php avanzado modulo 3

1. sesiones
2. cookies
3. formularios
4. validacion
5. sistema de archivos
6. subir ficheros
7. bloque de ejercicios

sesiones y cookies

sesiones

una sesion le permite estar a usuario dentro de la apliacion, hasta que finalice la sesion. pudiendo almacenar, consultar, eliminar, editar informacion

1. iniciar sesion con esta instruccion arrancamos la sesion, y con destroy cerramos

```
session_start();
session_destroy();
```

variable local

$variable_normal = "soy una cadena de texto";

variable de sesion

```
$_SESSION['variable_persistente'] = 'HOLA SOY UNA SESION ACTIVA';
```

con esta variable podemos estar en cualquier pagina de nuestra app, ya que esta dandome paso

ejemplo en session/

1. index.php
2. pagina1.php

3. logout.php

Lo que hace el index es crear una variable global, que me activa la session, permitiendome acceder a las demas paginas, pero si paso por la de logout.php, este archivo destruye la sesion dejandome sin acceso a las paginas

cookies en php

1. las cookies almacenan datos temporales del usuario, asi se cierre el navegador, recordando los inicios de sesion, acciones en la web, ideal para rastrear.
2. desventajas: limitadas, se almacenan en el disco duro

crear cookie

1. sintaxis

- setcookie("nombre_cookie", "valor que solo puede ser texto", caducidad, ruta, dominio);

1. ejercicioModulo3/cookies

formularios campos y atributos

los formularios permiten recoger datos del usuario y enviarlos al servidor

los formularios tiene un atriuto action en la etiqueta de form

```
<form action="" method="" enctype=""></form>
```

action: envia los datos al archivo que le pasemos, si no se le pasa nada, los envia al mismo archivo

method: metodo de envio POST O GET

enctype: permite enviar archivos desde el formulario hacia el servidor

dentro de los formularios se manejan los input

```
input type="text" name=""
```

1. type="text" //

2. type="button" // boton
3. type="color" // paleta de colores
4. type="checkbox" checked // seleccion
5. type="date" // calendario
6. type="datetime" //
7. type="datetime-local" //
8. type="email" // valida que se un correo
9. type="file" multiple // seleccionar archivo, multiples archivos
10. type="hidden" // oculta el campo, pero si envia dato
11. type="month" //
12. type="number" // contador de numero
13. type="password" // oculta paracteres por ****
14. type="radio" // seleccion un campo de varias opciones
15. type="range"
16. type="reset"
17. type="search"
18. type="submit"
19. type="tel"
20. type="text"
21. type="time"
22. type="url"
23. type="week"
24. name="campo" será en nombre con el que vamos a recoger el dato de ese campo.

```
$_POST['campo'];
```

atributos

1. autofocus="autofocus" autocarga el puntero
2. disabled="disabled" deshabilita
3. maxlength="5" maximo de letras en el campo
4. minlength="5" minimo de letras en el campo
5. pattern="[A-Z]" patron de letras expresiones regulares
6. required="required" requiere que el campo este lleno
7. placeholder="texto" texto indicativo
8. value="texto real"

tipos de campos

Explicacion de los campos linea 65-87

etiquetas, textarea y select

textarea, campo donde caben muchos caracteres

```
<textarea></textarea>
```

select

seleccion entre varias opciones disponibles

```
<select name="peliculas">
<option value="Spiderman">Spiderman</option>
<option value="Batman">Batman</option>
<option value="Wonder">Wonder Woman</option>
<option value="Joker">Joker</option>
<option value="Flash">Flash</option>
<option value="Joda">Joda</option>
</select>
```

como lo recibe el backend, con el value que se le de a cada campo

Validar Formularios

como se reciben los datos del formulario y se validan

metodo GET

url ***guardar.php¿titulo=Jared&&descripcion=enviando***

> index.html

```
<form action="guardar.php" method="GET">
<div>Titulo</div>
<input type="text" name="titulo">
<div>Descripcion</div>
<textarea name="descripcion"></textarea>
<input type="submit" value='enviar'/>
</form>
```

> guardar.php

```
<¿php

if (isset($_GET['titulo']) && isset($_GET['descripcion'])) {
  echo '<h1>'.$_GET['titulo'].'</h1>';
  echo '<h2>'.$_GET['descripcion'].'</h2>';
}
?>
```

en este ejercicio, enviamos por get datos a un archivo, pero debido a lo inseguro de este metodo ya que se puede alterar los datos en la url, no se recomienda sino para paginaciones o algo asi

metodo POST

el metodo post permite enviar los datos de una manera oculta, o directamente al archivo, algo mas seguro para envio de datos

```
<form action="guardar.php" method="POST">
```

```
<¿php

if (isset($_POST['titulo']) && isset($_POST['descripcion'])) {
  echo '<h1>'.$_POST['titulo'].'</h1>';
```

```
  echo '<h2>'.$_POST['descripcion'].'</h2>';
}
?>
```

formularios para validar

siempre se deben validar los datos antes de enviarlos al servidor si es posible, importantisimo y con php

```
<form action="procesarFormulario.php" method="POST">
  <label for="name">Nombre</label>
  <input type="text" name="name" required="required" pattern="[A-Za-z]+">

  <label for="lastName">Apellido</label>
  <input type="text" name="lastName" required="required" pattern="[A-Za-z]+">

  <label for="age">Edad</label>
  <input type="number" name="age" required="required" pattern="[0-9]+">

  <label for="email">Correo</label>
  <input type="email" name="email" required="required">

  <label for="pass">Contraseña</label>
  <input type="password" name="pass" required="required">

  <input type="submit" value="Cargar">
```

validar el formulario

modulo de validacion

resumen:

1. se crea el formulario
2. se validan los datos recibidos, que contengan informacion
3. luego se imprimen el pantalla para comprobar las variables
4. se validan las variables y el tipo de dato
5. se regresa el error y el tipo de error al formulario
6. validados los datos en backend
7. se activa validador html5 en frontend

8. listo

Sistema de archivos

ficheros

1. fopen("nombre","permisos"); *abre*
2. fgets *recorre linea*
3. feof($archivo) *recorrer fichero*
4. fwrite($archivo, "string a insertar");
5. fclose($archivo); *cierra el documento*

```
// abrir archivos r de read, a+ todos los permisos.
$archivo = fopen("archivo.txt","a+");

// leer contenido linea por linea
while (!feof($archivo)) {
    $contenido = fgets($archivo);
    echo $contenido."<br/>";
}

// escribir dentro del archivo
fwrite($archivo, "probemos con mas agregando datos al fichearo\n mientras escribimos texto

// cerrar archivo
fclose($archivo);
```

Manipular archivos

1. copiar un archivo

```
copy('archivo.txt', copia_archivo.txt) or die("Error al copiar");
```

2. renombrar archivo

```
rename('archivo.txt', 'archivito.txt');
```

3. eliminar archivo

```
unlink('archivo.txt') or die('Error al borrar');
```

Comprobar si existe un archivo

```
if(file_exists('archivo.txt')) {
  echo 'El archivo existe';
} else {
  echo 'No Existe';
}
```

Directorios

Crear carpeta

con mkdir creamos carpetas y el segundo argumento son de permisos

```
if (is_dir('nombreCarpeta')) {
  mkdir('nombreCarpeta', 0777) or die ('No se puede crear la carpeta');
} else {
  echo 'Ya existe la carpeta';}
```

eliminar directorio

```
rmdir('nombreCarpeta');
```

ver el contenido del directorio

de esta manera muestra el contenido del directorio en la pagina de php

```
if($gestor = opendir ('./micarpeta')) {
  while(false !== ($archivo = readdir($gestor))){
    if($archivo != '.' && $archivo != ''){
      echo $archivo.'<br/>';
    }
  }
}
```

subir archivos al servidor

se requiere el input de tipo file

enctype="multipart/form-data"

formulario

```
<form action="./upload.php" method="post" enctype="multipart/form-data">
  <input type="file" name="archivo" id="" />
  <input type="submit" value="enviar" />
</form>
```

$_FILES[''];

```
$archivo = $_FILES['archivo'];
var_dump($archivo);
die();
```

con esta configuración podemos seleccionar archivos

con var_dump(), miramos los datos que captura del archivo

```
array(5) { ["name"]=> string(10) "Apt404.png" ["type"]=> string(9) "image/png" ["tmp_name"]
```

1. comprueba si existe el directorio, si no lo crea

```
if (!is_dir('images')) {
  mkdir('images', 0777);
}
```

2. mueve el archivo temporal al directorio, ya que se guarda temporalmente antes de enviarlo al servidor

```
move_uploaded_file($archivo['tmp_name'], 'images/'.$nombre)
```

3. redirecciona al index despues de 5 segundos de haber mostrado un mensaje ok

```
header("Refresh: 5; URL=index.php");
```

mostrar imagenes cargadas

forma correcta de insertar php en html, endif; endwhile;

```
$gestor = opendir('./images');

if ($gestor):
  while(($image = readdir($gestor)) !== false):
    if ($image != '.' && $image != '..'):
      echo "<img src='images/$image' width='300px' /><br/>";
    endif;
  endwhile;
endif;
```

Ejercicios de desarrollo bloque 3

1. ejercicio 1 persistencia de datos

Crear una session que aumente su valor en uno o disminuya en funcion de si el parametro de get counter esta a uno o a cero.

en este ejercicio, persisten los datos, al ingresar a ejercicio1.alt.php, vemos como no cambia el valor ya que esta guardado en la session ($_SESSION['numero];).

2. ejercicio 2 validacion de email
3. debe de ser una funcion
4. validar un email con filter_var
5. recoger variable por get y validar
6. mostrar el resultado Se recibe un email por el metodo get para validarlo si el formato es un email valido o no ***nombre@dominio.com***, atraves de una funcion que hace la validacion

3. ejercicio 3 calculadora
4. creamos una interface de una calculadora

5. comprobamos si las variables post estan creadas
6. comprobamos si contienen datos
7. comprobamos si la variable post es que operacion + - * /
8. los name de los campos son los que nos permiten acceder al dato del input por post
9. se mantiene una variable para mostrar el resultado