

- [Entrega 5: Acceso a datos](#)
 - Historia de php y sus versiones
 - [P1_mysqli](#)
 - [P2_mysqli](#)
 - Insertar
 - Update
 - Delete
 - SelectUpdate
 - Inyección
 - [P3_PDO](#)
 - [P4_login](#)
 - [P5_CRUD_MVC](#)
 - [P6_CRUD_SQLITE](#)
 - [P7_files](#)
 - Ejemplo1: escribir y leer un archivo
 - Ejemplo2: Lectura rápida
 - Ejemplo3: Añadir texto sin borrar archivo
 - Ejemplo 4: Crear y descargar un CSV
 - Ejemplo 5: Exportar a PDF con FPDF
 - Ejemplo 6: Exportar el contenido de un fichero de texto a PDF

Entrega 5: Acceso a datos

Historia de php y sus versiones



En el siguiente enlace podemos conocer un poco sobre la historia de php junto con sus diferentes versiones, desde php 3 hasta la más reciente php 8:

<https://www.php.net/manual/es/history.php.php>

Como conclusión personal podemos saber que PHP pasó de ser una herramienta simple a un lenguaje clave del desarrollo web gracias a su evolución constante y apoyo comunitario. Su historia demuestra que la adaptación es esencial para perdurar en la tecnología.

P1_mysqli

Nos crearemos una base de datos dentro de PHPMYADMIN llamada dwes, dentro de ella crearemos más adelante las demás tablas

Crearemos ahora el archivo de configuración de conexión a la base de datos

```

ConexionMysqli.php
<?php
// 1. CONEXIÓN con la BBDD:
// "SERVIDOR", "USUARIO", "CONTRASEÑA", "BASE DE DATOS"

try {
    //https://www.php.net/manual/es/function.mysql-connect.php
    $conexion = mysqli_connect(hostname: "localhost", username: "root", password: "", database: "dwes");
} catch (Exception $e) {
    echo "<br> 1. Se ha producido el siguiente error: " . $e->getMessage();
    echo "<br> 2. Falló la conexión: " . mysqli_connect_error();
    exit();
}

// 3. CONSULTA A LA BASE DE DATOS
echo "<h3>Listado de usuarios</h3>";
$consulta = "SELECT * FROM `ud4_persona`";
$listadoUsuarios = mysqli_query(mysql: $conexion, query: $consulta);
print_r(value: $listadoUsuarios);
echo "<br>";
// 4. COMPROBAMOS SI EL SERVIDOR NOS HA DEVUELTO RESULTADOS
if ($listadoUsuarios) {

    // RECORREMOS CADA RESULTADO QUE NOS DEVUELVE EL SERVIDOR
    foreach ($listadoUsuarios as $usuario) [
        echo "
            $usuario[id]
            $usuario[nombre]
            $usuario[apellidos]
            $usuario[telefono]
            <br>
        ";
    ]
    You, 2 weeks ago • ENtrega 5 actualizada y video subido
}

// 5. CERRAMOS LA CONEXIÓN A LA BASE DE DATOS

mysqli_close(mysql: $conexion);

?>

```

Crearemos además una tabla persona

```

CREATE TABLE `ud4_persona` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `nombre` tinytext NOT NULL,
    `apellidos` text NOT NULL,
    `telefono` tinytext NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `ud4_persona` (`id`, `nombre`, `apellidos`, `telefono`)
VALUES
    (1,'Pedro','Ramirez','123'),
    (2,'Juan','Serrano','456'),
    (3,'Rafa','López','789'),
    (6,'Antonio','Gómez','01254');      You, 2 weeks ago • ENtrega 5 actualizada y video subido

```

Opciones extra

			id	nombre	apellidos	telefono	
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	Pedro	Ramirez	123
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	Juan	Serrano	456
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	Rafa	Lopez	789
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	6	Antonio	Gomez	01254



Seleccionar todo

Para los elementos que están marcados:



P2_mysql

Crearemos ahora una tabla empresa, con el archivo de conexión a base de datos, sobre esta tabla haremos un CRUD completo

```
-- Crear base de datos (si no existe) y usarla
CREATE DATABASE IF NOT EXISTS dwes CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
USE dwes;

-- Crear tabla solicitada
DROP TABLE IF EXISTS ud4_empresa;
CREATE TABLE ud4_empresa (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    cargo VARCHAR(50),
    salario DECIMAL(10,2) DEFAULT 0.00,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- (Opcional) Datos de ejemplo
INSERT INTO ud4_empresa (nombre, cargo, salario) VALUES
('Ana López', 'Desarrolladora', 1800.50),
('Luis Pérez', 'Técnico Soporte', 1400.00);
```

```

<?php
/*
$cn = mysqli_connect("localhost", "root", "", "dwes");
if (!$cn) { die("Error de conexión: " . mysqli_connect_error()); }
mysqli_set_charset($cn, "utf8mb4");

mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
*/
4 references
function db(): mysqli {
    static $cn = null;
    if ($cn instanceof mysqli) {
        return $cn;
    }
    $host = "localhost";
    $user = "root";
    $pass = "";
    $db   = "dwes";           // <-- Base de datos: dwes

    $cn = mysqli_connect(hostname: $host, username: $user, password: $pass, database: $db);
    mysqli_set_charset(mysql: $cn, charset: "utf8mb4");
    echo "Conexión OK <br>";      You, 2 weeks ago • ENtrega 5 actualizada y video subido ...
    return $cn;
}

// mysqli_close($cn);

?>
<?php
/* ===== conexion_mysqli.php =====
   Conexión centralizada a MySQL (BD: dwes) con MySQLi en modo estricto.
   Incluye charset utf8mb4. Reutiliza esta función en todos tus scripts.
*/

```

Insertar

Vamos ahora a insertar una persona

```

<?php
/* ===== insertar_multiples_tx.php =====
|   INSERT atómico de 2 filas en dwes.ud4_empresa (o ninguna si falla).
*/
require_once __DIR__ . '/conexion_mysqli.php';
$cn = db();
//echo "<br>Conexión establecida.<br>";

// bloque try-catch-finally para manejar la transacción
try {
    // Inicio de la transacción
    mysqli_begin_transaction(mysql: $cn);

    // Preparar la sentencia INSERT con marcadores de posición: ?, ?, ?
    $stmt = mysqli_prepare(mysql: $cn,
        query: "INSERT INTO ud4_empresa (nombre, cargo, salario) VALUES (?, ?, ?)");
};

// 1ª fila
$nombre = "maria"; $cargo = "ceo"; $salario = 5000.00;

// Vincular los parámetros y ejecutar la sentencia
mysqli_stmt_bind_param(statement: $stmt, types: "ssd", var: &$nombre, vars: &$cargo, $salario);

// Ejecutar el INSERT
mysqli_stmt_execute(statement: $stmt);

// 2ª fila
$nombre = "juan"; $cargo = "ceo"; $salario = 4999.00;
mysqli_stmt_bind_param(statement: $stmt, types: "ssd", var: &$nombre, vars: &$cargo, $salario);
mysqli_stmt_execute(statement: $stmt);

mysqli_stmt_close(statement: $stmt);
mysqli_commit(mysql: $cn);
echo "Insertadas 2 filas en dwes.ud4_empresa.";

} catch (Throwable $e) {
    mysqli_rollback(mysql: $cn);
    echo "Insert cancelado: " . $e->getMessage();
} finally {
    mysqli_close(mysql: $cn);
}

```

Conexión OK

Insertadas 2 filas en dwes.ud4_empresa.

				id	nombre	cargo	salario	created_at
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	Luis Pérez	Técnico Soporte	1600.00	2025-10-29 11:06:49
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	maria	ceo	5000.00	2025-10-29 11:21:20
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	juan	ceo	4999.00	2025-10-29 11:21:20
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5	maria	ceo	5000.00	2025-11-12 10:06:09
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	6	juan	ceo	4999.00	2025-11-12 10:06:09

Update

En este caso vamos a updatear el salario de uno de ellos

```
<?php
mysqli_report(flags: MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
$cn = mysqli_connect(hostname: "localhost",username: "root",password: "",database: "dwes");
mysqli_set_charset(mysql: $cn, charset: "utf8mb4");

$id = 6; // empleado objetivo
$nuevo = 5000.00;

try {
    mysqli_begin_transaction(mysql: $cn);

    $stmt = mysqli_prepare(mysql: $cn,
        query: "UPDATE ud4_empresa SET salario = ? WHERE id = ?");
    mysqli_stmt_bind_param(statement: $stmt, types: "di", var: &$nuevo, vars: &$id);
    mysqli_stmt_execute(statement: $stmt);

    if (mysqli_stmt_affected_rows(statement: $stmt) !== 1) {
        throw new RuntimeException(message: "No se actualizó exactamente 1 fila.");
    }

    mysqli_stmt_close(statement: $stmt);      You, 2 weeks ago • ENtrega 5 actualizada y video
    mysqli_commit(mysql: $cn);
    echo "✅ Salario actualizado.";
} catch (Throwable $e) {
    mysqli_rollback(mysql: $cn);
    echo "❌ Update revertido: " . $e->getMessage();
} finally {
    mysqli_close(mysql: $cn);
}
```



Salario actualizado.

<input type="checkbox"/>		Editar		Copiar		Borrar	5	maria	ceo	5000.00	2025-11-12 10:06:09
<input type="checkbox"/>		Editar		Copiar		Borrar	6	juan	ceo	5000.00	2025-11-12 10:06:09

Delete

Vamos a eliminar ahora a un usuario de la tabla

```
...<?php
/* ===== borrar_empleado_tx.php =====
   DELETE seguro bajo transacción (valida filas afectadas = 1).
*/
require_once __DIR__ . '/conexion_mysqli.php';
$cn = db();

// Empleado a borrar
$id = 2;

try {
    mysqli_begin_transaction(mysql: $cn);

    $stmt = mysqli_prepare(mysql: $cn, query: "DELETE FROM ud4_empresa WHERE id =? ");
    mysqli_stmt_bind_param(statement: $stmt, types: "i", var: &$id);
    mysqli_stmt_execute(statement: $stmt);

    if (mysqli_stmt_affected_rows(statement: $stmt) !== 1) {
        throw new RuntimeException(message: "No se eliminó exactamente 1 fila ({ID inexistente}).");
    }

    mysqli_stmt_close(statement: $stmt);
    mysqli_commit(mysql: $cn);
    echo "Empleado eliminado de dwes.ud4_empresa.";
} catch (Throwable $e) {
    mysqli_rollback(mysql: $cn);
    echo "Delete revertido: " . $e->getMessage();
} finally {
    mysqli_close(mysql: $cn);
}
?>
```



Conexión OK

Empleado eliminado de dwes.ud4_empresa.

				id	nombre	cargo	salario	created_at
<input type="checkbox"/>	Editar	Copiar	Borrar	3	maria	ceo	5000.00	2025-10-29 11:21:20
<input type="checkbox"/>	Editar	Copiar	Borrar	4	juan	ceo	4999.00	2025-10-29 11:21:20
<input type="checkbox"/>	Editar	Copiar	Borrar	5	maria	ceo	5000.00	2025-11-12 10:06:09
<input type="checkbox"/>	Editar	Copiar	Borrar	6	juan	ceo	5000.00	2025-11-12 10:06:09

Hemos eliminado al usuario 2

SelectUpdate

Aquí buscamos al usuario completo por el id y luego actualiza el campo que se desee, en este caso el salario

```

<?php
/* ===== select_for_update_tx.php =====
| Lectura con bloqueo de fila (SELECT ... FOR UPDATE) + actualización consistente.
*/
require_once __DIR__ . '/conexion_mysqli.php';
$cn = db();

//selecciona el id para encontrar y actualizar
$id = 2;

try {
    mysqli_begin_transaction(mysql: $cn);

    // 1) Leer con bloqueo
    $stmt = mysqli_prepare(mysql: $cn,
        query: "SELECT salario FROM ud4_empresa WHERE id = ? FOR UPDATE"
    );
    mysqli_stmt_bind_param(statement: $stmt, types: "i", var: &$id);
    mysqli_stmt_execute(statement: $stmt);
    $res = mysqli_stmt_get_result(statement: $stmt);
    $row = mysqli_fetch_assoc(result: $res);
    if (!$row) {
        throw new RuntimeException(message: "Empleado no existe (id=$id).");
    }

    echo "Hemos leído el salario con FOR UPDATE.<br>";
    echo "Salario actual: " . $row['salario'] . "<br>";
    // Obtener salario actual
    $salarioActual = (float)$row['salario'];
    mysqli_free_result(result: $res);
    mysqli_stmt_close(statement: $stmt);

    // 2) Calcular nuevo salario (ej. +100€)
    $nuevo = $salarioActual +100.00; ;

    // 3) Actualizar
    $stmt = mysqli_prepare(mysql: $cn,           You, 2 weeks ago • ENTrega 5 actualizada y video subido
        query: "UPDATE ud4_empresa SET salario = ? WHERE id = ?"
    );
    mysqli_stmt_bind_param(statement: $stmt, types: "di", var: &$nuevo, vars: &$id);
    mysqli_stmt_execute(statement: $stmt);
    mysqli_stmt_close(statement: $stmt);

    mysqli_commit(mysql: $cn);
    echo "Salario actualizado con FOR UPDATE en awes.ud4_empresa.";
} catch (Throwable $e) {
    mysqli_rollback(mysql: $cn);
    echo "Error transaccional: " . $e->getMessage();
} finally {
    mysqli_close(mysql: $cn);
}

```



Conexión OK

Error transaccional: Empleado no existe (id=2).

Nos dice que el usuario 2 no existe así que vamos a probar con el usuario 6

Conexión OK

Hemos leído el salario con FOR UPDATE.

Salario actual: 5000.00

Salario actualizado con FOR UPDATE en dwes.ud4_empresa.

			id	nombre	cargo	salario	created_at
<input type="checkbox"/>	Editar	Copiar	3	maria	ceo	5000.00	2025-10-29 11:21:20
<input type="checkbox"/>	Editar	Copiar	4	juan	ceo	4999.00	2025-10-29 11:21:20
<input type="checkbox"/>	Editar	Copiar	5	maria	ceo	5000.00	2025-11-12 10:06:09
<input type="checkbox"/>	Editar	Copiar	6	juan	ceo	5100.00	2025-11-12 10:06:09

Seleccionar todo Para los elementos que están marcados: Editar Copiar

Como vemos se le han sumado 100 euros a su salario

Inyección

```

rou, 2 weeks ago | author (rou)
<?php
// conexión_mysqli.php debe proporcionar db()
require_once 'conexión_mysqli.php';
$cn = db();

// Supongamos un formulario que envía $_GET['nombre']
//$nombre = $_GET['nombre'] ?? '';
//$/nombre= "' OR '1'='1";
$nombre= "maria";
    You, 2 weeks ago • ENtrega 5 actualizada y video subido ...

// Vulnerable: concatenación directa de usuario en la consulta
$sql = "SELECT id, nombre, cargo, salario FROM ud4_empresa WHERE nombre = '$nombre'";

$res = mysqli_query(mysql: $cn, query: $sql);
while ($fila = mysqli_fetch_assoc(result: $res)) {
|   echo htmlspecialchars(string: $fila['id']) . ' - ' . htmlspecialchars(string: $fila['nombre']) . "<br>";
}
echo "<br> Se han mostrado todos los empleados si la inyección tuvo éxito.<br>";

// Cierre de conexión
mysqli_close(mysql: $cn);
?>

```



localhost/DWES/Dwes_Maria/UD4AccesoADatos/Entrega

Conexión OK

3 - maria

5 - maria

Se han mostrado todos los empleados si la inyección tuvo éxito.

P3_PDO

Vamos a crear la tabla musica para poder leerla con el PDO

```

-- Recreate table
DROP TABLE IF EXISTS ud4_musica;
CREATE TABLE ud4_musica (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(150) NOT NULL,
    artist VARCHAR(100) NOT NULL,
    genre VARCHAR(50),
    release_year YEAR,
    duration TIME, -- song duration
    unit_price DECIMAL(5,2) DEFAULT 0.00,
    quantity_in_stock INT UNSIGNED DEFAULT 0,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Sample data
INSERT INTO ud4_musica (title, artist, genre, release_year, duration, unit_price, quantity_in_stock)
VALUES
    ('Bohemian Rhapsody', 'Queen', 'Rock', 1975, '00:05:55', 1.29, 15),
    ('Shape of You', 'Ed Sheeran', 'Pop', 2017, '00:03:53', 0.99, 20),
    ('Billie Jean', 'Michael Jackson', 'Pop', 1982, '00:04:54', 1.19, 12),
    ('Smells Like Teen Spirit', 'Nirvana', 'Grunge', 1991, '00:05:01', 1.09, 10),
    ('Imagine', 'John Lennon', 'Rock', 1971, '00:03:07', 0.89, 25);

```

Generamos la conexión

```

<?php
/* conexión centralizada con PDO (MySQL/MariaDB) */
0 references
function pdo(): PDO {
    static $pdo = null;
    if ($pdo instanceof PDO) return $pdo;

    $host = 'localhost';
    $db   = 'dwes';
    $user = 'root';
    $pass = '';
    $charset = 'utf8mb4';

    $dsn = "mysql:host=$host;dbname=$db;charset=$charset";
    $opts = [
        PDO::ATTR_ERRMODE          => PDO::ERRMODE_EXCEPTION, // excepciones
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,           // arrays asociativos
        PDO::ATTR_EMULATE_PREPARES  => false,                      // prepara nativos
    ];
    $pdo = new PDO($dsn, $user, $pass, $opts);
    return $pdo;
}

?>      You, 2 weeks ago • Entrega 5 actualizada y video subido ...

```

Y ahora lo lista de una forma muy simple

You, 2 weeks ago | 1 author (You)

```
<?php
/* leer_musica.php - Listado simple con PDO (MySQL/MariaDB) */

$host = 'localhost';
$db   = 'dwes';
$user = 'root';
$pass = '';
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";

try {
    // Conexión PDO
    $pdo = new PDO($dsn, $user, $pass, [
        PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
        PDO::ATTR_EMULATE_PREPARES   => false,
    ]);
}

// Consulta (lee toda la tabla)
$sql = "SELECT id, title, artist, genre, release_year, duration, unit_price, quantity_in_stock
        FROM ud4_musica
        ORDER BY id DESC";
$stmt = $pdo->query($sql);
$rows = $stmt->fetchAll();

} catch (Throwable $e) {
    http_response_code(500);
    die("Error: " . $e->getMessage());
}
?>

<!doctype html>
<html lang="es">
<meta charset="utf-8">
<title>Listado ud4_musica</title>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/picocss@1/css/pico.min.css"/>
<body>
<h1>Listado de canciones (ud4_musica)</h1>
<table border="1" cellpadding="6">
    <tr>        You, 2 weeks ago • Entrega 5 actualizada y video subido ...
    <th>ID</th><th>Title</th><th>Artist</th><th>Genre</th>
    <th>Year</th><th>Duration</th><th>Price</th><th>Stock</th>
</tr>
<?php foreach ($rows as $r): ?>
    <tr>
        <td><?= htmlspecialchars(string: $r['id']) ?></td>
        <td><?= htmlspecialchars(string: $r['title']) ?></td>
        <td><?= htmlspecialchars(string: $r['artist']) ?></td>
        <td><?= htmlspecialchars(string: $r['genre'] ?? '') ?></td>
        <td><?= htmlspecialchars(string: $r['release_year'] ?? '') ?></td>
        <td><?= htmlspecialchars(string: $r['duration'] ?? '') ?></td>
        <td><?= number_format(num: (float)$r['unit_price'], decimals: 2) ?></td>
        <td><?= htmlspecialchars(string: $r['quantity_in_stock']) ?></td>
    </tr>
<?php endforeach; ?>
</table>
</body>
</html>
```

Listado de canciones (ud4_musica)

ID	Title	Artist	Genre	Year	Duration	Price	Stock
5	Imagine	John Lennon	Rock	1971	00:03:07	0.89	25
4	Smells Like Teen Spirit	Nirvana	Grunge	1991	00:05:01	1.09	10
3	Billie Jean	Michael Jackson	Pop	1982	00:04:54	1.19	12
2	Shape of You	Ed Sheeran	Pop	2017	00:03:53	0.99	20
1	Bohemian Rhapsody	Queen	Rock	1975	00:05:55	1.29	15

P4_login

En este programa se implementa un sistema de inicio de sesión (login) aplicando MVC (Modelo–Vista–Controlador).

El objetivo principal es comprender la estructura modular del patrón y cómo gestionar la conexión a una base de datos mediante PDO

Se ha creado la estructura de carpetas propuesta para la práctica, que nos permite separar correctamente los componentes del proyecto:

A continuación se crearon los archivos necesarios para cada capa del modelo MVC:

Modelo (Persona.php) → Define la estructura del usuario y la lógica relacionada con la autenticación.

```

<?php
require_once __DIR__ . '/../../config/database.php';

6 references | 0 implementations
class Persona {
    2 references
    private $conn;

    /**
     * The function creates a new Database object and establishes a connection to the database.
     */
    6 references | 0 overrides
    public function __construct() {
        $db = new Database();
        $this->conn = $db->connect();
    }

    /**
     * The login function queries a database table to retrieve a record based on the provided name and
     * phone number.
     *
     * @param usuario The `usuario` parameter in the `login`
     * @param contrasena The parameter `contrasena` is a string that represents the password of the user trying to log in.
     * @return The `login` function is returning a single record from the `persona`
     */
    1 reference | 0 overrides
    public function login($usuario, $contrasena): mixed {
        try {
            $query = "SELECT * FROM ud4_persona WHERE usuario = :usuario AND contrasena = :contrasena";
            $stmt = $this->conn->prepare(query: $query);
            $stmt->bindParam(param: ':usuario', var: &$usuario );
            $stmt->bindParam(param: ':contrasena', var: &$contrasena );
            $stmt->execute();
            return $stmt->fetch(mode: PDO::FETCH_ASSOC); // Devuelve un registro si lo encuentra
        } catch (PDOException $e) {
            die("Error al realizar la consulta: " . $e->getMessage());
        }
    }
}
?>

```

Controlador (PersonaController.php) → Gestiona las peticiones del usuario, recibe los datos del formulario y comunica el modelo con la vista.

```

<?php
require_once __DIR__ . '/../models/Persona.php';

1 reference | 0 implementations
class PersonaController
{
    1 reference | 0 overrides
    public function gestionLogin(): void
    {
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            $usuario = $_POST['usuario'] ?? '';
            $contrasena = $_POST['contrasena'] ?? '';

            $personaModel = new Persona();

            $persona = $personaModel->login(usuario: $usuario, contrasena: $contrasena);

            if ($persona) {
                // Redirige al usuario si las credenciales son correctas
                //header('Location: welcome.php');
                header(header: 'Location: app/views/welcome.php');
                //echo "Autenticación exitosa";
                exit();
            } else {
                // En caso de error, muestra un mensaje
                $error = "Credenciales incorrectas.";
                require_once __DIR__ . '/../views/login.php';
            }
        } else {
            require_once __DIR__ . '/../views/login.php';
        }
    }
}
?>

```

Vistas (login.php, welcome.php) → Muestran la interfaz gráfica del sistema (formulario de login, pantalla de bienvenida, etc.).

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="../../assets/css/styles.css">
7
8      <title>Login</title>
9  </head>
10 <body>
11     <h1>Login</h1>
12     <?php if (isset($error)): ?>
13         <p style="color: red;"><?= $error ?></p>
14     <?php endif; ?>
15     <form method="POST" action="index.php">
16         <label for="usuario">Usuario:</label>
17         <input type="text" id="usuario" name="usuario" required>
18         <br>
19         <label for="contrasena">Contraseña:</label>
20         <input type="text" id="contrasena" name="contrasena" required>
21         <br>
22         <button type="submit">Iniciar Sesión</button>
23     </form>
24 </body>
25 </html>

```

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta name="description" content="Información sobre el ciclo formativo de informática">
7      <title>Ciclo Formativo de Informática</title>
8      <link rel="stylesheet" href="../../assets/css/styles.css">
9  </head>
10 <body>
11     <header>
12         <h1>Ciclo Formativo de Informática</h1>
13     </header>
14     <section class="introduction">
15         <h2>Bienvenidos</h2>
16         <p>El ciclo formativo de informática está diseñado para preparar a los estudiantes en el uso de las tecnologías de la información, desarrollo de software, y adm</p>
17         
18     </section>
19
20     <section class="content">
21         <h2>¿Qué aprenderás?</h2>
22         <ul>
23             <li>Programación en lenguajes como Java, Python y PHP.</li>
24             <li>Administración de bases de datos con MySQL y PostgreSQL.</li>
25             <li>Gestión de redes y sistemas operativos.</li>
26             <li>Desarrollo de aplicaciones web y móviles.</li>
27         </ul>
28     </section>
29
30     <footer>
31         <p>© 2024 Ciclo Formativo de Informática - Todos los derechos reservados.</p>
32     </footer>
33
34 </body>
35 </html>

```

Uno de los aspectos más importantes del proyecto es el archivo database.php, ubicado dentro de la carpeta config/.

En este archivo se define una clase Database que contiene los parámetros necesarios para la conexión (host, usuario, contraseña y nombre de la base de datos).

Mediante una función connect() y utilizando PDO, se establece la conexión con la base de datos de forma segura y reutilizable

```
maria > UD4AccesoADatos > Entregas > P4_Login > config > database.php > ...
<?php
1 reference | 0 implementations
class Database
{
    1 reference
    private $host = 'localhost';
    1 reference
    private $dbName = 'dwes';
    1 reference
    private $username = 'root';
    1 reference
    private $password = '';
    3 references
    private $conn;

    1 reference | 0 overrides
    public function connect(): PDO
    {
        try {
            $this->conn = new PDO("mysql:host=$this->host;dbname=$this->dbName;charset=utf8", $this->username, $this->password);
            $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            return $this->conn;
        } catch (PDOException $e) {
            die("Error de conexión: " . $e->getMessage());
        }
    }
}
?>
```

Opciones extra

	← T →		id	nombre	apellidos	telefono	usuario	contraseña
<input type="checkbox"/>		Editar		Copiar		Borrar	1 Juan	Pérez 123456789 juan 123
<input type="checkbox"/>		Editar		Copiar		Borrar	2 María	López 987654321 maria 456
<input type="checkbox"/>		Editar		Copiar		Borrar	3 Carlos	García 456123789 carlos 789
<input type="checkbox"/>		Editar		Copiar		Borrar	4 Ana	Martínez 321654987 ana 012
<input type="checkbox"/>		Editar		Copiar		Borrar	5 Luis	Sánchez 654987321 luis 345

← → ⌂ ⓘ localhost/DWES/Dwes_Maria/UD4AccesoADatos/Entrega5/

Login

Usuario:

Contraseña:

Ciclo Formativo de Informática

Bienvenidos

El ciclo formativo de informática está diseñado para preparar a los estudiantes en el uso de las tecnologías de la información, desarrollo de software, y administración de sistemas.



P5_CRUD_MVC

En este programa se ha desarrollado una aplicación CRUD (Create, Read, Update, Delete) para la gestión de coches, aplicando el patrón de diseño MVC (Modelo–Vista–Controlador).

El acceso a la base de datos se realiza mediante PDO, de la misma forma que en el programa anterior.

En esta ocasión, el Controller contiene las funciones necesarias para realizar todas las operaciones CRUD sobre la tabla coches.

Entre las funciones principales del controlador se encuentran:

`index()` → Muestra todos los registros de la tabla.

```
<?php
include 'controllers/CocheController.php';

$controller = new CocheController(pdo: $pdo);

$action = $_GET['action'] ?? 'index';

switch ($action) {
    case 'crear':
        $controller->crear();
        break;
    case 'editar':
        $controller->editar();
        break;
    case 'eliminar':
        $controller->eliminar();
        break;
    default:
        $controller->index();
        break;
}
?>
```

crear() → Inserta un nuevo coche.

editar() → Modifica los datos de un coche existente.

eliminar() → Elimina un coche de la base de datos.

Una vez accedamos al programa podremos ver los coches que tenemos en la BD así como añadir, updatear o eliminar objetos de la BD

	<input type="checkbox"/>	<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	id	marca	modelo	anio
	<input type="checkbox"/>	<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	1	Toyota	Corolla	2020
	<input type="checkbox"/>	<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	2	Honda	Civic	2018
	<input type="checkbox"/>	<input type="checkbox"/>	Editar	<input type="checkbox"/>	Copiar	<input type="checkbox"/>	Borrar	3	Ford	Focus	2019

Gestión de Coches

Marca:

Modelo:

Año:

[Añadir Coche](#)

Lista
Lista de Coches

ID	Marca	Modelo	Año	Acciones
1	Toyota	Corolla	2020	<input type="text" value="Toyota"/> <input type="text" value="Corolla"/> <input type="text" value="2020"/> Actualizar Eliminar
2	Honda	Civic	2018	<input type="text" value="Honda"/> <input type="text" value="Civic"/> <input type="text" value="2018"/> Actualizar Eliminar
3	Ford	Focus	2019	<input type="text" value="Ford"/> <input type="text" value="Focus"/> <input type="text" value="2019"/> Actualizar Eliminar

P6_CRUD_SQLITE

Este programa no usaremos la BD de Mysql por lo que podremos apagar el puerto 3306, una bd sqlite, no es mas que una bd dentro de nuestro propio proyecto

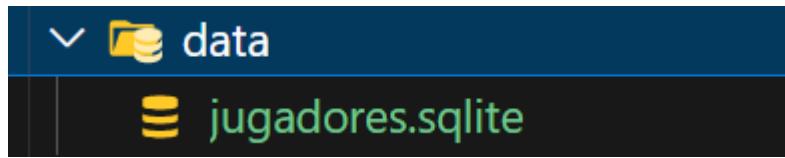
Para ello nos declaramos la ruta donde se nos creara el fichero para la BD, y usando nuevamente la clase PDO asignaremos los valores y tablas que tendra nuestra BD

```

1  <?php
2  declare(strict_types=1);
3
4  // Ruta al fichero SQLite
5  $dbDir = __DIR__ . '/data';
6  $dbFile = $dbDir . '/jugadores.sqlite';
7
8  if (!is_dir(filename: $dbDir)) {
9      mkdir(directory: $dbDir, permissions: 0777, recursive: true);
10 }
11
12 $pdo = new PDO(dsn: 'sqlite:' . $dbFile);
13 $pdo->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
14
15 // Crear tabla si no existe
16 $pdo->exec(statement: "
17 CREATE TABLE IF NOT EXISTS jugadores (
18     id INTEGER PRIMARY KEY AUTOINCREMENT,
19     usuario TEXT NOT NULL UNIQUE,
20     nombre TEXT NOT NULL,
21     apellidos TEXT NOT NULL,
22     created_at TEXT DEFAULT (datetime('now'))
23 );
24 ");
25
26 // Seed de 5 jugadores por defecto si la tabla está vacía
27 $count = (int)$pdo->query(query: "SELECT COUNT(*) FROM jugadores")->fetchColumn();
28 if ($count === 0) {
29     $pdo->beginTransaction();
30     $stmt = $pdo->prepare(query: "
31         INSERT INTO jugadores (usuario, nombre, apellidos)
32             VALUES (:usuario, :nombre, :apellidos)
33     ");
34
35     $seed = [
36         ['u' => 'jgarcia', 'n' => 'Juan', 'a' => 'García López'],
37         ['u' => 'mlopez', 'n' => 'María', 'a' => 'López Díaz'],
38         ['u' => 'aperez', 'n' => 'Alberto', 'a' => 'Pérez Martín'],
39         ['u' => 'cserrano', 'n' => 'Carmen', 'a' => 'Serrano Ruiz'],
40         ['u' => 'rnavarro', 'n' => 'Raúl', 'a' => 'Navarro Gil'],
41     ];
42
43     foreach ($seed as $s) {
44         $stmt->execute(params: [
45             ':usuario' => $s['u'],
46             ':nombre' => $s['n'],
47             ':apellidos' => $s['a'],
48         ]);
49     }
50     $pdo->commit();
51 }

```

Cuando ejecutamos este código llamado config.php se nos creará una nueva carpeta llamada data y dentro de ella podremos ver nuestra BD gracias a unas extensiones que hemos instalado previamente.



P7_files

Ejemplo1: escribir y leer un archivo

```
D:\wes_Maria\1-OB\1-AccesoADatos\Entregas\P7_files> Ejemplo1.php
1  <?php
2  $archivo = 'notas.txt';
3
4  // 1. Escribir
5  $texto = "Primera línea de texto\nSegunda línea\n";
6  $f = fopen(filename: $archivo, mode: 'w');
7  fwrite(stream: $f, data: $texto);
8  fclose(stream: $f);
9
10 // 2. Leer
11 $f = fopen(filename: $archivo, mode: 'r');
12 while (!feof(stream: $f)) {
13     echo fgets(stream: $f) . "<br>";
14 }
15 fclose(stream: $f);
16 ?>
```



localhost/DWES/

Primera línea de texto
Segunda línea

Dwes_Maria > UD4AccesoADatos > Entrega5 > P7_files > notas.txt

```
1 Primera línea de texto
2 Segunda línea
3
```

Ejemplo2: Lectura rápida

Dwes_Maria > UD4AccesoADatos > Entrega5 > P7_files > Ejemplo2.php

```
1 <?php
2 echo nl2br(string: file_get_contents(filename: 'notas.txt')); // lee todo el contenido
3 $lineas = file(filename: 'notas.txt'); // devuelve array con cada línea
4 print_r(value: $lineas);
5 ?>
```

Primera línea de texto
Segunda línea
Array ([0] => Primera línea de texto [1] => Segunda línea)

Ejemplo3: Añadir texto sin borrar archivo

```

1  <?php
2  $archivo = 'notas.txt';
3  $f = fopen(filename: $archivo, mode: 'a');
4  fwrite(stream: $f, data: "Nueva línea añadida.\n");
5  fclose(stream: $f);
6  ?>

```

```

1  Primera línea de texto
2  Segunda línea
3  Nueva línea añadida.
4

```

Ejemplo 4: Crear y descargar un CSV

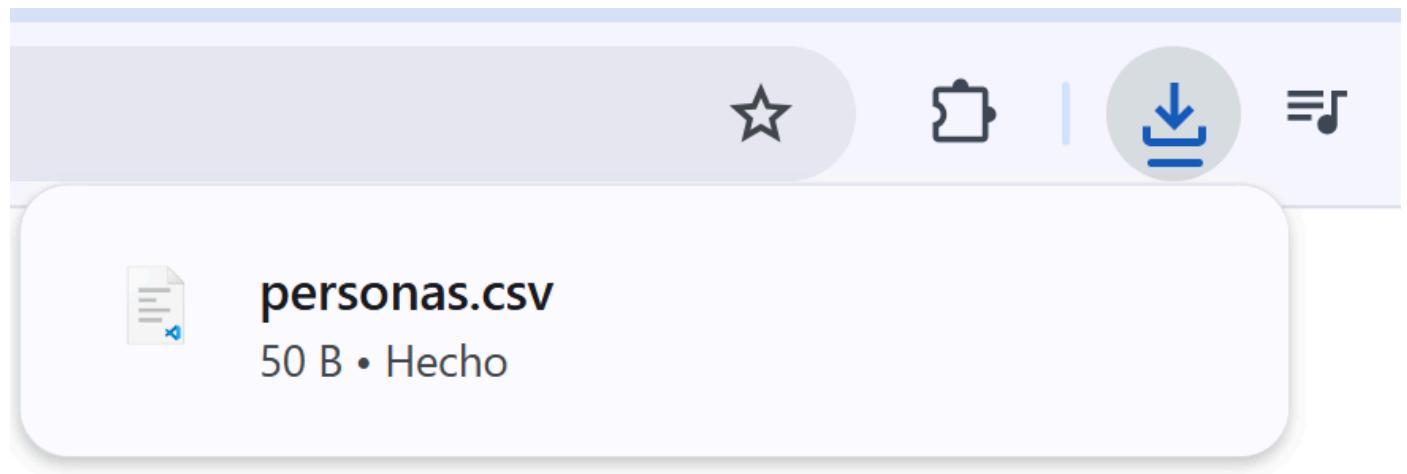
```

<?php
$datos = [
    [ 'Nombre', 'Edad', 'Ciudad'],
    [ 'Ana', 22, 'Sevilla'],
    [ 'Juan', 25, 'Málaga']
];

$fp = fopen(filename: 'personas.csv', mode: 'w');
foreach ($datos as $fila) {
    fputcsv(stream: $fp, fields: $fila, separator: ';');
}
fclose(stream: $fp);

header(header: 'Content-Type: text/csv');
header(header: 'Content-Disposition: attachment; filename="personas.csv"');
readfile(filename: 'personas.csv');
?>

```



Dwes_Maria > UD4AccesoADatos > Entrega5 > P7_files > personas.csv

	A	B	C	D	E	F
1	Nomb...	Edad	Ciudad			
2	Ana	22	Sevilla			
3	Juan	25	Málaga			
4						
5						

Ejemplo 5: Exportar a PDF con FPDF

```
1  <?php
2  require_once 'lib/fpdf.php';
3
4  $pdf = new FPDF();
5  $pdf->AddPage();
6  $pdf->SetFont('Arial', 'B', 16);
7  $pdf->Cell(0, 10, 'Ejemplo de exportacion PDF', 0, 1, 'C');
8
9  $pdf->SetFont('Arial', '', 12);
10 $pdf->Cell(0, 10, 'Generado con PHP + FPDF', 0, 1, 'L');
11
12 $pdf->Output('I', 'demo.pdf'); // I: Ver en navegador, D: Descargar
13 ...
```

Ejemplo de exportacion PDF

Generado con PHP + FPDF

Ejemplo 6: Exportar el contenido de un fichero de texto a PDF

```
Dwes_Maria > UD4AccesoADatos > Entrega5 > P7_files > Ejemplo6.php
1  <?php
2  require_once 'lib/fpdf.php';
3
4  $contenido = file(filename: 'notas.txt');
5  $pdf = new FPDF();
6  $pdf->AddPage();
7  $pdf->SetFont('Arial', '', 12);
8
9  foreach ($contenido as $linea) {
10     $pdf->Cell(0, 10, utf8_decode(string: trim(string: $linea)), 0, 1);
11 }
12
13 $pdf->Output('I', 'notas.pdf');
14 ?>
```

Primera línea de texto

Segunda línea

Nueva línea añadida.