

Subida de ficheros al servidor en el lenguaje PHP

Subida y gestión de ficheros en el servidor

- Para subir un fichero al servidor se utiliza
`<input type="file".../>`
- dentro de un formulario HTML
- El elemento **<form>** debe tener el atributo
`enctype="multipart/form-data"`
- El atributo **method** debe ser **post**, no se puede usar get para enviar ficheros.

Tamaño del fichero a subir

- Podemos fijar el límite de tamaño en el fichero php.ini
 - <http://php.net/manual/es/ini.core.php#ini.upload-max-filesize>
- También podemos fijar el límite (**en Kilobytes**) en el propio formulario a través de un campo oculto (`type="hidden"`) denominado **MAX_FILE_SIZE**.

```
<input type="hidden" name="MAX_FILE_SIZE" value="10240">
```

Ruta temporal del fichero subido

- Al cargar un archivo, se guardará en una ubicación temporal indicada por la opción **upload_tmp_dir** en el php.ini.
 - <http://php.net/manual/es/ini.core.php#ini.upload-tmp-dir>
- Si no movemos el archivo o lo cambiamos de nombre, cuando termine la ejecución del script, este será eliminado.

Acceder a los ficheros subidos (I)

- Accederemos al fichero mediante la variable superglobal **\$_FILES**.
- Tendremos un **array para cada archivo** que hayamos subido.
- Las **claves** del array serán los atributos **name** de los input file del formulario.
 - `<input name="imagen" type="file">`

Elementos del array del fichero subido

- `$_FILES['imagen']['tmp_name']`: lugar y nombre del archivo temporal en el servidor.
- `$_FILES['imagen']['name']`: Nombre original del fichero en la máquina cliente.
- `$_FILES['imagen']['size']`: Tamaño en bytes del fichero subido.
- `$_FILES['imagen']['type']`: Tipo MIME asociado al fichero. Por ejemplo, "image/gif" o "text/plain".
 - https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types
- `$_FILES['imagen']['error']`: Código de error asociado al fichero subido.

Problemas más habituales

- Especificar en *upload_tmp_dir* un directorio al cual no se tiene acceso.
- La directiva *memory_limit* tiene un valor muy bajo o inferior a *upload_max_filesize*.
- La directiva *max_execution_time* tiene un valor bajo y el script lo excede durante la subida del fichero.
- La directiva *post_max_size* tiene un valor bajo y tu fichero lo excede. Su valor debe ser mayor a *upload_max_filesize*.

Códigos de error

- Lo primero que habrá que hacer es comprobar el código de error del fichero subido (`$_FILES['imagen']['error']`).
- Si es distinto de **UPLOAD_ERR_OK** ha habido algún problema.
- Ver los códigos de error:
 - <http://php.net/manual/es/features.file-upload.errors.php>

Comprobación del tipo de fichero

- A continuación comprobaremos que el tipo del archivo está dentro de los esperados:

```
if ($_FILES['imagen']['type'] !== 'image/gif')
{
    echo 'Error: No se trata de un fichero .GIF.';
    exit;
}
```

Chequeo de archivo

- Es importante para ayudar a verificar que un usuario malicioso no ha intentado engañar al script haciéndole trabajar sobre otro archivos distinto.
- Ataque común: acceder a archivos del sistema operativo.
- Para evitar este tipo de ataques se incluyó la función:
 - `bool is_uploaded_file (string nombre_archivo)`
- Devuelve true si el archivo dado fue cargado a través de HTTP POST.
- Le pasaremos el parámetro:

```
$_FILES['imagen']['tmp_name']
```

Mover el archivo subido

- Para mover el archivo temporal a su ubicación correcta usamos la función:
 - `bool move_uploaded_file(string nombre_archivo, string destino)`
- Esta función se asegura de que el archivo sea un archivo cargado válido.
- Si existe un archivo destino con el mismo nombre, este será sobrescrito.

Evitar sobrescrituras

- Antes de mover el archivo subido es conveniente comprobar que no existe un archivo con el mismo nombre.
 - `bool is_file(string nombre_archivo)`
 - Le pasaremos `$_FILES['imagen']['name']`
- En caso de que exista habrá que cambiar el nombre, por ejemplo añadiendo una marca de tiempo como prefijo.