

Comprobación de la configuración

Antes de empezar con este tema debemos comprobar cuál es la configuración de nuestro **php.ini**.

Si por alguna circunstancia los valores no coincidieran con los que tenemos aquí a la derecha, tendríamos que abrir **php.ini** y modificar aquellas directivas.

Si te apetece, y como simple experimento, podemos cambiar el límite del tamaño máximo de transferencia poniendo:

```
upload_max_filesize=500K
```

un valor más reducido, que nos servirá para hacer una prueba muy sencilla.

Transferencia de ficheros

La transferencia de un fichero requiere dos documentos: un *formulario* que la inicie y un *script* que la recoja.

El formulario

Se diferencia del que hemos visto en páginas anteriores en tres aspectos. Dos de ellos se refieren a cambios dentro de la etiqueta `<form>` y el tercero es un nuevo tipo de *input* del que aún no hemos hablado.

En la etiqueta `<form>` hemos de incluir –obligatoriamente– el **method="POST"** y **ENCTYPE = "multipart/form-data"** ya que no *soporta* ni otro método ni otra forma de encriptación.

El cuerpo del *formulario* ha de contener un nuevo tipo de *input* que utiliza la siguiente sintaxis:

```
<input type='file' name='nm'>
```

La transferencia

Una vez enviado el formulario, el *fichero transferido* se guarda en un *directorio temporal* del servidor –salvo que **php.ini** especifique una cosa distinta– con un *nombre* que le es asignado de forma automática, y, además, en las *variables predefinidas* **\$_FILES** (si la versión admite *superglobales*) y en **\$HTTP_POST_FILES** se recogerán datos relativos al contenido del fichero y a los resultados de la

Requisitos de configuración

Cuando publicamos en un *hosting* no tenemos acceso al fichero de configuración **php.ini** pero sí podemos conocer su configuración mediante el script **info.php**.

Recordemos que ese fue nuestro primer script –lo hemos creado y utilizado– para comprobar nuestra instalación y lo hemos guardado con ese nombre en el root de nuestro servidor –C:\Apache\htdocs– así que podremos acceder a él escribiendo como dirección <http://localhost/info.php>.

La abundante lista que nos muestra **info.php** contiene las siguientes líneas (las podemos localizar fácilmente porque están ordenadas alfabéticamente), que en nuestro caso –configuración por defecto– tendrán los valores que vemos en la imagen.

file_uploads	On	On
upload_max_filesize	2M	2M
upload_tmp_dir	no value	no value

Es imprescindible que **file_uploads=On** (tal como aparece en la imagen) y resulta muy útil también conocer el valor de **upload_max_filesize** que por defecto –tal como ves en la imagen– es de **2Mb**. La primera directiva nos dice que PHP sí permite *subir* ficheros al servidor y la segunda nos indica el *tamaño máximo* (en Mbytes) de los ficheros que pueden ser objeto de esa transferencia.

Formulario para transferencia de archivos

Observa que en el formulario hemos insertado una variable *oculta* (`hidden`) con el fin de limitar el **tamaño máximo** e *impedir* la transferencia de ficheros que **excedan** ese tamaño.

```
<HTML>
<BODY>
<FORM ENCTYPE="multipart/form-data" ACTION="ejemplo86.php" METHOD="post">
# con este input "oculto" establecemos el limite máximo
# del tamaño del fichero a transferir. En este ejemplo 1.000.000 bytes
<INPUT type="hidden" name="lim_tamano" value="1000000">
<p><b>Archivo a transferir</b><br>
<INPUT type="file" name="archivo"></p>
<p><INPUT type="submit" name="enviar" value="Aceptar"></p>
</FORM>
</BODY>
</HTML>
```

ejemplo85.php

Puedes utilizar el script invocado por la *action* de formulario anterior (es este que tienes aquí debajo) para hacer algunas comprobaciones.

Envía un fichero cualquiera –de menos de 500K– y verás que el error es **0**. Repite el envío, ahora con un fichero que sobrepase ese tamaño, y comprobarás que el error toma valor **1** dado que la directiva **upload_max_filesize=500K** del fichero **php.ini** habrá bloqueado la transferencia.

```
<?
/* Mediante el bucle foreach leemos el array $_FILES.
Observa la sintaxis. Escribimos como nombre del array
$_['archivo'] con lo cual foreach leerá los elementos
del array que tienen 'archivo' como primer indice
(coincide con el name que hemos puesto
en la etiqueta input=file del formulario) */
foreach ($_FILES['archivo'] as $indice=>$valor){
```

El formato de `$_FILES` y de `$HTTP_POST_FILES` es el de *array bidimensional*.

El *primero de sus índices* es el nombre de variable usado para la transferencia (el especificado como `name='nm'` en el `input type='file'`).

Los segundos índices –se trata de un *array asociativo*– son estos: *name*, *type*, *tmp_name*, *error* y *size*.

En ellos se recogen: el nombre original de fichero transferido, su formato, el nombre con el que ha sido guardado en el directorio temporal, el tipo de error de transferencia y el tamaño del archivo.

El error puede ser CERO o UNO. Si es CERO indica que la transferencia se ha realizado con éxito. En caso contrario, el valor de ese error es UNO.

Copia del fichero

Tal como hemos visto, el fichero transferido *aún* no está en el servidor. Por el momento se encuentra en un directorio temporal y será preciso hacer una copia en nuestro espacio de servidor. Para este proceso puede utilizarse una función que ya hemos visto en páginas anteriores:

`copy(fich1, fich2)`

donde *fich1* sería el fichero temporal y *fich2* el del nuevo fichero.

El primero de los nombres es el valor contenido en:

`$_FILES['nm']['tmp_name']`

donde *nm* es el valor incluido como *name* en el formulario usado para la transferencia y *tmp_name* es una *palabra reservada* que debe escribirse exactamente con esa sintaxis.

Si la versión de PHP fuera anterior a 4.1.0 habría que utilizar `$HTTP_POST_FILES` en vez de `$_FILES`. En otros casos podrían usarse indistintamente ambos tipos de variable.

El valor *fich2* podría ser un nombre cualquiera asignado en el propio script –podemos verlo en el ejemplo– o el nombre original del fichero transferido. En este caso habría que recogerlo del elemento del array anterior cuyo segundo índice es *name*.

En la cadena *fich2* podría incluirse –recuerda que debes ponerlo entre comillas– un *path* señalando el directorio o subdirectorio donde

```
print $indice."--->".$valor."<br>";
}
/*Dependiendo del navegador que estés utilizando puede ocurrir
que varían los valores del índice type sean distintos.
Cuando se trata de un fichero jpg, con IE devolverá image/pjpeg,
mientras que con Mozilla, Firefox, Opera y Netscape
devolverá image/jpeg.*/

/* repetimos el proceso anterior ahora usando la otra
variable predefinida (recuerda que esta no es superglobal)
$HTTP_POST_FILES y podremos visualizar los índices
y los valores del fichero transferido */

foreach ($HTTP_POST_FILES['archivo'] as $indice=>$valor){
    print $indice."--->".$valor."<br>";
}
?>
```

Cuando está habilitada la opción de transferencias de ficheros es conveniente –en previsión de sorpresas desagradables– tomar algunas precauciones. Una de ellas sería limitar la posibilidad de transferencia a determinados tipos de archivos –imágenes, por ejemplo– impidiendo con ello que pudieran transferirse al servidor ficheros de riesgo, tales como: ejecutables, virus, etcétera.

Cuando se establece este tipo de limitaciones, PHP comprueba los contenidos de los ficheros sin tomar en consideración la extensión de los mismos. Con ello se evita el riesgo de que puedan *esconders* –cambiando la extensión– ficheros distintos de los permitidos.

Aquí tienes un ejemplo de *script* que impide la transferencia de ficheros con extensión distinta a *.jpg* o *.gif*.

```
<?
/* filtramos el tipo de archivos recibidos
de forma que solo se permitan imagenes en formato
jpg ó gif. Si el fichero transferido tuviera formato
distinto, la función exit() acabaría la ejecución del script */

if(!($_FILES['archivo']['type']=="image/pjpeg" OR
    $_FILES['archivo']['type']=="image/jpeg" OR
    $_FILES['archivo']['type']=="image/gif")){
    print "El formato ".$_FILES['archivo']['type'].
        " no está permitido";
    exit();
}else{
    # anidamos este segundo condicional
    # para guardar en una variable
    # la extensión real del fichero
    # mas adelante la utilizaremos
    if ($_FILES['archivo']['type']=="image/pjpeg" OR
        $_FILES['archivo']['type']=="image/jpeg" ){
        $extension=".jpg";
    }else{
        $extension=".gif";
    }
}

/* filtremos ahora el tamaño de modo que no supere
el máximo establecido en el hidden del formulario
(logicamente ese valor no puede superar el valor máximo
de la configuración de php, pero si puede ser menor)
y también evitaremos archivos sin contenido,
es decir con tamaño CERO */
if($_FILES['archivo']['size']>$_POST['lim_tamano']
    OR $_FILES['archivo']['size']==0){
    print "El tamaño ".$_FILES['archivo']['size'].
        " excede el límite";
    exit();
}

# asignemos un nombre a la imagen transferida
# de modo que se guarde en el servidor
# con un nombre distinto, asignado por nosotros
# con ello, podemos evitar duplicidades de nombres
# ya que si existiera un fichero con el mismo nombre
# que el enviado por el cliente, se sobrescribiría

$nuevo_nombre="foto_abuelita";
# añadámosle la extensión real de fichero que teníamos
```

no incluirlo, el fichero se copiaría en el directorio desde el que se está ejecutando el script.

Sintaxis alternativa

La opción anterior tiene una alternativa, igual de eficiente y mucho más segura. Se trata de:

move_uploaded_file(fich1, fich2)

que tiene la misma utilidad que *copy* y añade algunas ventajas tales como:

– Comprobar que el fichero ha sido transferido mediante el método POST e impedir que se *copie* en el servidor en caso de no cumplirse esa condición.

– Si la opción **safe mode=on** (configuración de php.ini en modo seguro) –por defecto está Off– comprueba, además, que la transferencia del fichero ha sido realizada por el mismo usuario que hace la petición de ejecución del script, evitando que alguien, *maliciosamente*, pueda *mover* ficheros desde el directorio temporal hasta el espacio de servidor.

¡Cuidado..!

Al usar esta función bajo Windows conviene indicar en el parámetro **fich2** la ruta absoluta completa junto con el nombre del fichero ya que –de no hacerlo así– en algunas ocasiones la imagen no será transferida al directorio desde el que se ejecuta el script.

register_globals = ON

Con esta opción habilitada (ON) existe una posibilidad añadida.

En ese caso, la sintaxis podría ser más simple ya que, en el script indicado en la *acción* del formulario, se crearía una variable cuyo nombre coincide con el valor de **name** en el **<INPUT type="FILE">** del formulario.

Esta variable (supongamos que su nombre es \$archivo) tiene una importante peculiaridad, ya que, además de guardar el fichero transferido, contiene (con el formato que tenemos a continuación) tres valores muy importantes:

• **\$archivo_name** (se añade **_name** al nombre de la variable) y recogerá el *nombre del fichero transferido*.

• **\$archivo_size** que contiene el *tamaño del fichero*.

• **\$archivo_type** que recoge el *tipo de fichero*.

```
# recogida en la variable nuevo_nombre
$nuevo_nombre .= $extension;
# aceptemos la transferencia siempre que el archivo tenga nombre
if ($_FILES['archivo']['tmp_name'] != "none" ){
/* con la función copy
pasaremos el archivo que está en el directorio temporal
al subdirectorio que contiene el script que estamos
ejecutando. Podríamos incluir un path y copiarlo
a otro directorio */
    if (copy($_FILES['archivo']['tmp_name'], $nuevo_nombre)) {
        echo "<h2>Se ha transferido el archivo</h2>";
    }
}
}
else{
    echo "<h2>No ha podido transferirse el fichero</h2>";
}
?>
```

Usando «copy»

Usando «move_uploaded_file»

Ejemplo de script para el caso register_globals=ON

Aquí debajo incluimos el código del script que –de forma alternativa a los anteriores– podría utilizarse con esta opción de configuración.

Pese a su simplicidad te sugerimos no usarlo. Los anteriores compensan con creces su mayor complejidad sintáctica con la versatilidad –funcionan con cualquier opción de register global– y con la seguridad que añaden a las transacciones.

```
<?
if ($archivo != "none" AND $archivo_size != 0
    AND $archivo_size<=$lim_tamano){
    if (copy ($archivo, $archivo_name)) {
        echo "<h2>Se ha transferido el archivo $archivo_name</h2>";
        echo "<br>Su tamaño es: $archivo_size bytes<br>";
        echo "<br>El fichero es tipo: $archivo_type <br>";
    }
}
}
else{
    echo "<h2>No ha podido transferirse el fichero</h2>";
    echo "<h3>su tamaño no puede exceder de $lim_tamano bytes</h2>";
}
?>
```

Ejercicio nº 29

Modifica el formulario que has creado en el ejercicio nº 9 añadiéndole la opción para que tus alumnos puedan enviarte –junto con los datos– una fotografía.

Modifica también el script asociado a aquel formulario de forma que puedas asignar un nombre a la fotografía, visualizarla –previamente copiada al directorio de trabajo– junto con los datos personales y guardar los datos personales y el nombre de la imagen en un fichero.

Anterior



Índice



Siguiente

