

TRABAJO ENTRE PÁGINAS Y SESIONES

ÍNDICE

MOTIVACIÓN.....	3
PROPÓSITOS.....	4
PREPARACIÓN PARA LA UNIDAD	5
1. LOS OBJETOS “\$_GET” Y “\$_POST”	6
CONCLUSIONES	9
RECAPITULACIÓN	10
BIBLIOGRAFÍA.....	11

MOTIVACIÓN

La principal característica de los lenguajes de programación como el PHP es la posibilidad de trabajar con datos de formularios, es decir, poder interactuar con el usuario para solicitar y servir información.

Gracias a esta característica, el PHP no se usa solamente en páginas web corporativas, sino que también se utiliza para la creación de CRM o intranets de empresas, ya que funciona sobre cualquier ordenador conectado a la red y tiene la potencia de cualquier herramienta de *software*.

PROPÓSITOS

Al finalizar esta sección serás capaz de:

- Interactuar con los formularios de HTML y datos pasados por URL.

PREPARACIÓN PARA LA UNIDAD

En esta unidad vas a trabajar con datos enviados desde formularios, por ello es conveniente que tengas un nivel bueno de trabajo con HTML y formularios de HTML.

1. LOS OBJETOS “\$_GET” Y “\$_POST”

Estos objetos son dos variables del método HTTP:

En el caso de **\$_GET**, se trata de un vector que contiene los elementos pasados por URL a la página, es decir, de la ruta:

- `www.miweb.com/listado.php?pagina=1&idioma=es`.

El vector **\$_GET** contendría dos elementos, uno con clave **‘pagina’** y valor **‘1’**, y otro con clave **‘idioma’** y valor **‘es’**.

La diferencia con **\$_POST** es que los valores del vector que contiene **\$_POST** no son pasados por URL (ni siquiera son visibles al usuario), sino que son pasados por formulario, y son cada uno de los campos o elementos del formulario que llamó a la web. Por ejemplo, del siguiente formulario:

```
<form name="prueba" method="post" action="listado.php">
<input type="text" name="nombre"/>
<input type="checkbox" name="presente"/>
<input type="submit" name="enviar"/>
</form>
```

Los elementos que contendrá el vector **\$_POST** son: el primero clave **‘nombre’** y valor lo que el usuario escribiera en el campo; el segundo clave **‘presente’** y valor **‘on’** en caso de que el usuario lo marque, o vacío en caso contrario.

Un formulario puede crearse con el método **‘get’** en vez de **‘post’**. En ese caso, deberíamos recoger los valores con el vector **\$_GET**.

Los datos recibidos en estos vectores están ya tratados con la función **urlencode** y no es necesario pasarlos por ella. De hecho, si los pasamos cambiará el contenido de los vectores por otro erróneo.

En PHP existe un segundo método para poder recoger los valores que recibimos en la web, ya sea mediante método POST o método GET. Este método consiste en usar la variable genérica **\$_REQUEST**, que contiene un vector asociativo con los valores de **\$_GET**, **\$_POST** y **\$COOKIE** (este último lo vemos más adelante).

Este ejemplo consta de dos páginas: la primera, un HTML que contiene un formulario con varios campos al que llamaremos **formulario.html** (puedes consultar este ejemplo en el archivo adjunto: "formulario.html"); y un segundo archivo PHP, que usaremos para leer los datos enviados mediante el formulario de la página anterior.

■ formulario.html.

```
<html>
<body>
<p>
<form method="post" action="ejemplo19.php" name="form">
Nombre <input type="text" name="nombre" /><br />
Apellidos <input type="text" name="apellidos" /><br />
Edad <input type="text" name="edad" /><br />
Color preferido <select name="color" >
<option value="rojo">rojo</option>
<option value="verde">verde</option>
</select>
```

```
<br />
Me gusta el cine <input type="checkbox"
name="cine" /><br />
Descripción <textarea cols="20" rows="3"
name="descripcion"></textarea><br />
<input type="submit" value="enviar"/>
</form>
</p>
</body>
</html>
```

■ ejemplo.php.

```
<html>

<body><p>

<?php
echo 'Bienvenido '.$_POST['nombre'].' '.$_POST['apellidos']
    . '<br/>';

echo 'Tu edad es '.$_POST['edad'].' años.<br/>';
echo 'Tu color favorito es el'.$_REQUEST['color'].'<br/>';

echo ' y te describes como '.$_REQUEST['descripcion']
?>

</p></body>

</html>
```

PHP también nos permite el trabajo con parámetros complejos como los **‘select’** múltiples, en cuyo caso recibiríamos un vector con los valores seleccionados:

```
<?php
echo '<pre>';
echo htmlspecialchars(print_r($_POST, true));
echo '</pre>';

?>

<form action="" method="post">
Nombre: <input type="text" name="nombre" /><br />
Email: <input type="text" name="email" /><br />
Colores favoritos: <br />
<select multiple name="colores[]">
<option value="rojo">rojo</option>
<option value="verde">verde</option>
<option value="amarillo">amarillo</option>
</select><br />
<input type="submit" value="mandar" />

</form>
```

En el ejemplo anterior usamos una función llamada **‘htmlspecialchars’**, que convierte los caracteres especiales (como el *ampersand*, las comillas...) a sus respectivas entidades HTML.

CONCLUSIONES

El trabajo con formularios es fundamental para desarrollar programas de utilidad para el usuario o para poder trabajar con los mismos datos entre pantallas. Para realizar este trabajo, pasar datos entre pantallas, también tenemos las variables de sesión, con las cuales podemos contener valores durante toda la ejecución de nuestra web.

RECAPITULACIÓN

En esta unidad has aprendido:

- El uso de la variable global **\$_GET**.
- El uso de la variable global **\$_POST**.
- El uso de la variable global **\$_REQUEST**.

BIBLIOGRAFÍA

- CABEZAS, L. M. *PHP 5*. Madrid: Anaya Multimedia, 2010.
- Manual de la página web oficial de PHP:
<http://www.php.net/manual/es/book.session.php>.