# 1 Almacenamiento de imágenes en bases de datos con PHP

## 1.1 Sumario



# 2 Almacenamiento de imágenes en bases de datos con PHP

## 2.1 Introducción, método clásico de subida de archivos

A la hora de almacenar imágenes en un servidor, lo más normal es subir el fichero y almacenarlo en alguna carpeta determinada en nuestro servidor.

Véase un ejemplo rápido de cómo se pueden almacenar ficheros en el servidor:

**Ejemplo de función PHP para subir archivos al servidor: funciones.php**

```
<?php

/**
 * subir_fichero()
 *
 * Sube una imagen al servidor  al directorio especificado teniendo el Atributo 'Name' del campo archivo.
 *
 * @param string $directorio_destino Directorio de destino dónde queremos dejar el archivo
 * @param string $nombre_fichero Atributo 'Name' del campo archivo
 * @return boolean
 */
function subir_fichero($directorio_destino, $nombre_fichero)
{
    $tmp_name = $_FILES[$nombre_fichero]['tmp_name'];
    //si hemos enviado un directorio que existe realmente y hemos subido el archivo
    if (is_dir($directorio_destino) && is_uploaded_file($tmp_name))
    {
        $img_file = $_FILES[$nombre_fichero]['name'];
        $img_type = $_FILES[$nombre_fichero]['type'];
        echo 1;
        // Si se trata de una imagen
        if (((strpos($img_type, "gif") || strpos($img_type, "jpeg") ||
 strpos($img_type, "jpg")) || strpos($img_type, "png")))
        {
            //¿Tenemos permisos para subir la imágen?
            echo 2;
            if (move_uploaded_file($tmp_name, $directorio_destino . '/' . $img_file))
            {
                return true;
            }
        }
    }
    //Si llegamos hasta aquí es que algo ha fallado
    return false;
}
?>
```

**Ejemplo de formulario de envío: formulario.html**

```
<form id="form1" enctype="multipart/form-data" method="post" action="recepcion.php">
    <label>Imagen
        <input id="campofotografia" name="campofotografia" type="file" />
    </label>
    <input id="enviar" name="enviar" type="submit" value="Enviar" />
</form>
```

**Ejemplo de uso de la función de envío de ficheros: recepcion.php**

```
<?php
require_once 'funciones.php';

if(!empty($_POST)){
      if (subir_fichero('imagenes','campofotografia'))
         echo 'Archivo recibido correctamente';
}
?>
```

## 2.2 Requerimientos previos

Hemos visto como se pueden subir imágenes a una carpeta del servidor usando PHP. Lo que queremos hacer ahora es almacenar esa imagen en una base de datos en lugar de un directorio del servidor. Podemos almacenar cualquier tipo de archivo y el método a utilizar se puede emplear en cualquier tipo de base de datos, aunque nosotros lo vamos a hacer para MySQL.

Lo primero que tendríamos que hacer es definir unos campos en la base de datos para almacenar dicho contenido:

```
CREATE TABLE `imagenes` (
  `imagen_id` int(11) AUTO_INCREMENT PRIMARY KEY,
  `imagen` mediumblob,
  `tipo_imagen` varchar(30)
)
```

El campo mediumblob se emplea para almacenar datos binarios. Existen cuatro tipos de BLOB:

- **TINYBLOB**: permite hasta 255 caracteres, máximo 256Bytes.
- **BLOB**: permite hasta 65535 caracteres, máximo 65KB.
- **MEDIUMBLOB**: permite hasta 16777215 caracteres, máximo 16MB.
- **LONGBLOB**: permite hasta 4294967295 caracteres, máximo 4GB.

En el campo tipo_imagen almacenaremos el **MIME TYPE** de ese fichero.

Ésto es imprescindible para PHP, ya que cuando el servidor envía los datos binarios a un navegador web, tiene que indicar previamente de qué tipo de datos se tratan, sino el navegador web no sabría como representar esos datos.

Hay que tener en cuenta que en el campo imagen sólo se almacenará el contenido binario de las imágenes, nunca el nombre del fichero o la extensión del archivo; por eso es necesario guardar el mime type, para que PHP indique el tipo de datos que está enviando y así el navegador pueda interpretar correctamente esos datos y representarlos.

## 2.3 Formulario para subir imágenes

**formulario.html**

```
<form action="almacenar.php" method="POST" enctype="multipart/form-data">
    <label for="imagen">Imagen:</label>
    <input type="file" name="imagen" id="imagen" />
    <input type="submit" name="subir" value="Subir Imagen"/>
</form>
```

## 2.4 Almacenar un fichero en base de datos

**almacenar.php**

```
<?php
// Conexion a la base de datos
mysql_connect("servidor", "usuario", "contrasena") or die(mysql_error());
mysql_select_db("base_de_datos") or die(mysql_error());

// Comprobamos si ha ocurrido un error.
if (!isset($_FILES["imagen"]) || $_FILES["imagen"]["error"] > 0)
{
    echo "Ha ocurrido un error.";
```

```
}
else
{
    // Verificamos si el tipo de archivo es un tipo de imagen permitido.
    // y que el tamaño del archivo no exceda los 16MB
    $permitidos = array("image/jpg", "image/jpeg", "image/gif", "image/png");
    $limite_kb = 16384;

    if (in_array($_FILES['imagen']['type'], $permitidos) && $_FILES['imagen']['size'] <= $limite_kb * 1024)
    {

        // Archivo temporal
        $imagen_temporal = $_FILES['imagen']['tmp_name'];

        // Tipo de archivo
        $tipo = $_FILES['imagen']['type'];

        // Leemos el contenido del archivo temporal en binario.
        $fp = fopen($imagen_temporal, 'r+b');
        $data = fread($fp, filesize($imagen_temporal));
        fclose($fp);

        //Podríamos utilizar también la siguiente instrucción en lugar de las 3 anteriores.
        // $data=file_get_contents($imagen_temporal);

        // Escapamos los caracteres para que se puedan almacenar en la base de datos correctamente.
        $data = mysql_escape_string($data);

        // Insertamos en la base de datos.
        $resultado = @mysql_query("INSERT INTO imagenes (imagen, tipo_imagen) VALUES ('$data', '$tipo')");

        if ($resultado)
        {
            echo "El archivo ha sido copiado exitosamente.";
        }
        else
        {
            echo "Ocurrió algun error al copiar el archivo.";
        }
    }
    else
    {
        echo "Formato de archivo no permitido o excede el tamaño límite de $limite_kb Kbytes.";
    }
}
?>
```

## 2.5 Cómo mostrar una imagen almacenada en la base de datos

Para mostrar una imagen utilizamos la marca html IMG:

```
<img src="fotografia.jpg"/>
```

Pero en este caso la imagen la vamos a obtener de la base de datos.

Para ello, en lugar de poner el nombre de la imagen en SRC, tendríamos que poner un fichero PHP al que le pasamos el ID de la fotografía que queremos mostrar (o bien otro dato por el que queramos buscar esa fotografía) y que se encargará de devolvernos la imagen desde la base de datos.

Podría ser algo así:

```
<img src="obtenerfotografia.php?id=21"/>
```

**obtenerfotografia.php**

```
<?php
// Conexion a la base de datos
mysql_connect("servidor", "usuario", "contrasena") or die(mysql_error());
mysql_select_db("base_de_datos") or die(mysql_error());

if ($_GET['id'] > 0)
```

```
{
    // Consulta de búsqueda de la imagen.
    $consulta = "SELECT imagen, tipo_imagen FROM imagenes WHERE imagen_id={$_GET['id']}";
    $resultado = @mysql_query($consulta) or die(mysql_error());
    $datos = mysql_fetch_assoc($resultado);

    $imagen = $datos['imagen']; // Datos binarios de la imagen.
    $tipo = $datos['tipo_imagen'];  // Mime Type de la imagen.
    // Mandamos las cabeceras al navegador indicando el tipo de datos que vamos a enviar.
    header("Content-type: $tipo");
    // A continuación enviamos el contenido binario de la imagen.
    echo $imagen;
}
?>
```

--Veiga (discusión) 12:28 7 may 2013 (CEST)