

U.T. 3: Formularios Ficheros Adjuntos

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
-

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

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

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

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
 - ❑ `$_FILES['imagen']['type']`
 - ❑ Tipo MIME del fichero. Por ejemplo, "image/gif"
 - ❑ `$_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

FORMULARIOS: Subida de ficheros al servidor

❑ Consideraciones (cont)

- ❑ 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

Validación de datos

No es suficiente la validación del lado cliente

Podemos realizarla de dos maneras:

- Utilizando filtros de validación de PHP.
<http://php.net/manual/es/filter.filters.validate.php>
- Utilizando expresiones regulares.
<http://php.net/manual/es/book.pcre.php>

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

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";
}
?>
```


FORMULARIOS: Subida de ficheros al servidor

❑ **move_uploaded_file (\$_FILES['imagen']['tmp_name'], \$destino)**



A red bracket originates from the `'tmp_name'` parameter in the function signature above and points down to a red-bordered box containing the text `nombre_temporal_archivo`.

`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.
-

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 4: 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.
-