

UD-06b: Generación dinámica de imágenes

Desarrollo Web en Entorno Servidor

Curso 2020/2021

Subida y gestión de ficheros en el servidor

Subimos archivos al servidor con `<input type="file"... />` de HTML.

Tendremos en cuenta que:

- El elemento `<form>` debe tener el atributo `enctype = "multipart/form-data"` y el atributo `method = "post"`. No se puede usar get para enviar ficheros.
- El fichero tiene un límite de tamaño que fijaremos de dos formas diferentes:

Selecionar archivo Ningún arch...eleccionado

En el fichero de configuración `php.ini`

```
;;;;;;;;;;  
; File Uploads ;  
;;;;;;;;;;  
; Whether to allow HTTP file uploads.  
file_uploads = On  
; Temporary directory for HTTP uploaded files (will use  
; system default if not specified).  
upload_tmp_dir = "ruta"  
; Maximum allowed size for uploaded files.  
upload_max_filesize = 2M
```

A través de un campo oculto en el propio formulario denominado `MAX_FILE_SIZE`. El valor de este campo indicará el tamaño máximo en bytes. No puede ser mayor al especificado en `upload_max_filesize`.

```
<h1>Subida de ficheros</h1>  
<form enctype="multipart/form-data"  
      action="subida.php" method="post">  
  <input type="hidden" name="MAX_FILE_SIZE" value="1000000">  
  <label for="imagen">Seleccione el fichero .GIF a subir:</label>  
  <input name="imagen" type="file">  
  <input type="submit" value="Enviar">  
</form>
```

Subida y gestión de ficheros en el servidor

Al cargar un archivo, se guardará en una ubicación temporal indicada en la opción `upload_tmp_dir` del `php.ini`. Si no cambiamos de nombre el archivo o lo movemos a otra carpeta, este será eliminado.

Algunos posibles problemas al trabajar con archivos subidos al servidor son:

- 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 nuestro script lo excede durante la subida del fichero.
- La directiva `post_max_size` tiene un valor bajo y el fichero lo excede. Su valor debe ser mayor a `upload_max_filesize`.

Subida y gestión de ficheros en el servidor

Subida de ficheros desde PHP

A la hora de manejar la subida de archivos tendremos en cuenta que:

- Los datos procesados se almacenarán en la variable superglobal **\$_FILES**.
- En esta variable tendremos un array para los archivos subidos.
- Para acceder al array usaremos como clave el nombre que hayamos puesto en el atributo **name** del `<input type="file" name="imagen" ... />` en el formulario.

`$_FILES['imagen']['tmp_name']` → Lugar y nombre del archivo almacenado temporalmente.

`$_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 ".*

`$_FILES['imagen']['error']` → Código de error asociado al fichero subido.

subida.php

<?php

```
// Comprobamos si hay un error al subirlo
if ($_FILES['imagen']['error'] != UPLOAD_ERR_OK) {
    echo 'Error: ';
    switch ($_FILES['imagen']['error'])
    {
        case UPLOAD_ERR_INI_SIZE:
        case UPLOAD_ERR_FORM_SIZE:
            echo 'El fichero es demasiado grande';
            break;
        case UPLOAD_ERR_PARTIAL:
            echo 'El fichero no se ha podido subir entero';
            break;
        case UPLOAD_ERR_NO_FILE:
            echo 'No se ha podido subir el fichero';
            break;
        default:
            echo 'Error indeterminado';
    }
    // Comprobamos que el fichero es del tipo que esperamos
} elseif ($_FILES['imagen']['type'] != 'image/gif') {
    echo 'Error: No se trata de un fichero .GIF.';
}
```

UPLOAD_ERR_OK

Valor: 0; No hay error, fichero subido con éxito.

UPLOAD_ERR_INI_SIZE

Valor: 1; El fichero subido excede la directiva upload_max_filesize de `php.ini`.

UPLOAD_ERR_FORM_SIZE

Valor: 2; El fichero subido excede la directiva `MAX_FILE_SIZE` especificada en el formulario HTML.

UPLOAD_ERR_PARTIAL

Valor: 3; El fichero fue sólo parcialmente subido.

UPLOAD_ERR_NO_FILE

Valor: 4; No se subió ningún fichero.

```
// Si se ha podido subir, lo guardamos.
/* Indica si el archivo fue subido mediante HTTP POST. Esto es útil para intentar
asegurarse de que un usuario malicioso no ha intentado engañar al script. */
elseif (is_uploaded_file($_FILES['imagen']['tmp_name']) === true)
{
    /* Concatenamos la ruta destino y comprobamos si el nombre ya existe previamente,
    en cuyo caso añadiremos una marca de tiempo. */
    $nombre = './subidas/' . $_FILES['imagen']['name'];
    if (is_file($nombre) === true)
    {
        $nombre = time(). '_' . $nombre;
    }
    // Movemos el fichero a su nueva ubicación
    // move_uploaded_file ( string $filename , string $destination ) : bool
    if (move_uploaded_file($_FILES['imagen']['tmp_name'], $nombre))
    {
        // Mostramos el archivo gif subido
        header("Content-type: image/gif");
        $fp = fopen($nombre, 'rb');
        $contenido = fread($fp, filesize($nombre));
        fclose ($fp);
        echo $contenido;
    } else {
        echo 'Error: No se puede mover el fichero a su destino.';
    }
} else {
    echo 'Error: posible ataque. Nombre: ' . $_FILES['imagen']['name'];
}
```

'rb' → lectura binaria

<https://www.php.net/manual/es/function.fopen.php>

<http://seclists.org/bugtraq/2000/Sep/0055.html>

?>

Subida y gestión de ficheros en el servidor

Funciones de gestión de directorio en php

Podemos gestionar los archivos subidos mediante diferentes funciones de gestión de directorio, siempre que dispongamos de permisos.

- **resource opendir(string ruta [, resource contexto])** → abre un gestor de directorio a ser usado en llamadas posteriores (o FALSE si ocurre un error).
- **string readdir(resource gestor_dir)** → devuelve el nombre del siguiente archivo leído desde el directorio. Los nombres de archivo son devueltos en el orden en que son almacenados por el sistema de archivos (o FALSE si ocurre un error).
- **void closedir(resource gestor_dir)** → cierra el un gestor de directorio.

```
<?php
$dir = "imagenes/";
if (is_dir($dir)) {
    if ($lectura = opendir($dir) !== false) {
        while (($file = readdir($lectura)) !== false) {
            echo "nombre archivo: $file : ";
            echo "tipo archivo: ".filetype($dir.$file)."<br />";
        }
        closedir($lectura);
    }
}
```

?>

```
nombre archivo: foto1.jpg : tipo archivo: file
nombre archivo: foto2.jpg : tipo archivo: file
nombre archivo: foto3.jpg : tipo archivo: file
nombre archivo: foto4.jpg : tipo archivo: file
```

Subida y gestión de ficheros en el servidor

Otras funciones para trabajar con directorios son las siguientes:

- `bool is_file(string nombre_archivo)` → devuelve true si el archivo con el nombre dado existe y es un archivo regular (no es un directorio).
- `bool is_dir(string nombre_archivo)` → devuelve TRUE si el archivo con el nombre dado existe y es un directorio.
- `array scandir(string directorio)` → devuelve un array de archivos y directorios que se encuentran en el directorio especificado.
- `string basename(string path [, string sufijo])` → dada una cadena que contiene el path de un archivo, esta función devuelve el nombre base del archivo. Si el nombre del archivo termina con sufijo, este también será cortado del nombre del archivo.
- `string dirname(string path)` → dada una cadena conteniendo el path a un fichero, esta función devolverá el nombre del directorio.
- `bool unlink(string nombre_archivo [, resource contexto])` → elimina nombre_archivo. Devuelve true si todo se llevó a cabo correctamente o false en caso de error.

Subida y gestión de ficheros en el servidor

Crear imágenes en miniatura

Quizá necesitemos mostrar un gran número de imágenes en una sola página. HTML permite definir el tamaño con el que se ve una la imagen, pero el tamaño real no cambia. Lo ideal es crear dos versiones de cada fotografía: una en miniatura con poco peso en bytes y otra, la original, que puede ser mostrada cuando se haga clic en la imagen pequeña. Gracias a la librería GD podemos modificar el tamaño de las imágenes que tenemos en nuestro servidor.



<?php

```
// Parámetros de la función: nombre de la imagen, nombre final en local, ancho y alto
function miniatura($name_org, $name_dst, $ancho, $alto){

    // Separamos el nombre y la extensión en un array de 2 elementos
    $arrNombre = explode(".", $name_dst);
    $nombre = $arrNombre[0];
    $extension = $arrNombre[1];

    // Creamos una nueva imagen, para cada tipo de extensión posible
    if($extension=="jpg" || $extension=="jpeg") $source = imagecreatefromjpeg($name_org);
    elseif($extension=="png") $source = imagecreatefrompng($name_org);
    elseif($extension=="gif") $source = imagecreatefromgif($name_org);

    // Creamos el thumbnail vacio
    $thumb = imagecreatetruecolor($ancho, $alto);
    $ancho_orig = imagesx($source);
    $alto_orig = imagesy($source);

    // Copiamos la imagen en un thumbnail pasándole los parámetros necesarios
    imagecopyresampled($thumb,$source,0,0,0,0,$ancho,$alto,$ancho_orig,$alto_orig);

    // Exportamos al formato elegido (Con el formato jpg o jpeg podemos elegir calidad).
    if($extension=="jpg" || $extension=="jpeg") imagejpeg($thumb, $name_dst, 90);
    elseif($extension=="png") imagepng($thumb, $name_dst);
    elseif($extension=="gif") imagegif($thumb, $name_dst);

}
```





```
// Llamamos dos veces a la función
```

```
miniatura("muestra.jpg", "muestra100.jpg", 100, 100);
```

```
miniatura("muestra.jpg", "muestra256.jpg", 256, 256);
```

```
?>
```

```
<!doctype html>
```

```
<html>
```

```
  <head>
```

```
    <meta lang="es" />
```

```
    <meta charset="utf-8" />
```

```
    <title>Miniatura de imágenes con PHP</title>
```

```
  </head>
```

```
  <body>
```

```
    <p></p>
```

```
    <p></p>
```

```
  </body>
```

```
</html>
```

EJERCICIO

Crea una nueva página de nombre `ud06ej02.php` con título '*UD 6 - Ejercicio 2*'.

Diseña un formulario en el que se pueda seleccionar un archivo de imagen, de manera que cuando subamos un archivo, nos muestre 2 thumbnails; uno con dimensiones 120x120 y el otro thumbnail de 200x200 del archivo seleccionado.

Podrán subirse archivos `.jpg`, `.gif` y `.png`



Almacenamiento de imágenes en MySQL

A la hora de almacenar imágenes podemos hacerlo de dos formas:

- Usar el sistema de ficheros, guardando en la BD un texto con la ubicación en el servidor y el nombre de la imagen.
 - Tener disponibles las imágenes en carpetas del servidor permite usar de forma sencilla herramientas de redimensionado y retoque.
- Guardar la imagen en la BD. Para ello usamos el tipo de datos **BLOB**, usado por MySQL para almacenar datos binarios, como puede ser una imagen.
 - Tiene ventajas en cuestión de seguridad, muy interesante si no disponemos de permisos de acceso al sistema de ficheros en el servidor.
 - Tiene el inconveniente de la lentitud en las consultas a la tabla con imágenes, dificultad a la hora de recuperar las imágenes, sobrecarga del servidor, etc.

Ejemplo: con BD llamada *images_db* y una tabla *images_tabla*:

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	id 🗄️	int(11)			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	<u>imagenes</u>	longblob			No	Ninguna		
<input type="checkbox"/>	3	creado	datetime			No	Ninguna		

formulario-cargar.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Almacenar imagen en la base de datos MySQL usando PHP</title>
</head>
<body>
  <h1>Cargar y Almacenar imagen en MySQL</h1>
  <form name="formulario" method="post" action="cargar.php" enctype="multipart/form-data">
    Seleccione imagen a cargar: <input type="file" name="image">
    <button name="submit">Cargar Imagen </button>
  </form>
</body>
</html>
```

cargar.php

<?php

```
if(isset($_POST["submit"])){
    //Devolverá las dimensiones de un fichero.
    $revisar = getimagesize($_FILES["image"]["tmp_name"]);
    //Comprobamos si el usuario ha seleccionado alguna imagen
    if($revisar !== false){
        $image = $_FILES["image"]["tmp_name"];
        //file_get_contents: Devuelve un fichero completo a una cadena
        $imgContenido = addslashes(file_get_contents($image));

        //Conexión Mysql
        $host = 'localhost';
        $username = 'root';
        $password = '';
        $dbname = 'images_db';

        $db = new mysqli($host, $username, $password, $dbname);
        //Comprobar la conexión
        if($db->connect_error) die("Error: ".$db->connect_error);

        //Insertar imagen en la base de datos
        $insertar = $db->query("INSERT into images_tabla (imagenes, creado)
                                VALUES ('$imgContenido', now())");

        // Verificamos la subida del fichero
        if($insertar) echo "Archivo Subido Correctamente.";
        else echo "Ha fallado la subida, reintente nuevamente.";
    }else{
        echo "Por favor selecciona imagen a subir.";
    }
}
```

Devuelve un string con barras invertidas delante de los caracteres que necesitan ser escapados.

Estos caracteres son la comilla simple ('), comilla doble ("), barra invertida (\) y NUL (el byte NULL).

Un ejemplo del uso de `addslashes()` es cuando se introducen datos en un string que es evaluado por PHP. Por ejemplo, si *O'Reilly* se almacena en `$str`, es necesario escapar `$str`.

Por ejemplo: `eval("echo '".addslashes($str)."'");`

ver.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Ver imágenes</title>
</head>
<body>
  <h1>Ver imágenes de la BD</h1>
  <img src='vista.php?id=4'/><br/>
  <img src='vista.php?id=5'/><br/>
  <img src='vista.php?id=6'/><br/>
</body>
</html>
```

vista.php

```
<?php
if(!empty($_GET['id'])){
  $host = 'localhost';
  $username = 'root';
  $password = '';
  $dbname = 'images_db';

  $db = new mysqli($host, $username, $password, $dbname);

  if($db->connect_error){
    die("Conexión Fallida: " . $db->connect_error);
  }
  //Extraer imagen de la BD mediante GET
  $result = $db->query("SELECT imagenes FROM images_tabla
                      WHERE id = {$_GET['id']} ");

  if($result->num_rows > 0){
    $datosimagen = $result->fetch_assoc();

    //Mostrar Imagen
    header("Content-type: image/jpg");
    echo $datosimagen['imagenes'];
  }else{
    echo 'Lo siento, la imagen no existe.';
  }
}
?>
```


Almacenamiento de archivos pdf

formulario-subir-pdf.html

```
<!DOCTYPE>
<html>
<head>
  <meta charset="UTF-8" />
  <title>Subir PDF</title>
</head>
<body>
  <form action="subir-pdf.php" method="post" enctype="multipart/form-data">
    <input type="file" name="archivo" id="archivo" size="35" />
    <input type="submit" name="enviar" id="enviar" value="colgar" />
    <input type="hidden" name="action" value="upload" />
  </form>
</body>
</html>
```

Almacenamiento de archivos pdf

subir-pdf.php

Aquí es donde se produce la subida a la carpeta pdf, previamente creada.

<?php

```
$resultado = "";  
if ($_POST["action"] == "upload") {  
    // obtenemos los datos del archivo  
    $tamano = $_FILES["archivo"]['size'];  
    $tipo = $_FILES["archivo"]['type'];  
    $archivo = $_FILES["archivo"]['name'];  
    if ($archivo != "") {  
        // guardamos el archivo en la carpeta ficheros  
        $destino = "pdf/" . $archivo;  
        if ( copy($_FILES['archivo']['tmp_name'], $destino) )  
            echo "Archivo subido.";  
        else  
            echo "Error al subir el archivo";  
    }  
}
```

?>

EJERCICIO

Crea una nueva carpeta llamada **ud06ej03** que contendrá varios archivos.

Añade dos campos a la tabla **libros** mediante *PHPMYAdmin*:

- '**foto**' de tipo *longblob*, admite nulos, predeterminado NULL
- '**creado**' de tipo *datetime*, admite nulos, predeterminado NULL

Partiendo del ejercicio de los libros, añade al formulario de '*insertar libro*' un botón para **Seleccionar archivo**, así como una columna extra a la tabla titulada FOTO.



Id:
Título:
Autor:
Páginas:
Imagen: No se eligió archivo

ID	TITULO	AUTOR	PAGINAS	FOTO
1	aaa	jajaja	111	
2	eee	jejeje	222	
3	iii	jijiji	333	
4	ooo	jojojo	444	
5	uuu	jujuju	555	

El campo 'imagen' no es obligatorio, pero el resto sí (esto ya lo tenemos hecho).

Id:

Título:

Autor: <-- ¡Debes introducir un autor!

Páginas: <-- ¡Debes introducir el número de páginas!

Imagen: No se eligió archivo

Quando introduzcamos un libro con imagen, el libro se insertará en la BD con foto y una *thumbnail* se mostrará en la tabla:

Id:

Título:

Autor:

Páginas:

Imagen: frodo.jpg

Id:

Título:

Autor:

Páginas:

Imagen: No se eligió archivo

ID	TITULO	AUTOR	PAGINAS	FOTO
1	aaa	jajaja	111	
2	eee	jejeje	222	
3	iii	jijiji	333	
4	ooo	jojojo	444	
5	uuu	jujuju	555	
6	libro sin foto	anonimo	666	
7	libro con foto	ni idea	777	