INTRODUCCIÓN AL MÓDULO

28 Septiembre 2022

Educar en la Verdad
para ser libres

CAPÍTULO 1: Inserción Código se Servidor

2.1. Ámbito de ejecución

2.2. Lenguajes de servidor

2.2.1. *PHP*

2.2.2. *JSP*

2.2.3. *PHP*

2.3. Etiquetas de Servidor

2.3.1 *Etiquetas PHP*

2.3.2 *Etiquetas JSP*

2.3.3 *Etiquetas ASP.NET*

2.4. Generación del código en el cliente

2.4. *Transformación del código de servidor*

Módulo 3: Inserción Código Servidor

28 Septiembre 2022

Educar en la Verdad
para ser libres

3.1. Elementos lenguaje PHP

En la unidad anterior, aprendiste a preparar un entorno para programar en PHP. Además también viste algunos de los elementos que se usan en el lenguaje, como las variables y tipos de datos, comentarios, operadores y expresiones.

También sabes ya cómo se integran las etiquetas HTML con el código del lenguaje, utilizando los delimitadores .

En esta unidad aprenderás a utilizar otros elementos del lenguaje que te permitan crear programas completos en PHP. Los programas escritos en PHP, además encontrarse estructurados normalmente en varias páginas (ya veremos más adelante cómo se pueden comunicar datos de unas páginas a otras), suelen incluir en una misma página varios bloques de código. Cada bloque de código debe ir entre delimitadores, y en caso de que genere alguna salida, ésta se introduce en el código HTML en el mismo punto en el que figuran las instrucciones en PHP.

Educar en la Verdad para ser libres

3.1. Elementos lenguaje PHP

Por ejemplo, en las siguientes líneas tenemos dos bloques de código en PHP:

```
<body>
<?php $a=1; ?>
<p>Página de prueba</p>
<?php $b=$a; ?>
```

Aunque no se utilice el valor de las variables, en el segundo bloque de código la variable ($a) mantiene el valor 1 que se le ha asignado anteriormente.

Educar en la Verdad para ser libres

3.2. Formularios WEB

La forma natural para hacer llegar a la aplicación web los datos del usuario desde un navegador, es utilizar formularios HTML.

Los formularios HTML van encerrados siempre entre las etiquetas <FORM> </FORM>. Dentro de un formulario se incluyen los elementos sobre los que puede actuar el usuario, principalmente usando las etiquetas <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 se tratará de un guión PHP.

Por su parte, el atributo method especifica el método usado para enviar la información. Este atributo puede tener dos valores:

- get: con este método los datos del formulario se agregan al URI utilizando un signo de interrogación "?" como separador.
- post: con este método los datos se incluyen en el cuerpo del formulario y se envían utilizando el protocolo HTTP.

Como vamos a ver, los datos se recogerán de distinta forma dependiendo de cómo se envíen.

3.2. Formularios WEB

Ejercicio: Crea un formulario HTML para introducir el nombre del alumno y el ciclo que cursa, a escoger entre “Desarrollo Web en Entorno Servidor” y “Desarrollo Web en Entorno Cliente”. Envía el resultado a la página “procesa.php”, que será la encargada de procesar los datos.

```
<!DOCTYPE HTML PUBLIC
<!-- Desarrollo Web en Entorno Servidor -->
<!-- Tema 2 : Características del Lenguaje PHP -->
<!-- Ejemplo: Formulario web -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Formulario web</title>
</head>
<body>
<form name="input" action="procesa.php" method="post">
Nombre del alumno: <input type="text" name="nombre" /><br />
<p>Ciclos que cursa:</p>
<input type="checkbox" name="ciclos[]" value="DWES" /> Desarrollo web en entorno
servidor<br />
<input type="checkbox" name="ciclos[]" value="DWEC" /> Desarrollo web en entorno
cliente<br />
<br />
Diseño de Aplicaciones Web Tema 2
- 25 -
<input type="submit" value="Enviar" />
</form>
</body>
</html>
```

Fíjate que si en un formulario web tienes que enviar alguna variable en la que sea posible almacenar más de un valor, como es el caso de las casillas de verificación en el ejemplo anterior (se pueden marcar varias a la vez), tendrás que ponerle corchetes al nombre de la variable para indicar que se trata de un array.

Educar en la Verdad para ser libres

3.3. Procesamiento de la información devuelta por un formulario web.

En el ejemplo anterior creaste un formulario en una página HTML que recogía datos del usuario y los enviaba a una página PHP para que los procesara. Como usaste el método POST, los datos se pueden recoger utilizando la variable $_POST. Si simplemente los quisieras mostrar por pantalla, éste podría ser el código de "procesa.php":

Código de "procesa.php"

```
<!DOCTYPE HTML PUBLIC “
<!-- Desarrollo Web en Entorno Servidor -->
<!-- Tema 2 : Características del Lenguaje PHP -->
<!-- Ejemplo: Procesar datos post -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Desarrollo Web</title>
</head>
<body>
<?php
$nombre = $_POST['nombre'];
$modulos = $_POST['modulos'];
print "Nombre: ".$nombre."<br />";
foreach ($modulos as $modulo) {
print "Modulo: ".$modulo."<br />";
}
?>
</body>
</html>
```

3.3. Procesamiento de la información devuelta por un formulario web.

Si por el contrario hubieras usado el método GET, el código necesario para procesar los datos sería similar; simplemente haría falta cambiar la variable $_POST por $_GET.

Código necesario para procesar los datos

```
<!DOCTYPE HTML PUBLIC “
<!-- Desarrollo Web en Entorno Servidor -->
<!-- Tema 2 : Características del Lenguaje PHP -->
<!-- Ejemplo: Procesar datos get -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Desarrollo Web</title>
</head>
<body>
<?php
$nombre = $_GET['nombre'];
$modulos = $_GET['modulos'];
print "Nombre: ".$nombre."<br />";
foreach ($modulos as $modulo) {
print "Modulo: ".$modulo."<br />";
}
?>
</body>
</html>
```

Educar en la Verdad para ser libres

3.3. Procesamiento de la información devuelta por un formulario web.

En cualquiera de los dos casos podrías haber usado $_REQUEST sustituyendo respectivamente a $_POST y a $_GET.

Ejemplo formulario web utilizando request

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "
http://www.w3.org/TR/html4/loose.dtd">
<!-- Desarrollo Web en Entorno Servidor -->
<!-- Tema 2 : Características del Lenguaje PHP -->
<!-- Ejemplo: Procesar datos request -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Desarrollo Web</title>
</head>
<body>
<?php
$nombre = $_REQUEST['nombre'];
$modulos = $_REQUEST['modulos'];
print "Nombre: ".$nombre."<br />";
foreach ($modulos as $modulo) {
print "Modulo: ".$modulo."<br />";
}
?>
</body>
</html>
```

Educar en la Verdad para ser libres

3.3. Procesamiento de la información devuelta por un formulario web.

Siempre que sea posible, es preferible validar los datos que se introducen en el navegador antes de enviarlos. Para ello deberás usar código en lenguaje Javascript.

Si por algún motivo hay datos que se tengan que validar en el servidor, por ejemplo, porque necesites comprobar que los datos de un usuario no existan ya en la base de datos antes de introducirlos, será necesario hacerlo con código PHP en la página que figura en el atributo action del formulario.

En este caso, una posibilidad que deberás tener en cuenta es 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 por qué.

Para hacerlo de este modo, tienes que comprobar si la página recibe datos (hay que mostrarlos y no generar el formulario), o si no recibe datos (hay que mostrar el formulario). Esto se puede hacer utilizando la función isset con una variable de las que se deben recibir (por ejemplo, poniéndole un nombre al botón de enviar y comprobando sobre él). En el siguiente código de ejemplo se muestra cómo hacerlo.

3.3. Procesamiento de la información devuelta por un formulario web.

Procesar datos en la misma página que el formulario

Fíjate en la forma de englobar el formulario dentro de una sentencia **else** para que sólo se genere si no se reciben datos en la página. Además, para enviar los datos a la misma página que contiene el formulario puedes usar

$_SERVER['PHP_SELF']

para obtener su nombre; esto hace que no se produzca un error aunque la página se cambie de nombre.

```
<!DOCTYPE HTML PUBLIC “
<!-- Desarrollo Web en Entorno Servidor -->
<!-- Tema 2 : Características del Lenguaje PHP -->
<!-- Ejemplo: Procesar datos en la misma página que el formulario -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Desarrollo Web</title>
</head>
<body>
<?php
if (isset($_POST['enviar'])) {
$nombre = $_POST['nombre'];
$modulos = $_POST['modulos'];
print "Nombre: ".$nombre."<br />";
foreach ($modulos as $modulo) {
print "Modulo: ".$modulo."<br />";
}
} else {
?>
<form name="input" action="<?php echo $_SERVER['PHP_SELF'];?>" 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="DWEC" />
Desarrollo web en entorno cliente<br />
<br />
<input type="submit" value="Enviar" name="enviar"/>
</form>
<?php
}
?>
</body>
</html>
```

3.3. Generación de formularios WEB

Vamos a volver sobre el ejemplo anterior, revisando los datos que se obtienen antes de mostrarlos.
Concretamente, tienes que comprobar que el nombre no esté vacío, y que se haya seleccionado como mínimo uno de los módulos.
Además, en el caso de que falte algún dato, deberás generar el formulario rellenando aquellos datos que el usuario haya introducido correctamente.
Lo primero que tienes que hacer es la validación de los datos. En el ejemplo propuesto será algo así como:

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

Para que el usuario no pierda, después de enviar el formulario, los datos correctamente introducidos,utiliza el atributo value en las entradas de texto:

```
Nombre del alumno:
<input type="text" name="nombre" value="<?php 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="checked"';
?>
/>
```

Educar en la Verdad para ser libres

3.4. Generación de formularios WEB

Fíjate en el uso de la función in_array para buscar un elemento en un array.

Para indicar al usuario los datos que no ha rellenado (o que ha rellenado de forma incorrecta), deberás comprobar si es la primera vez que se visualiza el formulario, o si ya se ha enviado. Se puede hacer por ejemplo de la siguiente forma:

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

Revisa el documento con el ejemplo completo, fijándote en las partes que hemos comentado anteriormente.

Educar en la Verdad para ser libres

3.4. Generación de formularios WEB

Ejemplo de validación

```
<!DOCTYPE HTML PUBLIC “
<!-- Desarrollo Web en Entorno Servidor -->
<!-- Tema 2 : Características del Lenguaje PHP -->
<!-- Ejemplo: Validar datos en la misma página que el formulario -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Desarrollo Web</title>
</head>
<body>
<?php
if (!empty($_POST['modulos']) && !empty($_POST['nombre'])) {
$nombre = $_POST['nombre'];
$modulos = $_POST['modulos'];
print "Nombre: ".$nombre."<br />";
foreach ($modulos as $modulo) {
print "Modulo: ".$modulo."<br />";
}
} else {
?>
<form name="input" action="<?php echo $_SERVER['PHP_SELF'];?>"
method="post">
Nombre del alumno:
<input type="text" name="nombre" value="<?php echo
$_POST['nombre'];?>" />
<?php
if (isset($_POST['enviar']) && empty($_POST['nombre']))
echo "<span style='color:red'> &lt;-- Debe introducir un nombre!!</span>"
?><br />
```

```
<p>Módulos que cursa:
<?php
if (isset($_POST['enviar']) && empty($_POST['modulos']))
echo "<span style='color:red'> &lt;-- 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="DWEC"
<?php
if(isset($_POST['modulos']) && in_array("DWEC",$_POST['modulos']))
echo 'checked="checked"';
?>
/>
Desarrollo web en entorno cliente<br />
<br />
<input type="submit" value="Enviar" name="enviar"/>
</form>
<?php
}
?>
</body>
</html>
```

Educar en la Verdad para ser libres

3.4. Generación de formularios WEB

Modifica el ejercicio que mostraba la fecha en castellano, para que obtenga lo mismo a partir de un día, mes y año introducido por el usuario. Antes de mostrar la fecha, se debe comprobar que es correcta. Utilizar la misma página PHP para el formulario de introducción de datos y para mostrar la fecha obtenida en castellano.
Consultar las funciones checkdate y mktime en el manual de PHP.

```
<!DOCTYPE HTML PUBLIC “
<!-- Desarrollo Web en Entorno Servidor -->
<!-- Tema 2 : Características del Lenguaje PHP -->
<!-- Ejemplo: Mostrar fecha completa a partir de día, mes y año introducidos
-->
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<title>Fecha completa a partir de día, mes y año</title>
</head>
<body>
<?php
date_default_timezone_set('Europe/Madrid');
if (!empty($_POST['dia']) && !empty($_POST['mes']) &&
!empty($_POST['ano'])) {
if (checkdate($_POST['mes'],$_POST['dia'],$_POST['ano'])){
$fecha = mktime(0,0,0,$_POST['mes'],$_POST['dia'],$_POST['ano']);
$numero_dia_semana = date("N", $fecha);
switch($_POST['mes']){
case 1: $mes = "Enero";
break;
case 2: $mes = "Febrero";
break;
case 3: $mes = "Marzo";
break;
```

Educar en la Verdad para ser libres

3.4. Generación de formularios WEB

```
case 4: $mes = "Abril";
break;
case 5: $mes = "Mayo";
break;
case 6: $mes = "Junio";
break;
case 7: $mes = "Julio";
break;
case 8: $mes = "Agosto";
break;
case 9: $mes = "Septiembre";
break;
case 10: $mes = "Octubre";
break;
case 11: $mes = "Noviembre";
break;
case 12: $mes = "Diciembre";
break;
}
switch($numero_dia_semana){
case 1: $dia_semana = "Lunes";
break;
case 2: $dia_semana = "Martes";
break;
case 3: $dia_semana = "Miércoles";
break;
case 4: $dia_semana = "Jueves";
break;
case 5: $dia_semana = "Viernes";
break;
case 6: $dia_semana = "Sábado";
break;
case 7: $dia_semana = "Domingo";
break;
}
```

```
print $dia_semana.", ".date("j", $fecha)." de ".$mes." de ".date("Y", $fecha);
}else
print "<span style='color:red'>La fecha introducida no es correcta!!</span>";
}
?>
<form name="input" action="<?php echo $_SERVER['PHP_SELF'];?>"
method="post">
Dia:
<input type="text" name="dia" value="<?php echo $_POST['dia'];?>" />
<?php

if (isset($_POST['enviar']) && empty($_POST['dia']))
echo "<span style='color:red'> &lt;-- Debe introducir un día!!</span>"
?><br />
Mes:
<input type="text" name="mes" value="<?php echo $_POST['mes'];?>" />
<?php
if (isset($_POST['enviar']) && empty($_POST['mes']))
echo "<span style='color:red'> &lt;-- Debe introducir un mes!!</span>"
?><br />
Año:
<input type="text" name="ano" value="<?php echo $_POST['ano'];?>" />
<?php
if (isset($_POST['enviar']) && empty($_POST['ano']))
echo "<span style='color:red'> &lt;-- Debe introducir un año!!</span>"
?><br />
<input type="submit" value="Enviar" name="enviar"/>
</form>
</body>
</html>
```

Educar en la Verdad para ser libres

Bibliografía

- Carro, JL. (2020) Desarrollo Web en entorno servidor

Educar en la Verdad para ser libres