

UNIDAD4.ACTIVIDAD2

Francisco Javier Signes Costa 2º DAW online

DESARROLLO

Esta actividad va a continuar el desarrollo que hemos creado en la unidad anterior.

Las páginas que hemos creado permiten ser accesibles a todos los usuarios por igual, pero no proporcionan un espacio que pueda ser utilizable por usuarios que se hayan identificado con credenciales.

Entre esta actividad y la siguiente vamos a crear un sistema de login que permita tener un espacio reservado para usuarios que estén registrados e identificados. Lee bien los requerimientos mínimos, en esta actividad vamos a crear los componentes necesarios para dotarlo de funcionalidad real en la siguiente unidad.

El proyecto que tienes que hacer tendrá los siguientes requerimientos mínimos:

- Crea una página web para establecer un sistema de login que contenga dos campos y un botón. Uno para el email, otro la contraseña y un botón para enviar el formulario.
- Cuando se presione enviar, la información será enviada mediante una petición HTTP de tipo POST a un controlador que tendrá un array con la información enviada. El array permitirá comprobar la información que se envíe desde el formulario, si son iguales, significa que el usuario y contraseña es correcto y si no son iguales es que la información enviada no pertenece al usuario guardado en el array.
- El controlador debe de contemplar los siguientes puntos:
 - Realizar validación de la información de los formularios
 - Comprobar si la información enviada es igual a la que contiene el array
 - Crear un estado de sesión personalizando la información guardada
 - Crear una cookie que contemple algún tipo de configuración personal
 - Crear un redireccionamiento a una página de bienvenida al usuario recién identificado, crea una página que pudiera servir de panel de administración de nuestro sitio web, con un botón que destruya la sesión y te reenvíe a la página principal de nuestro sitio
 - Crear un redireccionamiento al formulario informando que el usuario no se ha podido identificar correctamente e informando del error

Login.php

El archivo login.php es el archivo que nos da la bienvenida para que nos autentiquemos. Es un formulario tipo de HTML sin mayor complejidad.

```
1
2  <?php if (isset($_SESSION['error'])): ?>
3      <p class="error-message"><?php echo $_SESSION['error']; ?></p>
4      <?php unset($_SESSION['error']); ?>
5  <?php endif; ?>
6
7  <!-- Formulario para el login -->
8
9  <form action="procesar_login.php" method="POST">
10     <label for="email">Correo Electrónico:</label>
11     <input type="email" name="email" required>
12
13     <label for="password">Contraseña:</label>
14     <input type="password" name="password" required>
15
16     <button type="submit">Iniciar sesión</button>
17 </form>
18 </div>
```

En este fragmento también se procesa el mensaje de error por autenticación errónea. Recuperamos la variable error del archivo procesar_login.php y la metemos en un condicional if para que nos muestre en un párrafo <p> el mensaje en el formulario de logueo.

```
1  else {
2
3      // Si no coincide lo que el usuario teclea con lo que tenemos almacenado en las variables, nos salta este error
4      $_SESSION['error'] = 'Usuario o contraseña incorrectos';
5      header('Location: login.php');
6      exit;
7  }
8
```

Parte de procesar_login.php donde manejamos el error.

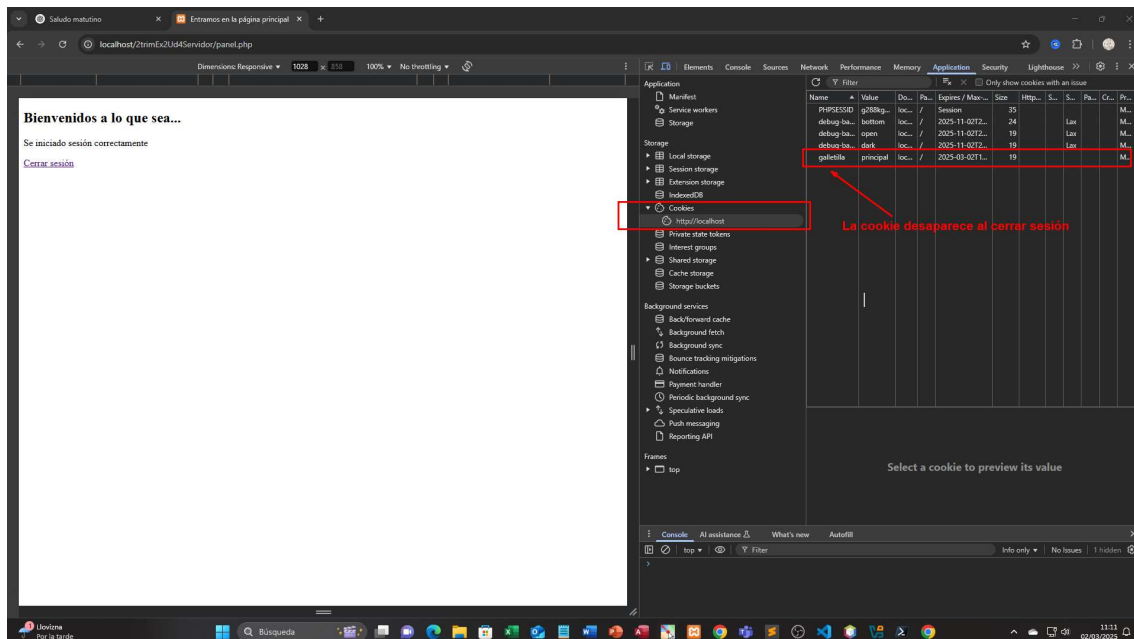
Logout.php

El ejercicio también pide salir de la sesión y destruir la cookie que hemos creado. Eso lo hacemos en este archivo.

```
1 <!-- Logout.php -->
2 <?php
3
4 //iniciamos sesión para poder tener acceso a cualquier variable de sesión
5 session_start();
6 //eliminamos las variables de sesión registradas pero la sesión sigue activa ojo!!
7 session_unset();
8 //nos cargamos la sesión y la eliminamos del servidor
9 session_destroy();
10
11 /* Borramos al cookie al finalizar sesión. Ponemos un valor en negativo para indicar que la cookie ya ha expirado.
12 De esa forma nos aseguramos que la borre. Se suele poner 1 hora (3600 segundos) pero podríamos poner cualquier valor, asegurándonos
13 que estemos en el "pasado"*/
14
15 setcookie('galletilla', 'principal', time() - 3600, '/');
16
17 // Volvemos a Login
18
19 header('Location: login.php');
20 exit;
21
```

Hacemos un `session_unset()` para eliminar las variables de sesión. Con `session_destroy()` la eliminamos del servidor.

La cookie la establecemos con un valor temporal negativo, en este caso una hora, para indicar su expiración. De esa manera se elimina automáticamente.



Procesar_login.php

Este es el archivo más importante de todos y es el encargado de dar validez al inicio de sesión de un usuario, almacenar su sesión y crear una cookie para mantener la autenticación un tiempo determinado (5 minutos en este ejercicio).

```
1 <!-- procesar_login.php -->
2 <?php
3 session_start();
4 include('functions.php'); //redirigimos a un archivo con una función para verificar el correo
5
6 // Aquí vamos a establecer un usuario ficticio para que se pueda loguear.
7 //Establecemos un array con dos parámetros (correo y contraseña)
8 $usuario = [
9     'email' => 'franciscosignes1978@gmail.com',
10    'password' => '123'
11 ];
12
13
14 /*Pasamos a validar el formulario. Con isset recogemos los datos introducidos en
15 los campos del formulario y los guardamos en $email y $password (variables)
16 para luego comprobarlas con el array que tenemos*/
17
18 if (isset($_POST['email']) && isset($_POST['password'])) {
19     $email = $_POST['email'];
20     $password = $_POST['password'];
21
22     /*En este segundo condicional anidado comprobamos que coincida lo
23     introducido el campo del formulario por el usuario con lo que
24     tenemos almacenado en las variables. Ponemos && para que ambas condiciones se cumplan*/
25
26     if ($email === $usuario['email'] && $password === $usuario['password']) {
27
28         // Almacenamos la sesión
29
30         $_SESSION['usuario'] = $email;
31
32         // Escribimos una cookie de sesión
33
34         setcookie('galletilla', 'principal', time() + (300), '/'); // 5 minutillos
35
36         // Damos acceso al panel, que es supuestamente la página de bienvenida o de acceso a la aplicación
37
38         header('Location: panel.php');
39         exit;
40
41     } else {
42
43         // Si no coincide lo que el usuario teclea con lo que tenemos almacenado en las variables, nos salta este error
44         $_SESSION['error'] = 'Usuario o contraseña incorrectos';
45         header('Location: login.php');
46         exit;
47     }
48
49 }
```

Iniciamos con `session_start` para poder usar `$_SESSION`.

Adjuntamos un archivo externo que contiene una función de validación del correo.

En un array asociativo definimos un usuario con dos claves: email y password.

Validamos el formulario contenido en login.php.

- Comprobamos que se han enviado los campos email y password a través del formulario `$_POST`.
- Si existen los guardamos en las variables `$email` y `$password`.

- Verificamos las credenciales con un condicional if. Ambas (&&) coincidencias han de darse (email y password).
- Si coincide lo que el usuario ingresa en el formulario con las variables que hemos definido, iniciamos sesión y almacenamos una cookie.
- Finalmente se redirecciona al usuario a panel.php (que sería la página principal)

Este es el funcionamiento básico del programa.

Functions.php

He visto en bibliografía que distintas partes del código, y dependiendo de la función que se les asigne, se suelen agrupar en archivos que luego se llaman cuando se necesitan. En este caso no era necesario, pero lo he hecho así porque dicen que son buenas prácticas. Aquí sólo tenemos una función que se encarga de comprobar la estructura de un correo electrónico.



```
1 <!-- functions.php -->
2 <?php
3 /*Función tipo regex para validar el formato de un correo electrónico común
4 pero que en PHP se llama filter_var. Le pasamos la variable $email y luego
5 filtramos con la filter_validate_email (filtro predefinido de php)*/
6 function validar_email($email) {
7     return filter_var($email, FILTER_VALIDATE_EMAIL);
8 }
9
```

Panel.php

Página a la que redireccionamos al usuario si se ha identificado correctamente. No tiene mayor misterio. Establecemos un enlace de cierre de sesión que nos redirige a login.php

```
1 <!-- panel.php -->
2 <?php
3 //iniciamos la sesión para poder acceder a la información del usuario. Necesario en cada archivo
4
5 session_start();
6
7 /*Aquí verificamos si la variable de sesión del usuario está establecida. Si no hemos iniciado sesión entramos en el if
8 por lo que nos redirige a login. No entraríamos a panel.php (página principal, digamos)*/
9
10 if (!isset($_SESSION['usuario'])) {
11     header('Location: login.php');
12     exit;
13 }
14 ?>
15
16
17 <!DOCTYPE html>
18 <html Lang="es">
19 <head>
20     <meta charset="UTF-8">
21     <title>Entramos en la página principal</title>
22 </head>
23 <body>
24     <h2>Bienvenidos a lo que sea...</h2>
25     <p>Se iniciado sesión correctamente</p>
26
27     <!-- Redireccionamos a logout.php -->
28     <a href="logout.php">Cerrar sesión</a>
29 </body>
30 </html>
31
```

Código completo

Te dejo el código en el repositorio de GitHub.

https://github.com/Frank512-lab/2T_EJERCICIOS_SERVIDOR/tree/main/UNIDAD%204/ACTIVIDAD%202/2trimEx2Ud4Servidor