



Ejercicios UT05

Vamos a trabajar con sesiones flash:

Una sesión flash es un dato guardado en `$_SESSION` que está pensado para usarse una sola vez y desaparecer automáticamente en la siguiente carga/redirección.

Se utiliza sobre todo para mensajes tipo:

- “Login correcto”
- “Credenciales incorrectas”
- “No tienes acceso a esta página”

Por qué existe:

Cuando haces un `header("Location: ...")` estás haciendo una redirección (cambia de página).

En esa redirección:

No puedes “arrastrar” fácilmente un mensaje en variables normales de PHP (porque ya es otra petición HTTP), pero sí puedes guardar el mensaje en `$_SESSION` antes de redirigir, y en la página destino lo lees y lo borras.

Comportamiento típico

En la página A:

- Guardas un mensaje flash en sesión.
- Rediriges a la página B.

En la página B:

- Lees el mensaje flash.
- Al leerlo, se elimina para que no vuelva a mostrarse si recargas.

Crear un formulario que permita a un usuario conectarse a una base de datos con los campos correo electrónico y contraseña.

- Si la conexión se ha realizado con éxito se muestra al usuario una nueva página web indicando que se ha realizado la conexión mostrando el id del usuario y la opción de cerrar sesión.
- Si el usuario cierra la sesión se elimina la información de sesión recogida del usuario.

Para llevar a cabo esta gestión se pide:

- Se debe configurar un archivo .env donde se encuentren definidas las variables DB_DSN, DB_USERNAME y DB_PASSWORD para realizar la conexión a una base de datos que debes crear denominada dwes_05_sesiones.
- Se debe crear una clase Conexión que siga el patrón singleton en el espacio de nombre App que atiende al directorio src
- Se debe crear un fichero helper.php en la raíz del proyecto, donde tendremos las siguientes funciones genéricas:
 - Definir una función flash, con dos parámetros de entrada de tipo cadena de caracteres con nombre clave y mensaje. Donde mensaje es un parámetro opcional y con valor por defecto nulo. La función devolverá nulo o un string.

El objeto de la función es crear una sesión flash. Denominamos sesión flash a la sesión que solo queremos que permanezca durante una redirección de páginas. La idea es que si existe en la sesión flash esa clave nos quedamos con su valor que devolvemos y eliminamos esa clave. Y si no existe la clave la creamos con su mensaje y retornamos el mensaje. Para distinguir estas sesiones flash a la hora de crear las claves serán \$_SESSION['flash'][\$clave]

\$_SESSION es un array global donde PHP guarda datos asociados a la sesión del usuario.

\$_SESSION['flash'] es un "sub-array" dentro de la sesión, reservado para mensajes flash.

[\$clave] es el índice concreto (por ejemplo "error", "email", "ok") dentro de ese sub-array.

Si llamas a la función flash...

```
flash('error', 'credenciales incorrectas');
```

Internamente la función hará...

```
$_SESSION['flash']['error'] = 'credenciales incorrectas';
```

- Definir la función iniciar_sesion, comprobará que si el estado de la sesión es igual PHP_SESSION_NONE inicia la sesión.

```
function iniciar_sesion(): void
{
    if (session_status() === PHP_SESSION_NONE) {
        session_start();
    }
}
```

- Definir una función estaLogueado, que devuelve un booleano si la sesión del usuario esta creada.
- Definir una función redireccionar, que tiene un parámetro de entrada con la url y realizar la redirección a esa página.
- Definir una función esPost, que devuelve un booleano cuando el método de la llamada ha sido por POST.

Además de la Clase Conexión tendremos una Clase Autenticarse con los siguientes métodos estáticos.

- Método inicializar, que llama a la función iniciar_sesion
- Método configurar, que crea la tabla de usuarios y llama al método privado crear Datos del usuario.
 - Método privado crearDatosUsuario, crea un usuario de ejemplo con el correo electrónico de educantabria y clave igual a password. La clave en base de datos se guarda aplicando una función hash utilizando la función de php password_hash (<https://www.php.net/manual/en/function.password-hash.php>)
- Método autenticar, verifica si el correo del usuario es un correo valido de nuestra base de datos. Para llevar a cabo este método debemos comprobar:
 - Que el método de la llamada sea Post (función esPost). No es una llamada post, utiliza la función flash creando una clave “error” y un mensaje “Método HTTP no permitido” realizando una redirección a la página index con la acción igual a paginaLogin.
- También comprobamos si el usuario esta ya logueado en el sistema (función estaLogueado). En este caso la redirección a la pagina index con la acción igual a paginaConectado.
 - Recogemos las variables post de correo y contraseña. Comprobamos en la base de datos si ese correo electrónico existe. Verificamos si la contraseña coincide utilizando la función de php password_verify (<https://www.php.net/manual/en/function.password-verify.php>)
 - Si todo está correcto reenviamos a la pagina de index con la acción página Conectado

- Si hay error, debemos utilizar la función flash para crear el error, con el mensaje “credenciales incorrectas” y el guardar en una variable el correo. Redireccionando a la página index con la acción paginaLogin.
- Método paginaConectado, verifica que el usuario esté conectado para poder mostrar la página una vez conectado.
 - No está conectado, utiliza la función flash para crear el error con el mensaje “No tienes acceso a esta página” y redirecciona a la página de index con la acción paginaLogin.
 - Si está conectado, redirecciona a la página paginaConectado.php
- Método desconectarse, se elimina la sesión de usuario. Destruye la sesión y redirecciona a la página de index con la acción paginaLogin.
- Método paginaLogin, verifica que el usuario no esté conectado para poder mostrar la página de login.
 - Si esta conectado. Redirecciona a la pagina de index con la acción paginaConectado
 - No esta conectado, redirecciona a la páginaLogin.php
- Método runAccion, es un método que controla en función de la variable get acción que función de la clase Autenticarse tiene que ejecutar. En función de la acción:
 - paginaLogin llama Autenticarse al método paginaLogin
 - autenticar llama Autenticarse al método autenticar
 - paginaConectado llama Autenticarse al método paginaConectado
 - desconectarse llama Autenticarse al método desconectarse

Si no existe la variable acción redireccionamos a la página index con la acción paginaLogin

Dentro de public tendremos las siguientes páginas:

- Index.php, es la página que determina el control de a que página hay que acceder. Recoge las variables de acceso a la base de datos y llama a los métodos de la clase autenticarse de inicializar, configurar y runAccion
- PaginaLogin.php, es el formulario de petición del correo electrónico y la contraseña.
 - LLama iniciar_sesion
 - Comprueba si esta logueado en ese caso redirecciona a la página index con la acción paginaConectado
 - Recupera de flash la clave del error y el correo. Y si hay error mostrará el mensaje.

- Que llama al index.php con la acción autenticar para realizar la validación de los datos.
- Que si ya hubiese tecleado el correo electrónico muestra su valor

Acceso

Correo electrónico

Contraseña

- PaginaConectado.php, es la página donde se muestra información del usuario conectado.
 - Llama iniciar_sesion
 - Comprueba que este logueado en caso contrario utiliza la función flash para crear el error y mensaje “No tienes acceso a esta página” y redirecciona a la página index con la acción paginaLogin
 - Si es correcto muestra, Te has conectado, un Hola tu id de usuario es y la acción de poderse desconectar – llama a la página index con la acción desconectarse

Te has conectado

Hola, tu id de usuario es 1

[Desconectarse](#)

Script de base de datos

```
CREATE DATABASE IF NOT EXISTS dwes_05_sesiones
```

```
CHARACTER SET utf8mb4
```

```
COLLATE utf8mb4_unicode_ci;
```

```
use dwes_05_sesiones;
```

```
CREATE TABLE IF NOT EXISTS usuarios (
```

```
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
```

```
user VARCHAR(190) NOT NULL UNIQUE,
```

```
password VARCHAR(255) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO usuarios (user, password)
```

```
VALUES (
```

```
'educantabria@example.com',
```

```
'$2y$10$ilSG4epjJ6ThrDNdzVUApeGgHq69NH2tEC6l1wP2HeIN/RpD9pcWq'
```

```
);
```

```
$email = 'educantabria@example.com';
```

```
$pass = 'password';
```