

U.T. 2: Características del lenguaje PHP

Parte 3: Formularios web

Contenidos:

- Formularios web

Formularios

Subida de ficheros al servidor

Procesamiento de la información devuelta por un formulario web

Generación de formularios web en PHP

1.- 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.

Crea un formulario HTML para introducir el nombre del alumno y el módulo 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. (Solución en Anexo I y en el curso de Moodle (ER1))

En el curso de moodle encontraras un curso sobre formularios en HTML (PSM1)

1.1.- Formularios: Acceso desde PHP

- **Formulario:**

```
<form action="respuesta.php?valor=10" method="post">  
<input type="text" name="nombre">  
</form>
```

- PHP (¿todos correctos?)

```
<?  
echo 'valor: '.$_GET['valor'].'<br>';  
echo 'valor: '.$_POST['valor'].'<br>'; → vacío  
echo 'valor: '.$_REQUEST['valor'].'<br>';  
echo 'nombre: '.$_GET['nombre'].'<br>'; → vacío  
echo 'nombre: '.$_POST['nombre'].'<br>';  
echo 'nombre: '.$_REQUEST['nombre'].'<br>';  
?>
```

1.1.- Formularios: Acceso desde PHP

- Acceso a los diferentes tipos de elementos de entrada de formulario
 - Elementos de tipo **INPUT**
 - TEXT
 - RADIO
 - CHECKBOX
 - BUTTON
 - FILE
 - HIDDEN
 - PASSWORD
 - SUBMIT / RESET
 - Elemento **SELECT**
 - Simple / múltiple
 - Elemento **TEXTAREA**

1.1.- Formularios: Acceso desde PHP

- TEXT

Introduzca la cadena a buscar:

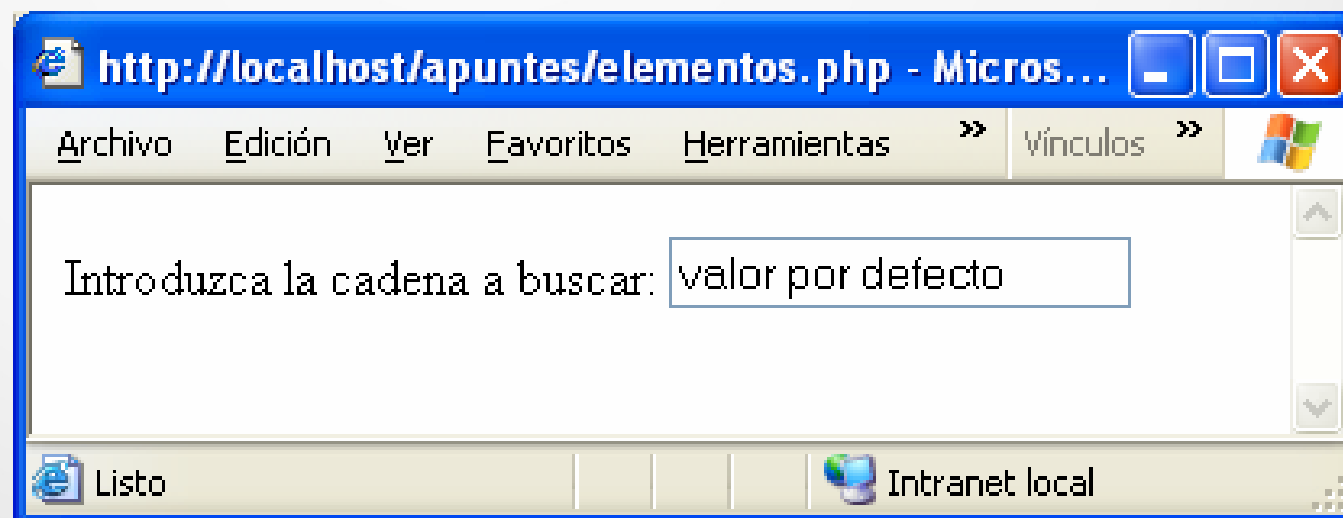
```
<INPUT TYPE="text" NAME="cadena" VALUE="valor por defecto" SIZE="20">
```

```
<?PHP
```

```
$cadena = $_REQUEST['cadena'];
```

```
echo $cadena;
```

```
?>
```



1.1.- Formularios: Acceso desde PHP

- RADIO

Sexo:

```
<INPUT TYPE="radio" NAME="sexo" VALUE="M" CHECKED>Mujer
```

```
<INPUT TYPE="radio" NAME="sexo" VALUE="H">Hombre
```

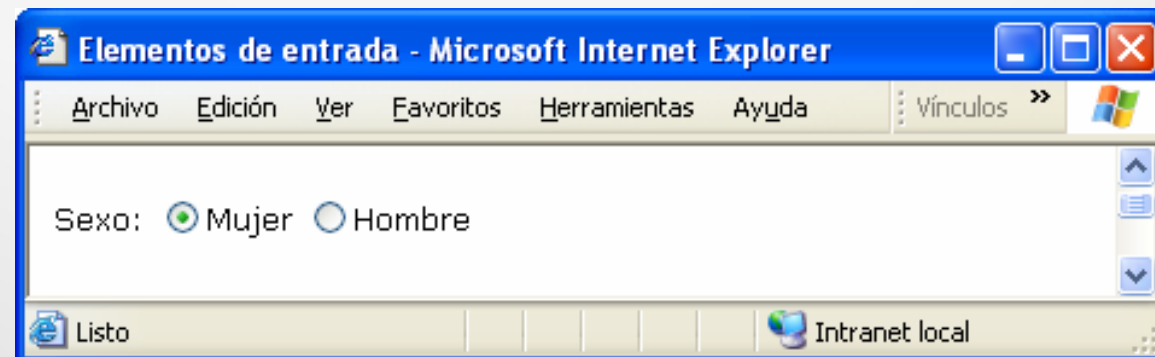
```
<?PHP
```

```
    $sexo = $_REQUEST['sexo'];
```

```
    echo ($sexo);
```

```
?>
```

- Los botones radio se llaman igual para que si se elije uno se desmarquen los otros.



1.1.- Formularios: Acceso desde PHP

- BUTTON

```
<INPUT TYPE="button" NAME="actualizar" VALUE="Actualizar datos">
```

```
<?PHP
```

```
    $actualizar = $_REQUEST['actualizar'];
```

```
    if ($actualizar)
```

```
        print ("Se han actualizado los datos");
```

```
?>
```



1.1.- Formularios: Acceso desde PHP

- HIDDEN

```
<INPUT TYPE='hidden' NAME='username' VALUE="$usuario">
```

```
<?PHP
```

```
    $username = $_REQUEST['username'];
```

```
    echo $username;
```

```
?>
```



1.1.- Formularios: Acceso desde PHP

- PASSWORD

Contraseña: <INPUT TYPE="password" NAME="clave">

<?PHP

```
$clave = $_REQUEST['clave'];
```

```
echo $clave;
```

?>



1.1.- Formularios: Acceso desde PHP

- SUBMIT

```
<INPUT TYPE="submit" NAME="enviar" VALUE="Enviar datos">
```

```
<?PHP
```

```
    $enviar = $_REQUEST['enviar'];
```

```
    if ($enviar)
```

```
        echo "Se ha pulsado el botón de enviar";
```

```
?>
```



1.1.- Formularios: Acceso desde PHP

- RESET

```
<INPUT TYPE="reset" NAME="borrar" VALUE="Limpiar datos">
```

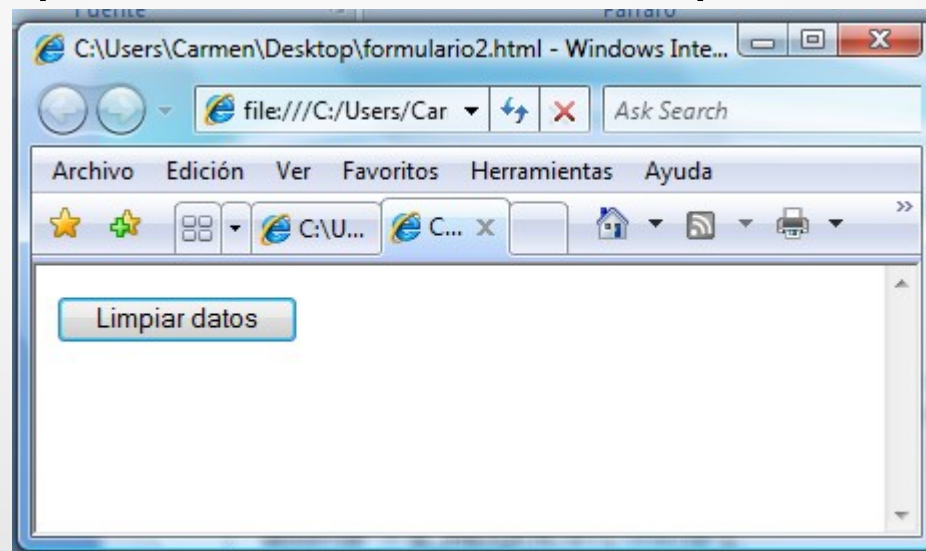
```
<?PHP
```

```
    $borrar = $_REQUEST['borrar'];
```

```
    if ($borrar)
```

```
        echo "Se ha pulsado el botón de limpiar datos";
```

```
?>
```



1.1.- Formularios: Acceso desde PHP

- SELECT Simple

Color:

```
<SELECT NAME="color">
```

```
<OPTION VALUE="rojo" SELECTED>Rojo</OPTION>
```

```
<OPTION VALUE="verde">Verde</OPTION>
```

```
<OPTION VALUE="azul">Azul</OPTION>
```

```
</SELECT>
```

```
<?PHP
```

```
    $color = $_REQUEST['color'];
```

```
    echo $color;
```

```
?>
```



1.1.- Formularios: Acceso desde PHP

- SELECT Múltiple
 - Valores vectoriales de un formulario:
 - Select múltiples o botones de comprobación con el mismo nombre
 - Si no se indica nada, sólo se tiene acceso a un valor
 - En el código HTML hay que añadir "[]" al nombre del control
 - Devuelve un array, con count() podemos conocer su tamaño

Idiomas:

```
<SELECT MULTIPLE SIZE="3" NAME="idiomas[]">
<OPTION VALUE="ingles" SELECTED>Inglés</OPTION>
<OPTION VALUE="frances">Francés</OPTION>
<OPTION VALUE="aleman">Alemán</OPTION>
<OPTION VALUE="holandes">Holandés</OPTION>
</SELECT>
```

```
<?PHP
    $idiomas = $_REQUEST['idiomas'];
    foreach ($idiomas as $idioma)
        echo "$idioma<BR>\n";
?>
```



1.1.- Formularios: Acceso desde PHP

- CHECKBOX

- botones de comprobación con el mismo nombre → usar arrays []

<INPUT TYPE="checkbox" NAME="extras[]" VALUE="garaje" CHECKED>Garaje

<INPUT TYPE="checkbox" NAME="extras[]" VALUE="piscina">Piscina

<INPUT TYPE="checkbox" NAME="extras[]" VALUE="jardin">Jardín

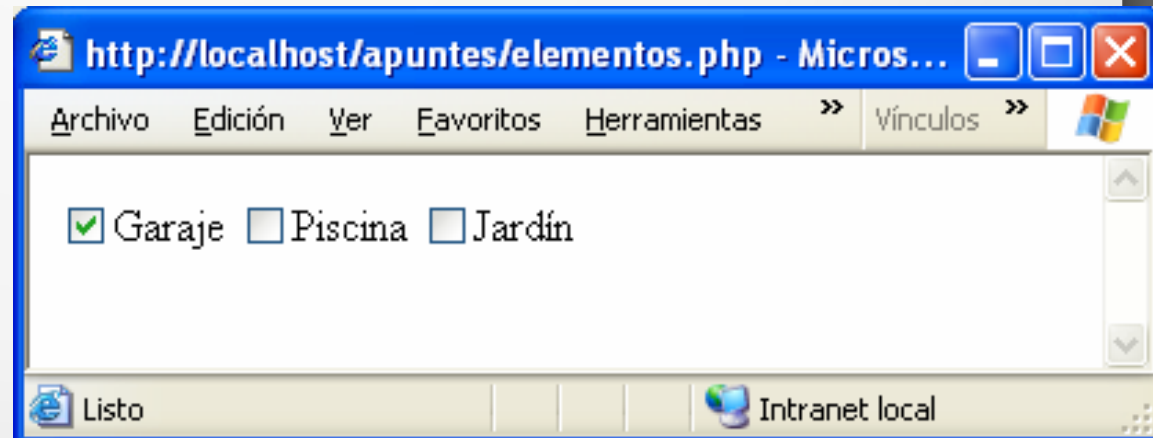
<?PHP

```
$extras = $_REQUEST['extras'];
```

```
foreach ($extras as $extra)
```

```
    echo "$extra<BR>\n";
```

?>



1.1.- Formularios: Acceso desde PHP

- TEXTAREA

Comentario:

```
<TEXTAREA COLS="50" ROWS="4" NAME="comentario">
```

Este libro me parece ...

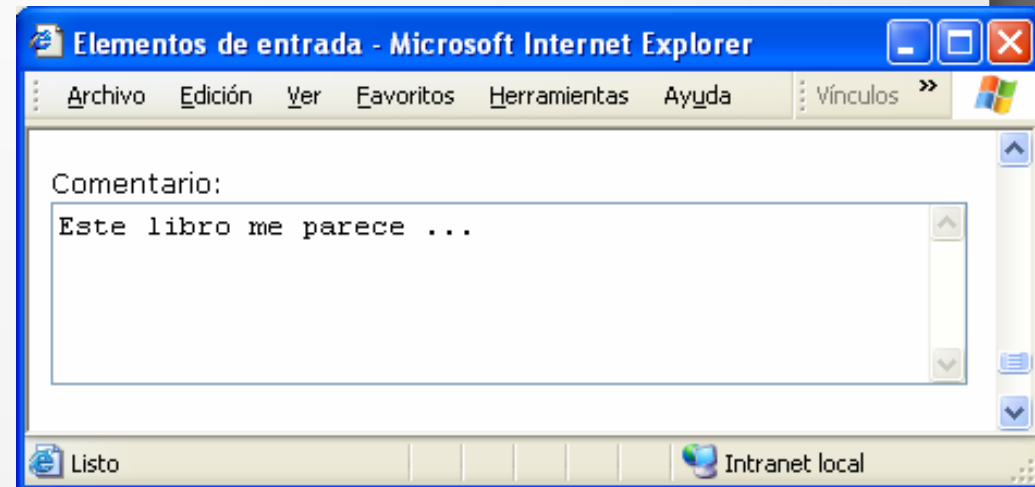
```
</TEXTAREA>
```

```
<?PHP
```

```
    $comentario = $_REQUEST['comentario'];
```

```
    echo $comentario;
```

```
?>
```



- Ejercicio: formulario simple que incluya todos los controles excepto FILE y BUTTON

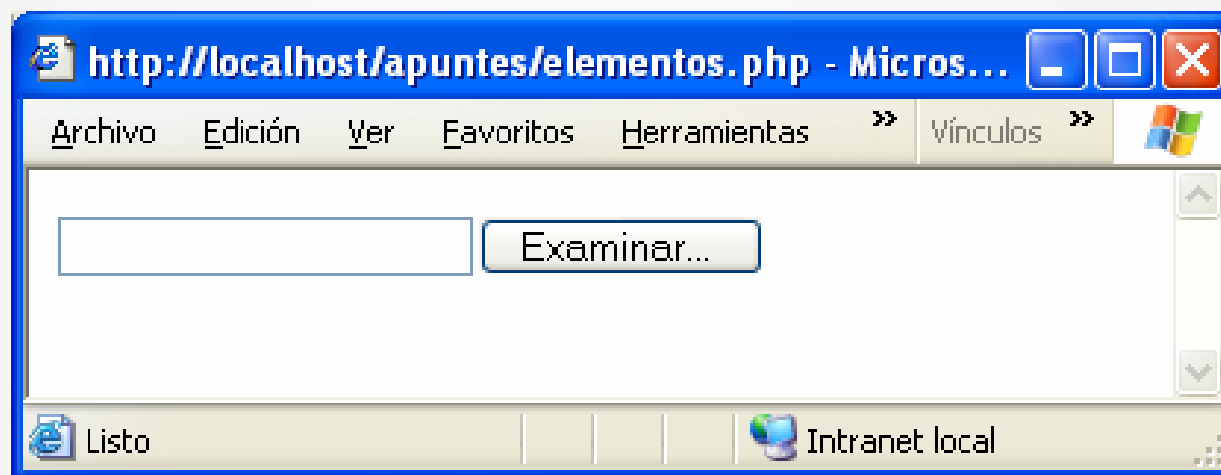
1.1.- Formularios: Acceso desde PHP

- FILE

<FORM ACTION="procesa.php" METHOD="post" ENCTYPE="multipart/form-data">

<INPUT TYPE="file" NAME="fichero">

</FORM>



- Ejercicio: página php que muestra los datos introducidos desde el formulario del ejercicio anterior.
 - Ilustra cómo acceder a los valores introducidos desde todos los tipos de elementos de entrada de un formulario, con excepción de los tipos BUTTON y FILE, que se tratan en ejercicios posteriores

1.2.- Formularios: Subida de ficheros al servidor

- Para subir un fichero al servidor se utiliza el elemento de entrada FILE
- Hay que tener en cuenta una serie de consideraciones importantes:
 - El elemento FORM debe tener el atributo **ENCTYPE="multipart/form-data"**
 - El fichero tiene un límite en cuanto a su tamaño. Este límite se fija de dos formas diferentes y complementarias:
 - En el fichero de configuración php.ini
 - En el propio formulario

1.2.- Formularios: Subida de ficheros al servidor

- php.ini

```
.....  
,,,,,,,,,,,,,,,,,,,,,
```

```
; File Uploads ;
```

```
.....  
,,,,,,,,,,,,,,,,,,,,,
```

```
; Si se permite o no subir archivos mediante HTTP
```

```
file_uploads = On
```

```
;
```

```
; Tamaño máximo de cada archivo subido.
```

```
upload_max_filesize = 2M
```

```
; Tamaño máximo de los datos mandados por POST
```

```
;(incluidos los que no sean archivos)
```

```
post_max_size = 8M
```

- formulario

```
<INPUT TYPE="HIDDEN" NAME="MAX_FILE_SIZE" VALUE='102400'>
```

```
<INPUT TYPE="FILE" NAME="fichero">
```

se tiene que llamar así y es un entero con el valor en bytes

1.2.- Formularios: Subida de ficheros al servidor

```
<INPUT TYPE="FILE" SIZE="44" NAME="imagen">
```

- La variable `$_FILES` contiene toda la información del fichero subido:
 - `$_FILES['imagen']['name']`
 - Nombre original del fichero en el cliente
 - `$_FILES['imagen']['type']` → ¡Inseguro!, mejor usar `mime_content_type()`
 - Tipo MIME del fichero. Por ejemplo, "image/gif"
 - `$_FILES['imagen']['size']`
 - Tamaño en bytes del fichero subido
 - `$_FILES['imagen']['tmp_name']`
 - Nombre temporal del fichero que se genera para guardar el fichero subido
 - `$_FILES['imagen']['error']`
 - Código de error asociado a la subida del fichero

1.2.- Formularios: Subida de ficheros al servidor

¿Por qué es inseguro usar `$_FILES['imagen']['type']`?

- Este dato es proporcionado por el navegador del cliente, por lo tanto:
 - Dependiendo del navegador la información puede ser correcta o no (algunos navegadores envía todos los archivos con el tipo genérico *application/octet-stream*)
 - Al ser proporcionado por el navegador del cliente y no ser verificado por el servidor php puede ser fácilmente manipulado con fines maliciosos.
- En su lugar debemos usar `mime_content_type()`
 - Es el servidor el que lee los primeros bytes del fichero y deduce qué contenido tiene.

`mime_content_type($_FILES['imagen']['tmp_name'])`

Listado de tipos aquí:

<http://svn.apache.org/repos/asf/httpd/httpd/trunk/docs/conf/mime.types>

1.2.- Formularios: Subida de ficheros al servidor

- Consideraciones:
 - Debe darse al fichero un nombre único. Por ello, y como norma general, debe descartarse el nombre original del fichero y crear uno nuevo que sea único, p.e añadiéndole la fecha y hora
 - El fichero subido se almacena en un directorio temporal y hemos de moverlo al directorio de destino usando la función

move_upload_file()

- Procedimiento:
 - si se ha subido correctamente el fichero
 - Asignar un nombre al fichero
 - Mover el fichero a su ubicación definitiva
 - si no
 - Mostrar un mensaje de error
 - finsi

1.2.- Formularios: Subida de ficheros al servidor

- Procedimiento:

si se ha subido correctamente el fichero

- Lo comprobamos con **is_uploaded_file("nombre temporal de \$_FILES")**
- Devuelve **true** si el archivo que se le pasa se ha subido por HTTP POST. Evita que el usuario intente usar archivos del servidor /etc/passwd

Asignar un nombre al fichero

- Añadir marca de tiempo

Mover el fichero a su ubicación definitiva

- **move_uploaded_file (\$_FILES['archivo'] ['tmp_name'], \$destino)**
- Lo mueve y si no puede da error

si no

Mostrar un mensaje de error

finsi

1.2.- Formularios: Subida de ficheros al servidor

- Ejemplo(I)

```
<html><body>
```

Inserción de la fotografía del usuario:

```
<form action="inserta.php" method="post" enctype="multipart/form-data">
```

```
<?php
```

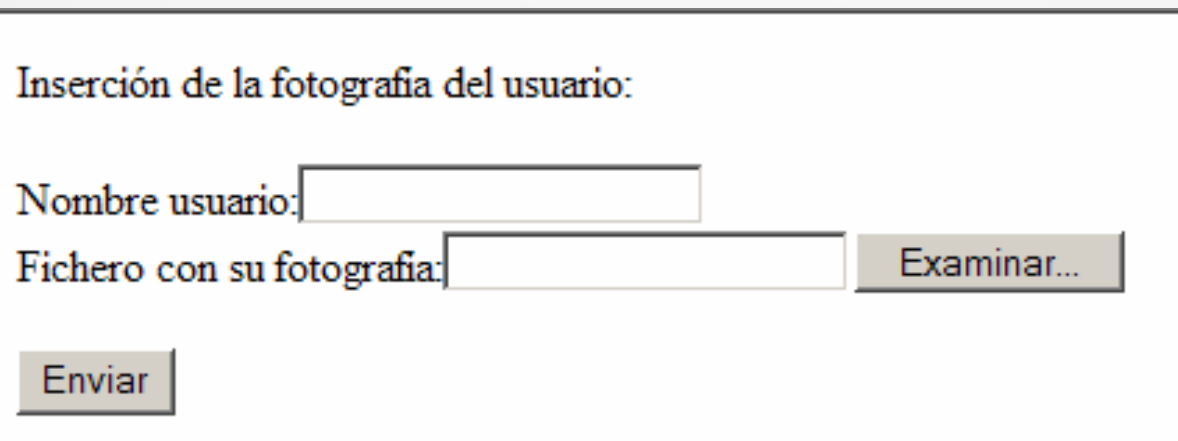
```
echo "Nombre usuario:<input type='text' name='usuario'/><br/>";
```

```
echo "Fichero con su fotografía:<input type='file' name='imagen'/><br/>";
```

```
?>
```

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

```
</form></body></html>
```



Inserción de la fotografía del usuario:

Nombre usuario:

Fichero con su fotografía:

1.2.- Formularios: Subida de ficheros al servidor

- Ejemplo(II) - inserta.php

```
<html><body>
```

```
<?php
```

```
    echo "name:".$_FILES['imagen']['name']."\n";
```

```
    echo "tmp_name:".$_FILES['imagen']['tmp_name']."\n";
```

```
    echo "size:".$_FILES['imagen']['size']."\n";
```

```
    echo "type:".$_FILES['imagen']['type']."\n";
```

```
    if (is_uploaded_file ($_FILES['imagen']['tmp_name'] )){
```

```
        $nombreDirectorio = "img/";
```

```
        $nombreFichero = $_FILES['imagen']['name'];
```

```
        if (is_dir($nombreDirectorio)){ // es un directorio existente
```

```
            $idUnico = time();
```

```
            $nombreFichero = $idUnico."-".$nombreFichero;
```

```
            $nombreCompleto = $nombreDirectorio.$nombreFichero;
```

```
            move_uploaded_file ($_FILES['imagen']['tmp_name'],$nombreCompleto);
```

```
            echo "Fichero subido con el nombre: $nombreFichero<br>";
```

```
        }
```

```
    else
```

```
        echo 'Directorio definitivo inválido';
```

```
    } else
```

```
        print ("No se ha podido subir el fichero\n");
```

```
?>
```

```
</body></html>
```

```
name:Foto.png
```

```
tmp_name:C:\xampp\tmp\php7EE.tmp
```

```
size:15811
```

```
type:image/x-png
```

```
Fichero subido con el nombre: 1241894493-Foto.png
```


1.2.- Formularios: Subida de ficheros al servidor

- **is_uploaded_file (\$_FILES['imagen']['tmp_name'])**
 - Devuelve TRUE si el archivo que se pasa fue cargado a través de HTTP POST. Evita que un usuario intente que se manejen ficheros no cargados por POST. p.e /etc/passwd
 - Necesita como argumento \$_FILES['archivo_usuario']['tmp_name']
 - Si se le pasa \$_FILES['archivo_usuario']['name'] **no funciona**.

<?php

```
if (is_uploaded_file($_FILES['archivo_usuario']['tmp_name'])) {  
    echo "El archivo ". $_FILES['archivo_usuario']['name'] ." fue cargado correctamente.\n";  
    echo "Mostrando su contenido\n";  
    readfile($_FILES['archivo_usuario']['tmp_name']);  
} else {  
    echo "Posible ataque de carga de archivo: ";  
    echo "nombre de archivo ". $_FILES['archivo_usuario']['tmp_name'] . ".\n";  
}
```

?>

1.2.- Formularios: Subida de ficheros al servidor

- `move_uploaded_file ($_FILES['imagen']['tmp_name'], $destino)`

`nombre_temporal_archivo`



- Esta función realiza un chequeo para asegurar que el archivo indicado por el primer parámetro sea un archivo cargado a través de HTTP POST.
- Si el archivo es válido, será movido al nombre de archivo dado por **destino**.
- Si *nombre_temporal_archivo* no es un archivo cargado válido, no hará nada, y devolverá FALSE.
- Si *nombre_temporal_archivo* es un archivo cargado válido, pero no puede ser movido por alguna razón, no hará nada, devolverá FALSE y dará una advertencia.

1.2.- Formularios: Subida de ficheros al servidor

- Variable predefinida `$_FILES`
 - Precauciones:
 - Permisos de escritura en el directorio temporal
 - Permisos de escritura en el directorio de destino
 - Atención con los ficheros que puedan subir los usuarios: Troyanos, scripts, ejecutables, etc.
- Ejercicio: subida de un fichero al servidor
 - Ilustra cómo subir ficheros a un servidor, cómo controlar su tamaño, cómo crear un nombre único para el fichero y cómo almacenarlo en el lugar deseado .

1.3.- 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":

Consulta el Anexo II con el Código de "procesa.php"

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`.

Consulta el Anexo III con el Código necesario para procesar los datos

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

Consulta el Anexo IV Usando `$_REQUEST` para sustituir respectivamente a `$_POST` y a `$_GET`

1.3.- 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.

Consulta el Anexo V que muestra como 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.

1.4.- Generación de formularios web en PHP

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 (utilizamos la función **isset** para comprobar que la variable que queremos mostrar existe, para que no nos muestre un mensaje de error si es la primera vez que se carga la página y todavía no ha enviado nada el formulario):

Nombre del alumno:

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

1.4.- Generación de formularios web en PHP

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"';
  ?>
/>
```

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 />
```

1.4.- Generación de formularios web en PHP

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

Consulta el Anexo VI que contiene el Documento con el ejemplo anterior completo

Una forma de enviar información de una página PHP a otra, es incluyéndola en campos ocultos dentro de un formulario.

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. (Solución en Anexo VII y en el curso de Moodle (ER2))

Anexo I.- Solución propuesta V

```
<!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: 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="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" />  
    </form>  
  </body>  
</html>
```

Anexo II.- Código de “procesa.php”

```
<!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 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>
```

Anexo III.- Código necesario para procesar datos

```
<!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 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>
```

Anexo IV.- Ejemplo de 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>
```

Anexo V.- Procesar datos en la misma página que el formulario

```
<!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 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="DWECE" />
  Desarrollo web en entorno cliente<br />
  <br />
  <input type="submit" value="Enviar" name="enviar"/>
</form>
<?php
  }
?>
</body>
</html>
```

Anexo VI.- Ejemplo de validación

```
<!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: 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 if (isset ($_POST['nombre'])) 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="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>
```

```
</html>
```

Anexo VII.- Solución propuesta VI

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