Empregar un formulario web para subir ficheiros ao servidor

É posible empregar o método POST para subir un ficheiro ao servidor web. Neste caso é necesario que o tipo de codificación para os datos que se envían estea especificado como "multipart/form-data" na etiqueta "<form>", e crear unha etiqueta HTML de tipo "<input>" co atributo "type='file'".

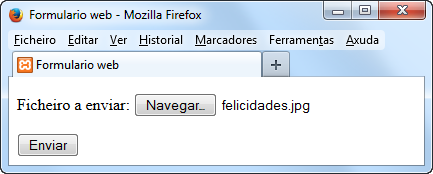
<form enctype="multipart/form-data" action="envio.php" method="POST">

<p>Ficheiro a enviar: <input name="imaxe" type="file" /></p>

<input type="hidden" name="MAX\_FILE\_SIZE" value="1024" />

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

</form>



É posible limitar o tamaño máximo do arquivo a enviar dende o formulario web. Faise engadindo ao formulario un campo oculto de tipo "<input>" co atributo "name='MAX\_FILE\_SIZE'". O tamaño máximo (en KB) indícase no atributo "value".

El elemento MAX\_FILE\_SIZE no puede especificar un tamaño de fichero mayor que el establecido en la directiva **upload\_max\_filesize** del fichero *php.ini*

Procesamento da información introducida nos formularios

A localización temporal na que se reciben os ficheiros subidos polos clientes web e outros datos relativos ás subidas poden xestionarse empregando as seguintes directivas de configuración:

|  |  |
| --- | --- |
| Directiva de configuración de PHP | Significado |
| **file\_uploads** | Indica se se permiten ou non as subidas de ficheiros ao servidor. |
| **post\_max\_size** | Tamaño máximo da información transmitida co método POST. Afecta aos ficheiros e a calquera outra información transmitida por ese método. |
| **upload\_max\_filesize** | Tamaño máximo de cada un dos ficheiros que se suben ao servidor. |
| **upload\_tmp\_dir** | Directorio no que se almacenan de forma temporal os ficheiros que se suben ao servidor web. |
| **max\_file\_uploads** | Número máximo de ficheiros que se poden enviar de xeito simultáneo.. |

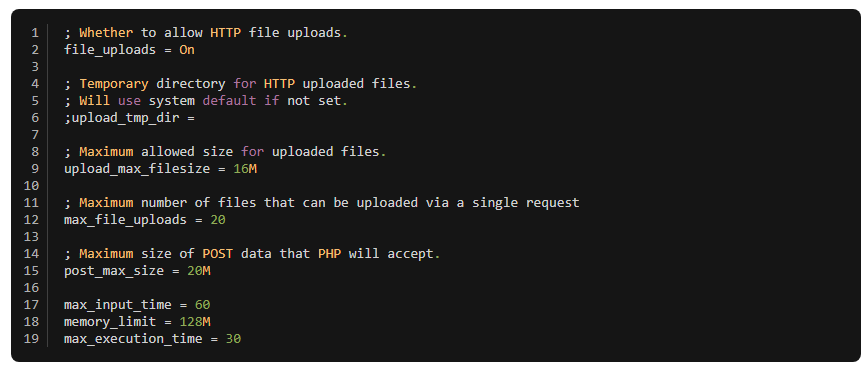
**Configurar las Opciones de PHP**

Hay un par de ajustes de configuración de PHP que querrás verificar de antemano para cargar archivos correctamente. En esta sección, veremos todas y cada una de las opciones que son importantes en lo que respecta a la carga de archivos PHP. Estas opciones se pueden configurar en el archivo **php.ini**.

Si no estás seguro de dónde encontrar tu archivo **php.ini**, puedes usar php\_ini\_loaded\_file() para localizarlo. Simplemente crea un archivo PHP en su servidor con la siguiente línea y ábrelo desde el navegador.



Aquí hay un extracto de un archivo de configuración con algunos valores predeterminados útiles.



**file\_uploads**

El valor de la directiva file\_uploads se debe establecer en On para permitir la carga de archivos. El valor predeterminado de esta directiva es On.

**upload\_max\_filesize**

La directiva upload\_max\_filesize te permite configurar el tamaño máximo del archivo cargado. De forma predeterminada, se establece en 2M (dos megabytes), y puede anular esta configuración utilizando el archivo **.htaccess** también. Dos megabytes no son mucho para los estándares de hoy, por lo que es posible que tenga que aumentar esto. Si recibes un error de cuando el archivo excede upload\_max\_filesize cuando intenta cargar un archivo, debes aumentar este valor. Si lo hacea, asegúrate de aumentar también post\_max\_size (ver más abajo).

**upload\_tmp\_dir**

Establece un directorio temporal que se utilizará para almacenar los archivos cargados. En la mayoría de los casos, no necesita preocuparse por esta configuración. Si no lo configuras, se utilizará el directorio temporal predeterminado del sistema.

**post\_max\_size**

La directiva post\_max\_size te permite configurar el tamaño máximo de los datos POST. Dado que los archivos se cargan con solicitudes POST, este valor debe ser mayor que lo que has establecido para la directiva upload\_max\_filesize. Por ejemplo, si tu upload\_max\_filesize es 16M (16 megabytes), es posible que desee establecer post\_max\_size en 20M.

**max\_file\_uploads**

Te permite establecer el número máximo de archivos que se pueden cargar a la vez. El valor predeterminado es 20, una cantidad razonable.

O nome do arquivo enviado recíbese no array global $\_FILES. Se o ficheiro se sube correctamente ao servidor, crearase un novo elemento no array $\_FILES co nome definido no atributo "name" do elemento "<input>", e cos seguintes membros.

|  |  |
| --- | --- |
| Membros dos elementos do array $\_FILES | Contido |
| **$\_FILES['imaxe']['name']** | Nome orixinal do ficheiro. |
| **$\_FILES['imaxe']['size']** | Tamaño do ficheiro. |
| **$\_FILES['imaxe']['type']** | Tipo MIME do ficheiro, tal e como o proporciona o navegador (por exemplo, "image/png"). Pode non corresponder co tipo real do ficheiro recibido, polo que debería verificarse. |
| **$\_FILES['imaxe']['tmp\_name']** | Nome temporal do ficheiro recibido no servidor. |
| **$\_FILES['imaxe']['error']** | Código de erro no caso dun envío incorrecto |

Defínense as seguintes constantes de erro para o membro **$\_FILES['imaxe']['error']** que indicarán se houbo ou non erro e o motivo do mesmo.

|  |  |
| --- | --- |
| Constante de erro | Significado |
| **UPLOAD\_ERR\_OK** | Arquivo subido correctamente. |
| **UPLOAD\_ERR\_INI\_SIZE** | Tamaño do arquivo superior ao definido na directiva upload\_max\_filesize. |
| **UPLOAD\_ERR\_FORM\_SIZE** | Tamaño do arquivo superior ao definido na directiva MAX\_FILE\_SIZE do formulario HTML. |
| **UPLOAD\_ERR\_PARTIAL** | Arquivo parcialmente subido. |
| **UPLOAD\_ERR\_NO\_FILE** | Non se recibiu o arquivo. |
| **UPLOAD\_ERR\_NO\_TMP\_DIR** | Non existe a directiva upload\_tmp\_dir que define a carpeta de almacenamento temporal. |
| **UPLOAD\_ERR\_CANT\_WRITE** | Non se puido escribir o arquivo na carpeta de almacenamento temporal. |
| **UPLOAD\_ERR\_EXTENSION** | Subida detida por unha extensión de PHP. |

Unha vez recibido correctamente o ficheiro ($\_FILES['imaxe']['error'] === UPLOAD\_ERR\_OK), este gárdase de xeito automático na ruta temporal definida pola directiva "upload\_tmp\_dir". Logo pode moverse a unha nova localización empregando a función "move\_uploaded\_file", indicando o nome do ficheiro temporal e a ruta/nome do novo destino. Por exemplo:

<?php

try {

// Comprobamos se existe o elemento no array $\_FILES

if (!isset($\_FILES['imaxe']['error']))

throw new RuntimeException('Produciuse un erro no envío do ficheiro.');

// Comprobamos que o código de erro sexa UPLOAD\_ERR\_OK

switch ($\_FILES['imaxe']['error']) {

case UPLOAD\_ERR\_OK: // Todo correcto

break;

case UPLOAD\_ERR\_NO\_FILE:

throw new RuntimeException('Non se recibiu o arquivo.');

case UPLOAD\_ERR\_INI\_SIZE:

case UPLOAD\_ERR\_FORM\_SIZE:

throw new RuntimeException('Tamaño do arquivo demasiado grande.');

default:

throw new RuntimeException('Erro descoñecido.');

}

// Renomeamos e movemos a imaxe recibida á súa localización definitiva

$res = move\_uploaded\_file($\_FILES['imaxe']['tmp\_name'], 'imagenes/foto.jpg');

if (!$res) throw new RuntimeException('A imaxe non puido ser movida.');

echo 'Imaxe subida correctamente.';

}

catch (RuntimeException $e) {

echo $e->getMessage();

}

?>

Envío simultáneo de varios ficheiros

O xeito máis sinxelo de subir varios arquivos cun único envío dun formulario é crear varios controis de tipo "<input> type='file'>" con distintos atributos "name".

<form enctype="multipart/form-data" action="envio.php" method="POST">

<p>Ficheiros a enviar: </p>

<input name="imaxe1" type="file" /><br />

<input name="imaxe2" type="file" /><br />

<input name="imaxe3" type="file" /><br />

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

</form>

Outra opción é crear un array de controis co mesmo nome:

<form enctype="multipart/form-data" action="envio.php" method="POST">

<p>Ficheiros a enviar: </p>

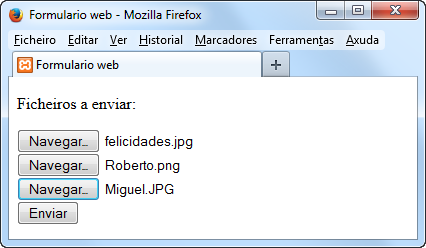
<input name="imaxe[]" type="file" /><br />

<input name="imaxe[]" type="file" /><br />

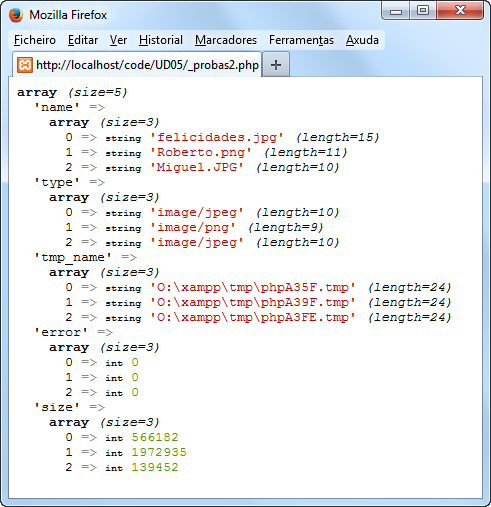
<input name="imaxe[]" type="file" /><br />

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

</form>



Neste caso, os membros da variable $\_FILES['imaxe'] serán todos eles arrays. Podemos comprobalo facendo var\_dump($\_FILES['imaxe']):



Existe unha variable de configuración para establecer un límite ao número máximo de ficheiros que pode enviar de xeito simultáneo un formulario web.

|  |  |
| --- | --- |
| Directiva de configuración de PHP | Significado |
| **max\_file\_uploads** | Número máximo de ficheiros que se poden enviar de xeito simultáneo.. |

Gestión de ficheros subidos a través de un formulario web

Tipos MIME <https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/MIME_Types>

El **tipo Extensiones multipropósito de Correo de Internet (MIME)** es una forma estandarizada de indicar la naturaleza y el formato de un documento, archivo o conjunto de datos. Está definido y estandarizado en [IETF RFC 6838](https://tools.ietf.org/html/rfc6838). La [Autoridad de Números Asignados de Internet (IANA)](https://www.iana.org/) es el organismo oficial responsable de realizar un seguimiento de todos los tipos MIME oficiales, y puede encontrar la lista más actualizada y completa en la página de [tipos de medios (Media Types)](https://www.iana.org/assignments/media-types/media-types.xhtml).

Los navegadores a menudo usan el tipo MIME (y no la extensión de archivo) para determinar cómo procesará un documento; por lo tanto, es importante que los servidores estén configurados correctamente para adjuntar el tipo MIME correcto al encabezado del objeto de respuesta.

Sintaxis

*tipo/subtipo*

La estructura de un tipo MIME es muy simple; consiste en un tipo y un subtipo, dos cadenas, separadas por un '/'. No se permite espacio. El tipo representa la categoría y puede ser de tipo discreto o multiparte. El subtipo es específico para cada tipo.

Un tipo MIME no distingue entre mayúsculas y minúsculas, pero tradicionalmente se escribe todo en minúsculas.

### Tipos discretos



### Tipos multiparte 🡪 *multipart/form-data*

Los tipos de *partes múltiples* indican una categoría de documento que está rota en distintas partes, a menudo con diferentes tipos de MIME. Es una forma de representar un documento compuesto. Con la excepción de multipart/form-data, que se utilizan en relación con formularios HTML y el método POST, y multipart/byteranges, que se utilizan para enviar solo un subconjunto de un documento completo, HTTP no maneja documentos multiparte de una manera específica: el mensaje simplemente se transmite al navegador (que probablemente propondrá una ventana Guardar como, sin saber cómo mostrar el documento en línea.)