

## Subida con el método POST

Esta característica permite la subida de ficheros de texto y binarios. Con la autenticación de PHP y las funciones de manipulación de ficheros se tiene control completo sobre quién está autorizado a realizar una subida y qué hay que hacer con el fichero una vez subido.

PHP es capaz de recibir subidas de ficheros de cualquier navegador compatible con el RFC-1867.

**Nota:** Nota sobre configuraciones relacionadas

Véase también las

directivas [file\\_uploads](#), [upload\\_max\\_filesize](#), [upload\\_tmp\\_dir](#), [post\\_max\\_size](#) y [max\\_input\\_time](#) de

PHP también admite el método PUT para subir ficheros tal y como lo utilizan los clientes Netscape Composer y Amaya del W3C. Véase el [Soporte para el método PUT](#) para más detalles.

### Ejemplo #1 Formulario para la subida de ficheros

Se puede construir una página de subida de ficheros creando un formulario especial parecido a este:

```
<!-- El tipo de codificación de datos, enctype, DEBE especificarse como
<form enctype="multipart/form-data" action="__URL__" method="POST">
  <!-- MAX_FILE_SIZE debe preceder al campo de entrada del fichero --
  <input type="hidden" name="MAX_FILE_SIZE" value="30000" />
  <!-- El nombre del elemento de entrada determina el nombre en el ar
  Enviar este fichero: <input name="fichero_usuario" type="file" />
  <input type="submit" value="Enviar fichero" />
</form>
```

El `__URL__` del ejemplo anterior se debe sustituir y debe apuntar a un fichero de PHP.

El campo oculto `MAX_FILE_SIZE` (medido en bytes) debe preceder al campo de entrada del fichero, siendo su valor el tamaño de fichero máximo aceptado por PHP. Se debe utilizar siempre este elemento del formulario, ya que evita a los usuarios la molestia de esperar a que un fichero grande sea transferido sólo para descubrir que falló la transferencia porque era demasiado grande. Hay que tener en cuenta que engañar a esta configuración en el lado del navegador es muy fácil; nunca dependa de que los ficheros que tengan un tamaño mayor sean bloqueados por esta característica. Es simplemente una característica conveniente para los usuarios en el lado cliente de la aplicación. No obstante, la configuración de PHP (en el lado del servidor) para un tamaño máximo no puede ser engañada.

**Nota:**

Asegúrese de que el formulario de subida de ficheros tiene el

atributo `enctype="multipart/form-data"` o de lo contrario la subida de ficheros no

funcionará.

El array global `$_FILES` contendrá toda la información de los los ficheros subidos. Su contenido en el formulario del ejemplo es el siguiente. Observe que se asume el empleo del nombre `fichero_usuario` para el fichero subido, tal como se utiliza en el script de ejemplo anterior. Este puede ser cualquier nombre.

`$_FILES['fichero_usuario']['name']`

El nombre original del fichero en la máquina del cliente.

`$_FILES['fichero_usuario']['type']`

El tipo MIME del fichero, si el navegador proporcionó esta información. Un ejemplo sería `"image/gif"`. Este tipo MIME, sin embargo, no se comprueba en el lado de PHP y por lo tanto no se garantiza su valor.

`$_FILES['fichero_usuario']['size']`

El tamaño, en bytes, del fichero subido.

`$_FILES['fichero_usuario']['tmp_name']`

El nombre temporal del fichero en el cual se almacena el fichero subido en el servidor.

`$_FILES['fichero_usuario']['error']`

El código de error asociado a esta subida.

Por omisión, los ficheros se almacenan en el directorio temporal predeterminado del servidor, a menos que se haya indicado otra ubicación con la directiva `upload_tmp_dir` en `php.ini`. Se puede cambiar el directorio predeterminado del servidor estableciendo la variable de entorno `TMPDIR` en el entorno en que se ejecuta PHP. Configurarlos usando `putenv()` desde un script de PHP no funcionará. Esta variable de entorno también se puede utilizar para asegurarse de que las demás operaciones están trabajando sobre los ficheros subidos.

## Ejemplo #2 Validación de la subida de ficheros

Véanse también las entradas de las funciones `is_uploaded_file()` y `move_uploaded_file()` para más información. El siguiente ejemplo procesará la subida de fichero desde un formulario.

```
<?php
$dir_subida = '/var/www/uploads/';
$fichero_subido = $dir_subida . basename($_FILES['fichero_usuario']['name']);

echo '<pre>';
if (move_uploaded_file($_FILES['fichero_usuario']['tmp_name'], $fichero_subido)) {
    echo "El fichero es válido y se subió con éxito.\n";
} else {
    echo "¡Posible ataque de subida de ficheros!\n";
}

echo 'Más información de depuración: ';
print_r($_FILES);
```

```
print "</pre>";

?>
```

El script de PHP que recibe el fichero subido debería implementar cualquier lógica necesaria para determinar qué se debe hacer con el fichero subido. Se puede, por ejemplo, utilizar la variable `$_FILES['fichero_usuario']['size']` para descartar cualquier fichero que sea demasiado pequeño o demasiado grande. Se podría utilizar la variable `$_FILES['fichero_usuario']['type']` para descartar cualquier fichero que no corresponda con un cierto criterio de tipo, aunque esto se debe emplear solo como la primera de una serie de comprobaciones debido a que este valor está completamente bajo el control del cliente y no se comprueba en el lado de PHP. También se puede usar `$_FILES['fichero_usuario']['error']` y planear la lógica de acuerdo con los [códigos de error](#). Cualquiera que sea la lógica, se debería borrar el fichero del directorio temporal o moverlo a otra ubicación.

Si no se selecciona ningún fichero en el formulario para realizar la subida, PHP devolverá `$_FILES['fichero_usuario']['size']` como 0, y `$_FILES['fichero_usuario']['tmp_name']` como ninguno.

El fichero será borrado del directorio temporal al final de la solicitud si este no ha sido movido o renombrado.

### Ejemplo #3 Subir un array de ficheros

PHP admite la [funcionalidad de array en HTML](#) incluso con ficheros.

```
<form action="" method="post" enctype="multipart/form-data">
<p>Imágenes:
<input type="file" name="imágenes[]" />
<input type="file" name="imágenes[]" />
<input type="file" name="imágenes[]" />
<input type="submit" value="Enviar" />
</p>
</form>
```

```
<?php
foreach ($_FILES["imágenes"]["error"] as $clave => $error) {
    if ($error == UPLOAD_ERR_OK) {
        $nombre_tmp = $_FILES["imágenes"]["tmp_name"][$clave];
        // basename() puede evitar ataques de denegación del sistema de ficheros;
        // podría ser apropiado más validación/saneamiento del nombre de fichero
        $nombre = basename($_FILES["imágenes"]["name"][$clave]);
        move_uploaded_file($nombre_tmp, "datos/$nombre");
    }
}
?>
```

Se puede implementar una barra de progreso de subida de ficheros con el [Progreso de subida en sesiones](#).