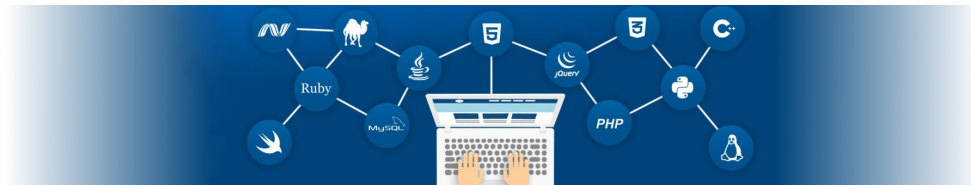


unidad didáctica 4

Ficheros en formularios



1. **ficheros en formularios**

- 1. 1. [etiqueta html para adjuntar fichero](#)
- 1. 2. [manejo de ficheros subidos con php](#)
- 1. 3. [filtrado con php de tipos de ficheros subidos con html](#)
- 1. 4. [escritura de imágenes en carpeta del servidor](#)
- 1. 5. [seguridad de escritura de imagenes en carpeta del servidor](#)
- 1. 6. [extra 1: mostrar imagenes subidas con html](#)
- 1. 7. [extra 2: descargar ficheros subidos con html](#)

1. ficheros en formularios

1.1. etiqueta html para adjuntar fichero

La etiqueta *input* se autoconfigurará para subir ficheros tan solo indicando *file* en el atributo *type*. Los navegadores identificarán inmediatamente este atributo y mostrarán gráficamente un botón para seleccionar archivos del sistema.

Por último añadir un botón de tipo *submit* para confirmar el envío del formulario al servidor.

```
<form name="subida-imagenes" type="POST" enctype="multipart/formdata" >
  <input type="file" name="imagen" />
  <input type="submit" name="subir-imagen" value="Enviar imagen" />
</form>
```


1.2. manejo de ficheros subidos con php

Una vez los usuarios de nuestro formulario confirmen el envío, en el servidor, en nuestra programación habitual de tratamiento de los datos enviados, tendremos acceso al fichero o ficheros que hayan adjuntado. Pero así como por norma general tenemos acceso a los datos recibidos mediante la variable superglobal `$_POST` o `$_GET` según el método del formulario, el acceso a ficheros será mediante el **array** también **superglobal** `$_FILES`.

Veamos de que información disponemos en el array `$_FILES` para una imagen llamada 'saludo.jpg' subida mediante nuestro formulario:

```
<?php
  echo $_FILES['imagen1']['name'];
  echo $_FILES['imagen1']['tmp_name'];
  echo $_FILES['imagen1']['type'];
  echo $_FILES['imagen1']['size'];
?>
```

Mediante estos *echo's* se ha accedido a toda la información disponible del fichero en PHP. La información impresa sería la siguiente:

```
saludo
213mnuashduahs0923
image/jpg
120304
0
```

Respectivamente estos valores del primero al último representan la siguiente información:

1. Nombre del fichero subido.
2. Nombre temporal asignado al fichero por el servidor. Este nombre es único y permite identificarlo dentro de la carpeta de temporales.
3. Tipo *MIME* de fichero subido: *img. png. gif. ndf. etc.*

4. Tamaño en *bytes* del fichero, si quisiéramos calcular el tamaño a *kilobytes* deberíamos dividir entre 1024 y para pasar a megabytes volver a dividirlo entre 1024.
5. Código de error de la subida, en nuestro caso 0 o UPLOAD_ERR_OK que indica que no se ha producido error alguno. [Códigos de error subida de fichero](#).

1.3. filtrado con php de tipos de ficheros subidos con html

Una vez sabemos cómo acceder a la información de los ficheros subidos, vamos a centrarnos en el filtrado de los tipos de ficheros aceptados. Limitar el tipo de fichero subido es altamente recomendable para evitar posibles problemas de seguridad.

Para este ejemplo voy comprobar que la imagen subida sea en efecto una imagen con una de las extensiones más comunes y que su tamaño sea menor a 1 MB:

```
<?php
$extensiones = ['image/jpg','image/jpeg','image/png'];
$max_tamanyo = 1024 * 1024 * 8;
if ( in_array($_FILES['imagen1']['type'], $extensiones) ) {
    echo 'Es una imagen';
    if ( $_FILES['imagen1']['size'] < $max_tamanyo ) {
        echo 'Pesa menos de 1 MB';
    }
}
?>
```

1.4. escritura de imágenes en carpeta del servidor

Una vez tengamos nuestros ficheros filtrados vamos a proceder a guardarlos de forma permanente en una carpeta de nuestro servidor.

Las imagenes o ficheros subidos mediante los formularios HTML son almacenados siempre en una carpeta temporal del sistema, por lo tanto deberemos moverlos para poder guardarlos permanentemente.

Para trasladar los ficheros de la carpeta temporal directamente a nuestra carpeta elegida usaremos la función [move_uploaded_file\(origen, destino \)](#).

El siguiente ejemplo sería un script alojado en la carpeta raíz de nuestra web, p.e. index.php:

```
<?php
$ruta_indexphp = dirname(realpath(__FILE__));
$ruta_fichero_origen = $_FILES['imagen1']['tmp_name'];
$ruta_nuevo_destino = $ruta_indexphp . '/uploads/' . $_FILES['imagen1']
['name'];

if ( in_array($_FILES['imagen1']['type'], $extensiones) ) {
    echo 'Es una imagen';
    if ( $_FILES['imagen1']['size'] < $max_tamanyo ) {
        echo 'Pesa menos de 1 MB';
        if( move_uploaded_file ( $ruta_fichero_origen, $ruta_nuevo_destino ) ) {
            echo 'Fichero guardado con éxito';
        }
    }
}
```

- El nombre temporal del fichero subido, que se encuentra en la carpeta de temporales, en `$ruta_fichero_origen`.
- La ruta completa de destino del fichero, que se compone por una parte de la ruta raíz del script donde estamos trabajando (estamos programando en `index.php`) más el nombre de la carpeta que se ha creado para guardar las imagenes (`uploads`) y por último el nombre definitivo que tendrá el fichero (*el nombre original del fichero*).

1.5. seguridad de escritura de imagenes en carpeta del servidor

Al guardar los archivos subidos por los usuarios en nuestro servidor, puede ocurrir que no filtremos los ficheros introducidos, o guardemos ficheros susceptibles de provocar problemas de seguridad. Para evitar problemas de este tipo lo mejor será incluir en la carpeta donde los almacenamos un pequeño *script htaccess* que evite la ejecución de código:

```
RemoveHandler .phtml .php3 .php .pl .py .jsp .asp .htm .shtml .sh .cgi .dat
RemoveType .phtml .php3 .php .pl .py .jsp .asp .htm .shtml .sh .cgi .dat
```

Con estas dos líneas en un fichero con extensión `.htaccess` evitaremos una posible ejecución de código por parte de usuarios malintencionados. Recuerda que debes incluir este fichero en la misma carpeta donde almacenas los ficheros.

Y ya está, con esto tendríamos terminado un **formulario para subir imagenes con php** totalmente funcional, con comprobaciones de seguridad para evitar subidas de ficheros inesperadas que puedan provocar problemas o hackeos inesperados.

1.6. extra 1: mostrar imagenes subidas con html

Mostrar las imagenes guardadas en nuestra carpeta de almacenamiento es sencillo, tan solo deberemos incluir la ruta hasta el fichero en una etiqueta `IMG` html:

```

```

1.7. extra 2: descargar ficheros subidos con html

Si queremos incluir un enlace de descarga para el fichero almacenado, en vez de utilizar una etiqueta `IMG` usaremos una etiqueta para enlaces con el atributo `HREF` la ruta al fichero:

```
<a href="uploads/nombreImagen.jpg"> Descarga de la imagen </a>
```

Este enlace producirá que el usuario descargue el fichero en cuestión, no obstante si el archivo es por ejemplo una imagen o pdf, el usuario en vez de lograr una descarga directa visualizará el contenido, teniendo que descargarlo haciendo uso de la opción descargar del menú desplegable con clic derecho.

Evitar la visualización de ficheros es posible gracias a HTML5 y los navegadores más modernos: Chrome, Firefox, Opera. Deberemos incluir el atributo `download` en la etiqueta de enlace anterior (A), así, el enlace final para una descarga forzada quedaría así:

```
<a href="uploads/nombreImagen.jpg" download="nombreImagen">Descarga la imagen</a>
```