléfono: " . $_POST["telefono"];
}
?>
</body>
</html>
```