





Objetivo de la clase


Aprender cómo subir
archivos a un servidor
mediante php.





Formularios


```
<form action="registrar.php" method="post">  
  Nombre: <input type="text" name="nombre">  
  Apellido: <input type="text" name="apellido">  
  <button type="submit">ENVIAR</button>  
</form>
```





Formularios

```
<form action="registrar.php" method="post">  
  Nombre: <input type="text" name="nombre">  
  Apellido: <input type="text" name="apellido">  
  Archivo: <input type="file" name="avatar">  
  <input type="submit" value="Enviar">  
</form>
```





Formularios

Nombre:

Apellido:

Archivo: No se eligió archivo





Formularios

```
<form action="" method="post" enctype="multipart/form-data">  
  Nombre: <input type="text" name="nombre">  
  Apellido: <input type="text" name="apellido">  
  Archivo: <input type="file" name="avatar">  
  <input type="submit" value="Enviar">  
</form>
```






El atributo `enctype` especifica cómo deben codificarse los datos de formulario al enviarlo al servidor.


Los valores que puede recibir el atributo:

- *`application/x-www-form-urlencoded`*
- *`text/plain`*
- *`multipart/form-data`*





¿Y ahora?
¿Cómo manejamos la
carga del archivo?





Uploads

PHP nos da ciertas facilidades para poder manejar la carga de archivos en formularios





La variable super global `$_FILES` contendrá toda la información de los archivos subidos a través del método POST.

En el caso del formulario anterior va a tener una clave llamada `archivo`.





Por cada archivo, contará con un array asociativo.

En este ejemplo, `$_FILES["avatar"]` es un array asociativo que contiene las siguientes claves:

- `error` - `$_FILES["avatar"]["error"]`
- `name` - `$_FILES["avatar"]["name"]`
- `tmp_name` - `$_FILES["avatar"]["tmp_name"]`
- `size` - `$_FILES["avatar"]["size"]`
- `type` - `$_FILES["avatar"]["type"]`





`$_FILES` – error

Indica si la imagen se subió exitosamente.

Habitualmente se compara con la constante `UPLOAD_ERR_OK` que indica que no hubo errores.

Se pueden ver más errores de subida [aquí](#).

```
<?php
    if($_FILES["avatar"]["error"] === UPLOAD_ERR_OK) {
        // ...
    }
?>
```





\$_FILES - name

Indica el nombre del archivo que se subió.

Adicionalmente, con `pathinfo` podemos obtener la extensión del archivo.

```
<?php
    $nombre = $_FILES["avatar"]["name"];
    $ext = pathinfo($nombre, PATHINFO_EXTENSION);
?>
```





\$_FILES - tmp_name

Representa la **ubicación real** en servidor del archivo subido.

Combinándolo con **move_uploaded_file** podemos guardar el archivo.

```
<?php
$nombre = $_FILES["avatar"]["name"];
$ext = pathinfo($nombre, PATHINFO_EXTENSION);
$archivo = $_FILES["avatar"]["tmp_name"];
move_uploaded_file($archivo, $nombre.".".$ext);
?>
```





dirname

Obtenemos la ruta actual del archivo que está siendo ejecutado.

```
<?php
    dirname(__FILE__);
?>
```





¡Subamos
un archivo!




```
<?php
```

```
function guardarArchivo() {
```

```
    if( $_FILES["avatar"]["error"] === UPLOAD_ERR_OK ) {
```

```
        $name = $_FILES["avatar"]["name"];
```

```
        $file = $_FILES["avatar"]["tmp_name"];
```

```
        $ext = pathinfo($name, PATHINFO_EXTENSION);
```


```
        $finalFile = dirname(__FILE__) . "/images/" . $name . "." . $ext;
```

```
        move_uploaded_file($file, $finalFile);
```


```
    }
```

```
}
```

```
?>
```

A cluster of hexagons in the top-left corner, featuring a large teal hexagon, a smaller teal one above it, a dark blue one to its right, and a light blue one to its left.

¿Y si se quiere
subir varios
archivos?

A cluster of hexagons in the bottom-right corner, featuring a dark blue hexagon, a teal one below it, and a light blue one to its right.

```
<input type="file" name="imagen1">  
<input type="file" name="imagen2">
```

\$_FILES

\$_FILES["imagen1"]

--	--	--	--	--

error name size type tmp_name

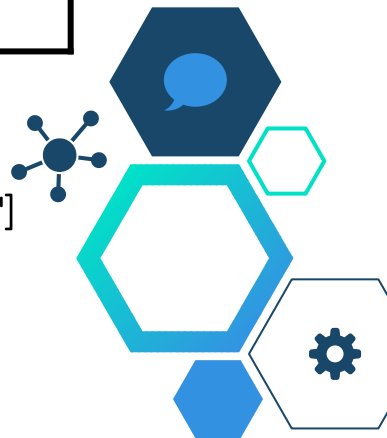
\$_FILES["imagen2"]

--	--	--	--	--

error name size type tmp_name

```
$_FILES["imagen1"]["error"]  
$_FILES["imagen1"]["name"]  
$_FILES["imagen1"]["tmp_name"]  
$_FILES["imagen1"]["size"]  
$_FILES["imagen1"]["type"]
```

```
$_FILES["imagen2"]["error"]  
$_FILES["imagen2"]["name"]  
$_FILES["imagen2"]["tmp_name"]  
$_FILES["imagen2"]["size"]  
$_FILES["imagen2"]["type"]
```



```
<input type="file" name="file[]" multiple>
```

`$_FILES["file"]`

<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>			<table><tr><td></td><td></td></tr></table>		
error	name	size	type	tmp_name										

`$_FILES["file"]["error"][0]`

`$_FILES["file"]["name"][0]`

`$_FILES["file"]["tmp_name"][0]`

`$_FILES["file"]["size"][0]`

`$_FILES["file"]["type"][0]`

`$_FILES["file"]["error"][1]`

`$_FILES["file"]["name"][1]`

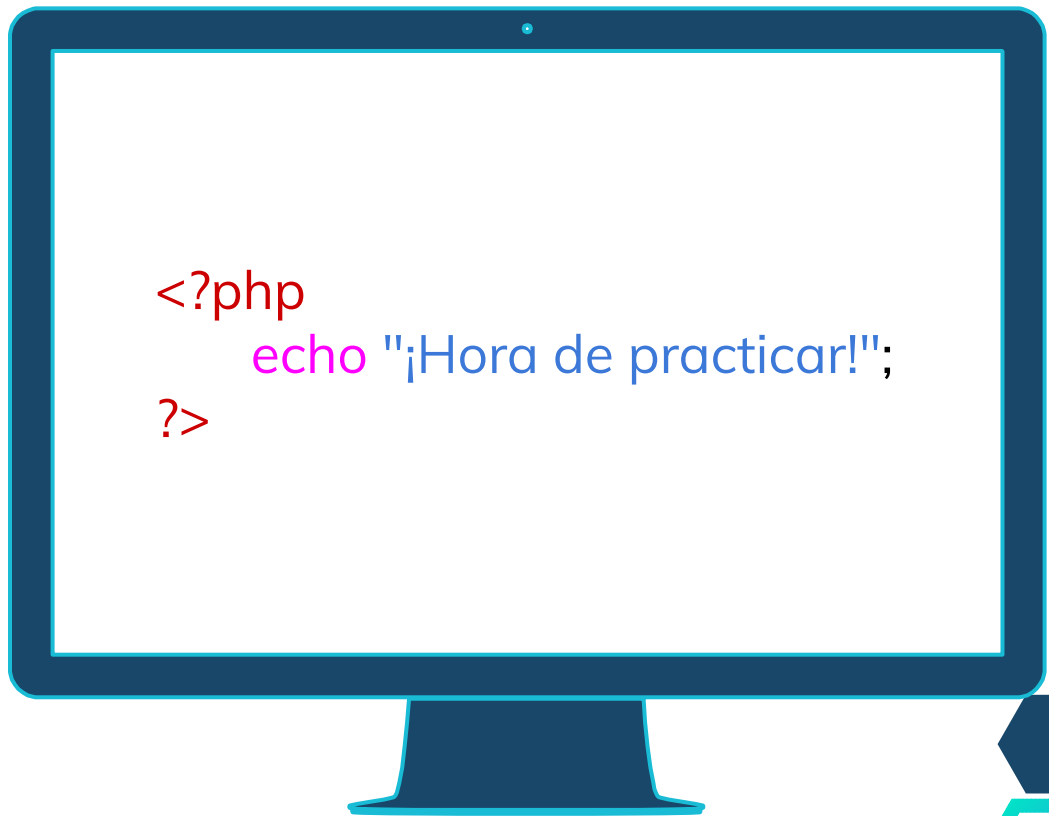
`$_FILES["file"]["tmp_name"][1]`

`$_FILES["file"]["size"][1]`

`$_FILES["file"]["type"][1]`



¡A practicar!





¿Qué aprendimos hoy?

- Subir archivos al servidor usando PHP.
- Mostrar archivos.
- Generar enlace de descarga del archivo subido al servidor.





Gracias!

¿Preguntas?

