

# UNIDAD5.ACTIVIDAD2

Francisco Javier Signes Costa 2º DAW online

DESARROLLO

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

Tenemos desarrollado el diseño HTML, así como la infraestructura para poder crear un sistema de login. Pero ahora, debemos de crear unas bases de datos y permitir que se compruebe la información como se haría en una aplicación real.

Hasta ahora, tenemos una página para poder hacer login y una página que nos permitirá administrar nuestro sitio web, pero tenemos que crear una nueva página para que, en caso de que el usuario no tenga una cuenta, pueda crearla. Ten en cuenta que esto conlleva crear una nueva página HTML y la funcionalidad correspondiente en el controlador.

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

- Crea una página web que permita crear nuevas cuentas de usuario
- Crea la funcionalidad en el controlador
- Crea una base de datos con una tabla que contenga a los usuarios
- Crea un modelo que maneje la información de los usuarios
- Conecta los controladores que tienes desarrollados con el modelo y la base de datos

---

### *Introducción a la actividad*

---

Esta actividad es continuación de la que se hizo en el tema anterior (actividad 2 de la unidad 4). He reutilizado todo el código que he podido. De esa manera sólo voy a comentar los archivos que he añadido para conectarme a la base de datos y de esa manera no ser redundante. De todas formas, el código completo te lo dejo en el repositorio de GitHub para que lo puedas ver entero.

---

*Db.php*

---

A screenshot of a code editor with a dark background and light-colored text. The code is a PHP script for connecting to a MySQL database. It includes comments in Spanish and uses the mysqli extension. The code is numbered from 1 to 18 on the left side. The script defines variables for servername, username, password, and dbname, then creates a connection, checks for errors, and sets the charset to UTF-8.

```
1  <?php
2  // db.php
3  $servername = "localhost";
4  $username = "root";
5  $password = "";
6  $dbname = "mi_sistema_login";
7
8  // Crear conexión
9  $conn = new mysqli($servername, $username, $password, $dbname);
10
11 // Comprobar la conexión
12 if ($conn->connect_error) {
13     die("Conexión fallida: " . $conn->connect_error);
14 }
15
16 $conn->set_charset("utf8");
17 ?>
18
```

Este fragmento es el que nos permite la conexión a una base de datos MySQL utilizando PHP y la extensión mysqli.

En primer lugar definimos las credenciales de la base de datos. Estas son:

- Localhost: que es el servidor donde se encuentra la base de datos (usamos el mismo ordenador)
- Root: es el usuario predeterminado para XAMPP
- "": dejamos la contraseña vacía
- Mi\_sistema\_login: es el nombre de la base de datos

Todos estos datos los vamos almacenando en sus respectivas variables.

Creamos la conexión (conn) y le pasamos las cuatro variables. Lo podríamos haber escrito directamente, pero creo que así queda más limpio.

A continuación, comprobamos la conexión con un condicional if. Si no lo conseguimos, die() termina la ejecución del script y nos muestra el error.

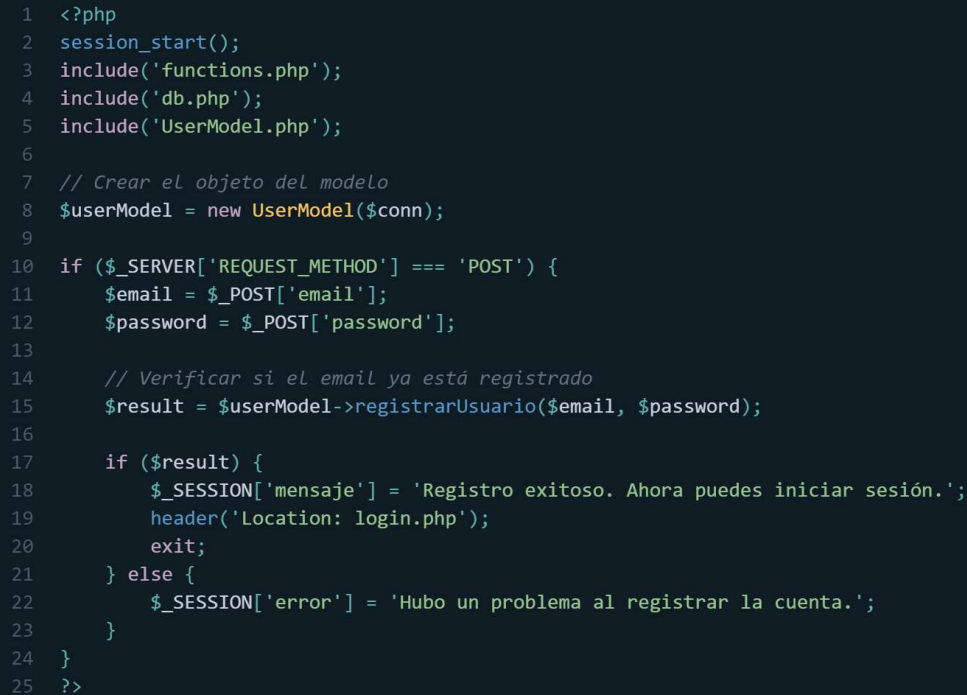
Finalmente configuramos el conjunto de caracteres UTF8 (así no nos da problemas, entre otros caracteres, nuestra querida ñ).

---

*Registro.php*

---

Este archivo es el encargado de manejar el registro de nuevos usuarios en el sistema. Procesa el formulario para registrar a los usuarios en la base de datos.



```
1 <?php
2 session_start();
3 include('functions.php');
4 include('db.php');
5 include('UserModel.php');
6
7 // Crear el objeto del modelo
8 $userModel = new UserModel($conn);
9
10 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
11     $email = $_POST['email'];
12     $password = $_POST['password'];
13
14     // Verificar si el email ya está registrado
15     $result = $userModel->registrarUsuario($email, $password);
16
17     if ($result) {
18         $_SESSION['mensaje'] = 'Registro exitoso. Ahora puedes iniciar sesión.';
19         header('Location: login.php');
20         exit;
21     } else {
22         $_SESSION['error'] = 'Hubo un problema al registrar la cuenta.';
23     }
24 }
25 ?>
```

Primero incluimos unos archivos externos.

Instanciamos un objeto de la clase UserModel que es la que maneja las operaciones sobre la base de datos relacionadas con los usuarios. Le pasamos como parámetro la conexión (conn).

Procesamos el formulario verificando que se ha enviado con el método POST y recoge los datos.

Llamamos al método registrarUsuario para verificar si existe el email y la password. El resultado lo guardamos en \$result.

Si \$result es true nos redirige a login.php y nos muestra un mensaje ('Registro exitoso. Ahora puedes iniciar sesión'). En caso contrario nos muestra un mensaje de error.

---

### *UserModel.php*

---

Este archivo sirve de capa intermedia entre el código principal y la base de datos. Definimos la clase UserModel que se encarga de gestionar las operaciones relacionadas con los usuarios en la base de datos.

Lo primero que hacemos es definir una clase a la que llamamos UserModel. La variable \$conn la ponemos como privada para que sólo se pueda tener acceso desde la clase.

Definimos el constructor que recibe la conexión como parámetro y la asigna a \$this->conn. Esto permite que podamos usar la conexión en toda la clase.

A continuación, debemos establecer el modo de registrar un nuevo usuario. Lo hacemos con una función que llamamos registrarUsuario(). En esta ocasión ya ciframos con bcrypt (hash) la contraseña a la hora de almacenarla en la base de datos.

Escribimos el código SQL para ingresar los datos. Prevenimos inyecciones SQL con (?) y con ss indicamos que van a ser valores de tipo string (s).

Ejecutamos la consulta retornando true si se inserta correctamente.

Con la función verificarUsuario buscamos un usuario en la base de datos cuyos datos coincidan con los que ha introducido el usuario en el login.php. Si lo encuentra lo convertimos en un array asociativo y contrasta la información que recibe con la que tiene almacenada en la base de datos. Si todo es correcto, retorna \$usuario. Si nos retorna false es porque o bien la contraseña es incorrecta o el usuario no existe (o no coincide).

---

### *Código completo*

---

Víctor, te dejo el código completo en el repositorio.

[https://github.com/Frank512-lab/2T\\_EJERCICIOS\\_SERVIDOR/tree/main/UNIDAD%205/ACTIVIDAD%202](https://github.com/Frank512-lab/2T_EJERCICIOS_SERVIDOR/tree/main/UNIDAD%205/ACTIVIDAD%202)

```
1  <?php
2  // UserModel.php
3  class UserModel {
4      private $conn;
5
6      public function __construct($db) {
7          $this->conn = $db;
8      }
9
10     // Método para registrar un nuevo usuario
11     public function registrarUsuario($email, $password) {
12         // Hashear la contraseña antes de almacenarla
13         $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
14
15         // SQL para insertar el nuevo usuario
16         $sql = "INSERT INTO usuarios (email, password) VALUES (?, ?)";
17         $stmt = $this->conn->prepare($sql);
18         $stmt->bind_param("ss", $email, $hashedPassword);
19
20         if ($stmt->execute()) {
21             return true;
22         }
23         return false;
24     }
25
26     // Método para verificar si el usuario existe
27     public function verificarUsuario($email, $password) {
28         $sql = "SELECT * FROM usuarios WHERE email = ?";
29         $stmt = $this->conn->prepare($sql);
30         $stmt->bind_param("s", $email);
31         $stmt->execute();
32         $result = $stmt->get_result();
33
34         if ($result->num_rows > 0) {
35             $usuario = $result->fetch_assoc();
36             // Verificar la contraseña
37             if (password_verify($password, $usuario['password'])) {
38                 return $usuario;
39             }
40         }
41         return false;
42     }
43 }
44
```

## BBDD

The screenshot displays the phpMyAdmin interface for a database named 'mi\_sistema\_login'. The 'usuarios' table is selected, showing 5 rows of data. The data is highlighted with a red box. The XAMPP Control Panel v3.3.0 is also visible, showing the status of services like Apache, MySQL, FileZilla, Mercury, and Tomcat.

	id	email	password	creado_en
<input type="checkbox"/>	1	franciscosignes1976@gmail.com	\$2y\$10\$WpU3SsJCeyHLEaD8OTR2Pj68V8FvLSOMFKQ...	2025-02-24 17:56:23
<input type="checkbox"/>	3	antonio@gmail.com	\$2y\$10\$OjNCqWjH0PqYYH09Qey5TL5d1Gt1GsePqkHP3...	2025-02-27 18:20:17
<input type="checkbox"/>	4	pedro@gmail.com	\$2y\$10\$nuD7KEMuAgOQpyGajrG3d0rQVwEIR6CJBXUCL...	2025-02-27 18:39:26
<input type="checkbox"/>	7	fosca@gmail.com	\$2y\$10\$M2q4gG63G9B0kyjP3FwRH9H9RL51n1cgAKS3...	2025-02-28 16:28:01
<input type="checkbox"/>	8	pancho@gmail.com	\$2y\$10\$E4N9ILK1t1BIPgkxmuHfGmuq3HwMT8qMTDAJH...	2025-02-28 16:29:50

La exporto y la pongo también en GitHub.