UPLOAD – Subir Archivos al Server

Introducción

En este capítulo se aprenderá la forma de subir un archivo al servidor por medio del navegador. El único requisito para que todo funcione es que la carpeta donde se vaya a subir el archivo tenga los permisos adecuados, es decir, permisos de escritura en el servidor (hosting) donde se instalará la aplicación.

No es importante el tipo de archivo que se desea subir al servidor. Pueden ser archivos con extensión doc, pdf, xls, gif, jpg, etc.

El proceso de UPLOAD (subir un fichero al servidor) siempre se realiza a través de un formulario HTML, o sea, utilizando la etiqueta <form>. Dentro del mismo se utiliza la etiqueta HTML <input> con su atributo type establecido a "file" para especificar el archivo que se desea copiar desde el Cliente al Servidor.

A continuación, se explicará en detalle cada uno de los conceptos mencionados en la introducción y se realizará un practica integradora.

Formulario para subir archivos

Tipos de contenido del Formulario

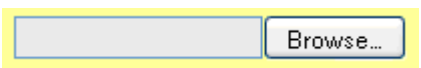
El atributo enctype del elemento FORM especifica el tipo de contenido usado para codificar el conjunto de datos del formulario para su envío al servidor.

application/x-www-form-urlencoded: (contenido por defecto) Los nombres de control y los valores se transforman en secuencias de escape. Los caracteres de espacio se sustituyen por '+', y los caracteres reservados se transforman en secuencias de escape. Los caracteres no alfanuméricos se reemplazan por '%HH', un signo de porcentaje y dos dígitos hexadecimales que representan el código ASCII del carácter.

multipart/form-data: Se utiliza para enviar formularios que contengan ficheros, datos no ASCII y datos binarios. El contenido "multipart/form-data" sigue las reglas de todos los flujos de datos MIME multiparte

Etiqueta FILE

El campo tipo file define un archivo que puede ser enviado junto con los datos del formulario. En este tipo de elemento encontramos asociados una caja de texto y un botón en el que encontramos escrito "examinar...","browse..." , dependiendo del lenguaje de nuestro navegador. Al ser pulsado se abre la típica ventana de exploración de unidades de disco para que se seleccione el archivo que se quiere enviar al servidor. Cuando se elige uno, su ruta aparece en la caja de texto.



Campo oculto MAX_FILE_SIZE

El campo de tipo hidden llamado MAX_FILE_SIZE sirve para indicar el tamaño en bytes de los archivos a subir. En este campo algunos navegadores no tienen porqué entenderlo o hacerle caso. Además, es fácil saltarse esa protección por lo que en las propias páginas PHP se deberá comprobar que el archivo tenga el tamaño que se desea.

Este campo es opcional pero recomendado para limitar el tamaño del archivo a transmitir. La macro MAX_FILE_SIZE se encargará de esa gestión.

Formulario completo de Carga

```
<form action="upload2.php" method="post" enctype="multipart/form-data">
    <b>Campo de tipo texto:</b>
    <br>
    <input type="text" name="cadenatexto" size="20" maxlength="100">
    <input type="hidden" name="MAX_FILE_SIZE" value="100000">
    <br>
    <br>
    <b>Enviar un nuevo archivo: </b>
    <br>
    <input name="userfile" type="file">
    <br>
    <input type="submit" value="Enviar">
</form>
```

Página Destino – Copiar al Servidor

La variable $_FILES

La variable de PHP llamada $_FILES (reemplazante de la obsoleta $HTTP_POST_FILES, la cual se utilizaba en la versiones anteriores a PHP 4.0.1) es fundamental para concretar el proceso de UPLOAD de un archivo desde el Cliente al Servidor Web.

La variable $_ FILES tiene el siguiente formato:

$_FILES["nombre_etiqueta_file"]["name"]: Nombre del arribo a subir

$_FILES["nombre_etiqueta_file"]["type"] - Tipo del arribo a subir

$_FILES["nombre_etiqueta_file"]["size"] – Tamaño en bytes del arribo a subir

$_FILES["nombre_etiqueta_file"]["tmp_name"] – Nombre temporal cuando se copia del arribo almacenado en el Servidor.

$_FILES["nombre_etiqueta_file"]["error"] – Código de error resultante del proceso de upload

$_FILES['userfile']['error']	Este valor ofrece información importante y pertinente para el resultado de la tentativa de carga. Son posible 5 valores de error: Uno significa un resultado exitoso y los otros cuatro son diferentes errores que surgen del intento.
$_FILES['userfile']['name']	Esta variable especifica el nombre original del archivo, incluida la extensión, como se declara en la máquina del cliente.
$_FILES['userfile']['size']	Esta variable especifica el tamaño en bytes del archivo enviado desde el equipo del cliente.
$_FILES['userfile']['tmp_name']	Esta variable especifica el nombre temporal asignado al archivo una vez que se ha cargado en el servidor. Este es un nombre asignado por PHP mientras que se almacena en el directorio temporal.
$_FILES['userfile']['type']	Esta variable especifica el tipo MIME del archivo enviado. Por lo tanto, en el caso de una imagen, a esta variable se le asigna el valor de la imagen / png. Si es un PDF, se asigna el valor de la aplicación / pdf.

“nombre_etiqueta_file” es el nombre de la etiqueta HTML:

<input type=”file” name =”nombre_etiqueta_file”>

En el siguiente ejemplo se muestra el uso de la variable $_FILE para subir archivos únicamente de imágenes y con un peso inferior a 20000 bytes.

<?php

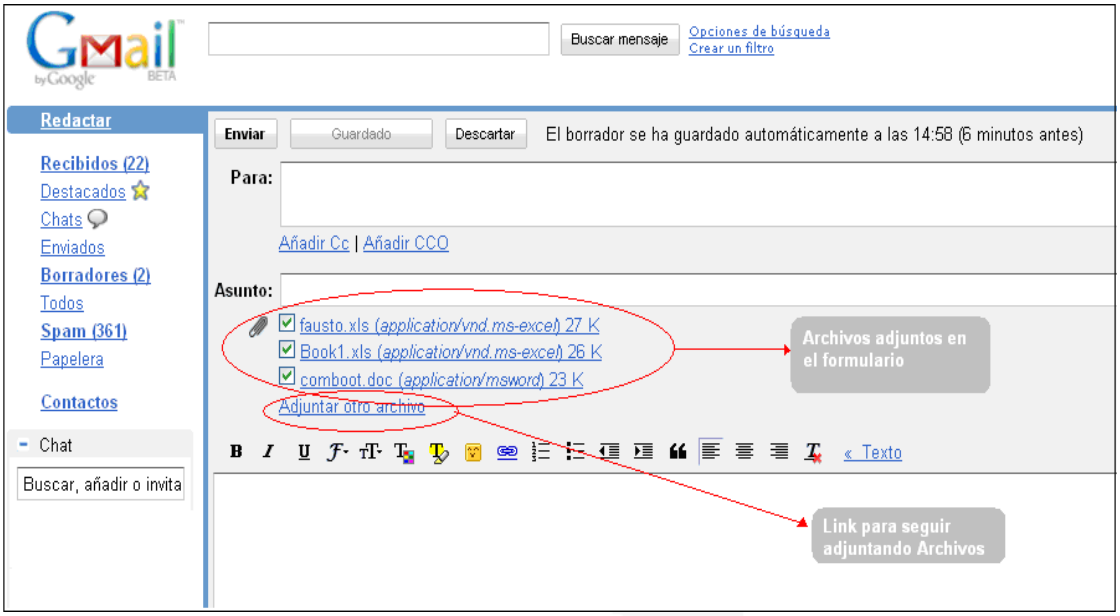
if ((($_FILES["file"]["type"] == "image/gif")

|| ($_FILES["file"]["type"] == "image/jpeg")

```
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
  {
  if ($_FILES["file"]["error"] > 0)
    {
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
    }
  else
    {
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
    }
  }
else
  {
  echo "Archivo Invalido";
  }

?>
```

En un formulario pueden haber varias etiquetas FILE. Eso significa que se pueden hacer varios UPLOADS en un mismo formulario de carga. A continuación, se muestra un ejemplo de la página www.gmail.com



Según el ejemplo anterior la variable $_FILES, quedaría cargada de la siguiente manera:

file	name	type	size	tmp_name	error
File1	Fausto.xls	xls	27000	Fausto.xls	0
File2	Book1.xls	xls	26000	Book1.xls	0
File3	comboot.doc	doc	23000	comboot.doc	0

En este caso la variable $_FILE actúa como matriz ya que tiene varias filas de archivos.

Grabando el archivo subido

Para guardar físicamente en el Servidor Web un archivo subido desde la máquina de un cliente, se utiliza la función “move_uploaded_file”. Su sintaxis es:

bool move_uploaded_file (string $nombre_archivo , string $destino)

Esta función realiza un chequeo para asegurar que el archivo indicado por nombre_archivo sea un archivo cargado válido (lo que quiere decir que fue cargado a través del mecanismo de carga HTTP POST de PHP). Si el archivo es válido, será movido al nombre de archivo dado por destino.

Este tipo de chequeo es especialmente importante si hay alguna chance de que cualquier cosa hecha con archivos cargados pueda revelar sus contenidos al usuario, o incluso a otros usuarios en el mismo sistema.

Continuando con el ejemplo anterior de subir una imagen al servidor, se le aplica la función “move_uploaded_file”

```
<?php

if (($_FILES["file"]["type"] == "image/gif")

|| ($_FILES["file"]["type"] == "image/jpeg")
```

```
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
  {
  if ($_FILES["file"]["error"] > 0)
    {
    echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
    }
  else
    {
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

    if (file_exists("upload/" . $_FILES["file"]["name"]))
      {
      echo $_FILES["file"]["name"] . " already exists. ";
      }
    else
      {
      move_uploaded_file($_FILES["file"]["tmp_name"],
      "upload/" . $_FILES["file"]["name"]);
      echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
      }
    }
  }
else
  {
```

```
  echo "Archivo Invalido";
  }
?>
```