

«Del Caos Monolítico al MVC»

Objetivo:

Transformar una aplicación PHP CRUD (clientes) que está todo en un único archivo, en una aplicación con separación progresiva de responsabilidades siguiendo el patrón MVC. El ejercicio se realiza en 6 etapas:

Paso 0 (monolítico) → **Paso 1**

Paso 1 (separa conexión + header/footer) → **Paso 2**

Paso 2 (modelo) → **Paso 3**

Paso 3 (controlador + vistas) → **Paso 4**

Paso 4 (router / front controller) → **Paso 5**

Paso 5 (layout + helpers + seguridad básica).

Entregables por alumno:

Una carpeta por paso: paso0/, paso1/, ... paso5/.

Cada carpeta debe estar funcional y contener un README.md corto que describa qué se cambió respecto al paso anterior.

SQL: Crear la base de datos y 20 registros (ejecutar primero)

Ejecuta esto en tu servidor MySQL (phpMyAdmin o consola):

```
CREATE DATABASE IF NOT EXISTS dwes_mvc_ej CHARACTER SET utf8mb4 COLLATE  
utf8mb4_general_ci;  
USE dwes_mvc_ej;
```

```
CREATE TABLE IF NOT EXISTS clientes (  
  id INT AUTO_INCREMENT PRIMARY KEY,
```

```
nombre VARCHAR(100) NOT NULL,  
email VARCHAR(150) NOT NULL,  
telefono VARCHAR(30) DEFAULT NULL,  
direccion VARCHAR(255) DEFAULT NULL,  
creado_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO clientes (nombre, email, telefono, direccion) VALUES  
('Ana Martín','ana.martin@example.com','600111222','C/ Mayor 1, Madrid'),  
('Luis Gómez','luis.gomez@example.com','600111223','C/ Luna 2, Sevilla'),  
('María López','maria.lopez@example.com','600111224','Av. Sol 3, Valencia'),  
('Carlos Ruiz','carlos.ruiz@example.com','600111225','Pza. Real 4, Bilbao'),  
('Sara Díaz','sara.diaz@example.com','600111226','C/ Olivo 5, Málaga'),  
('Pablo Ortega','pablo.ortega@example.com','600111227','C/ Río 6, Zaragoza'),  
('Elena Vega','elena.vega@example.com','600111228','C/ Mar 7, Murcia'),  
('Jorge Molina','jorge.molina@example.com','600111229','C/ Norte 8, Vigo'),  
('Carmen Santos','carmen.santos@example.com','600111230','C/ Sur 9, A Coruña'),  
('Raúl Fernández','raul.fernandez@example.com','600111231','C/ Este 10, Salamanca'),  
('Lucía Ramos','lucia.ramos@example.com','600111232','Pl. España 11, Cádiz'),  
('Diego Castro','diego.castro@example.com','600111233','C/ Jardín 12, León'),  
('Nuria Campos','nuria.campos@example.com','600111234','Pza. Nueva 13, Almería'),  
('Óscar Prieto','oscar.prieto@example.com','600111235','C/ Violeta 14, Huelva'),  
('Marta Blanco','marta.blanco@example.com','600111236','Av. Libertad 15, Burgos'),  
('Iván Herrera','ivan.herrera@example.com','600111237','C/ Roble 16, Logroño'),  
('Patricia Cruz','patricia.cruz@example.com','600111238','C/ Encina 17, Teruel'),  
('Andrés Gil','andres.gil@example.com','600111239','C/ Pino 18, Cuenca'),  
('Bea Navarro','bea.navarro@example.com','600111240','C/ Lirio 19, Segovia'),  
('Sergio León','sergio.leon@example.com','600111241','C/ Olmo 20, Huesca');
```

INSTRUCCIONES PREVIAS

Crea una carpeta `paso0/` en tu servidor web (por ejemplo `www/paso0` o `htdocs/paso0`).

- Asegúrate de que MySQL tiene la base `dwes_mvc_ej` creada con las tablas anteriores.
- Copia el archivo `index.php` que te doy abajo dentro de `paso0/` y abre `http://localhost/paso0/index.php` (ruta según tu setup).

PASO 0 — Código monolítico (TODO en uno)

Estructura (obligatoria para este paso):

```
paso0/  
  index.php      <-- TODO aquí: conexión, lógica CRUD y VISTA mezcladas  
  assets/ (opcional para css/imagenes)
```

Objetivo de este paso:

Mostrar un ejemplo real de "código espagueti": todo en un fichero. No hay `header.php` ni `footer.php`. Esto permitirá ver claramente por qué necesitamos separar responsabilidades.

`paso0/index.php`

```
<?php  
// paso0/index.php  
// PASO 0 – Código monolítico: conexión, lógica y vistas en un solo fichero.  
// NO incluir header/footer en este paso (se hará a partir del paso1).  
  
// ----- CONFIGURACIÓN DE LA BD (ajusta si tu usuario/clave difieren) -----  
----  
$host = '127.0.0.1';  
$db    = 'dwes_mvc_ej';  
$user  = 'root';  
$pass  = '';  
$dsn   = "mysql:host=$host;dbname=$db;charset=utf8mb4";  
  
try {  
    $pdo = new PDO($dsn, $user, $pass, [  
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION  
    ]);  
} catch (PDOException $e) {  
    die("Error de conexión a la base de datos: " . $e->getMessage());  
}  
  
// ----- LÓGICA (acción = list | add | edit | delete) -----  
$action = $_REQUEST['action'] ?? 'list';  
  
// ADD: insertar nuevo cliente  
if ($action === 'add' && $_SERVER['REQUEST_METHOD'] === 'POST') {  
    $nombre    = $_POST['nombre'] ?? '';  
    $email     = $_POST['email'] ?? '';  
    $telefono  = $_POST['telefono'] ?? '';  
    $direccion = $_POST['direccion'] ?? '';  
  
    $stmt = $pdo->prepare("INSERT INTO clientes (nombre,email,telefono,direccion)  
VALUES (?, ?, ?, ?)");  
    $stmt->execute([$nombre, $email, $telefono, $direccion]);  
}
```

```

    // PRG (Post-Redirect-Get) sería mejor, pero para simplicidad actualizamos
$action
    $action = 'list';
}

// EDIT: actualizar cliente
if ($action === 'edit' && $_SERVER['REQUEST_METHOD'] === 'POST') {
    $id      = (int)($_POST['id'] ?? 0);
    $nombre  = $_POST['nombre'] ?? '';
    $email   = $_POST['email'] ?? '';
    $telefono = $_POST['telefono'] ?? '';
    $direccion = $_POST['direccion'] ?? '';

    $stmt = $pdo->prepare("UPDATE clientes SET nombre=?, email=?, telefono=?,
direccion=? WHERE id=?");
    $stmt->execute([$nombre, $email, $telefono, $direccion, $id]);

    $action = 'list';
}

// DELETE: borrar cliente (por GET en este ejemplo; en producción usar POST)
if ($action === 'delete') {
    $id = (int)($_GET['id'] ?? 0);
    if ($id) {
        $stmt = $pdo->prepare("DELETE FROM clientes WHERE id=?");
        $stmt->execute([$id]);
    }
    $action = 'list';
}

// Obtener datos para editar (cuando accedemos a ?action=edit&id=...)
$cliente = null;
if ($action === 'edit' && isset($_GET['id']) && $_SERVER['REQUEST_METHOD'] ===
'GET') {
    $id = (int)$_GET['id'];
    $stmt = $pdo->prepare("SELECT * FROM clientes WHERE id=?");
    $stmt->execute([$id]);
    $cliente = $stmt->fetch(PDO::FETCH_ASSOC);
}

// ----- VISTA (HTML mezclado con PHP) -----
?><!doctype html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>Paso 0 – Código Monolítico (CRUD Clientes)</title>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <style>
        body{font-family:Arial,Helvetica,sans-serif;max-width:1000px;margin:18px
auto;padding:10px}
        header{background:#f6f8fa;padding:10px;border-radius:6px}
        h1{margin:0}
        .note{color:#666;font-size:0.95em}
        table{width:100%;border-collapse:collapse;margin-top:12px}
        th,td{border:1px solid #ddd;padding:8px;text-align:left}
        form{margin-top:12px;padding:10px;border:1px solid #eee;background:#fafafa}
        .actions a{margin-right:6px}
        .small{font-size:0.9em;color:#666}
        a.button{display:inline-block;padding:6px 10px;border-radius:4px;border:1px
solid #bbb;text-decoration:none;background:#fff}
    </style>

```

```

</head>
<body>

<header>
    <h1>Paso 0 - Código Monolítico (CRUD Clientes)</h1>
    <p class="note">Todo en un único fichero: conexión, lógica y vista. No hay
header/footer separados en este paso.</p>
</header>

<?php if ($action === 'list'): ?>

    <p><a class="button" href="?action=add">Añadir cliente</a></p>

    <?php
        // obtener todos los clientes
        $stmt = $pdo->query("SELECT * FROM clientes ORDER BY id DESC");
        $clientes = $stmt->fetchAll(PDO::FETCH_ASSOC);
    ?>

    <table>
        <thead>
            <tr>

<th>ID</th><th>Nombre</th><th>Email</th><th>Teléfono</th><th>Dirección</th><th>Cr
eado</th><th>Acciones</th>
            </tr>
        </thead>
        <tbody>
            <?php if (empty($clientes)): ?>
                <tr><td colspan="7">No hay clientes todavía.</td></tr>
            <?php else: ?>
                <?php foreach ($clientes as $c): ?>
                    <tr>
                        <td><?= htmlspecialchars($c['id']) ?></td>
                        <td><?= htmlspecialchars($c['nombre']) ?></td>
                        <td><?= htmlspecialchars($c['email']) ?></td>
                        <td><?= htmlspecialchars($c['telefono']) ?></td>
                        <td><?= htmlspecialchars($c['direccion']) ?></td>
                        <td><?= htmlspecialchars($c['creado_at']) ?></td>
                        <td class="actions">
                            <a href="?action=edit&id=<?= $c['id'] ?>">Editar</a>
                            |
                            <a href="?action=delete&id=<?= $c['id'] ?>" onclick="return
confirm('¿Borrar cliente #<?= $c['id'] ?>?')">Borrar</a>
                        </td>
                    </tr>
                <?php endforeach; ?>
            <?php endif; ?>
        </tbody>
    </table>

<?php elseif ($action === 'add'): ?>

    <h2>Añadir cliente</h2>

    <form method="post" action="?action=add">
        <label>Nombre<br><input name="nombre" required></label><br><br>
        <label>Email<br><input name="email" type="email" required></label><br><br>
        <label>Teléfono<br><input name="telefono"></label><br><br>
        <label>Dirección<br><input name="direccion"></label><br><br>
        <button type="submit">Crear</button>
    </form>

```

```

    <a class="button" href="?action=list">Volver</a>
</form>

<?php elseif ($action === 'edit' && $cliente): ?>

    <h2>Editar cliente #<?= htmlspecialchars($cliente['id']) ?></h2>

    <form method="post" action="?action=edit">
        <input type="hidden" name="id" value="<?= htmlspecialchars($cliente['id'])
?>">
        <label>Nombre<br><input name="nombre" required value="<?=
htmlspecialchars($cliente['nombre']) ?>"></label><br><br>
        <label>Email<br><input name="email" type="email" required value="<?=
htmlspecialchars($cliente['email']) ?>"></label><br><br>
        <label>Teléfono<br><input name="telefono" value="<?=
htmlspecialchars($cliente['telefono']) ?>"></label><br><br>
        <label>Dirección<br><input name="direccion" value="<?=
htmlspecialchars($cliente['direccion']) ?>"></label><br><br>
        <button type="submit">Guardar</button>
        <a class="button" href="?action=list">Volver</a>
    </form>

<?php else: ?>

    <p>Acción no reconocida. <a href="?action=list">Volver a la lista</a></p>

<?php endif; ?>

<hr>
<p class="small">Estructura de este paso:<br><code>paso0/ index.php (todo en un
solo fichero)</code></p>

</body>
</html>

```

EXPLICACIÓN (qué hay y por qué) — Paso 0

- **Conexión:** directamente en el archivo (`new PDO(...)`), con `try/catch`.
- **Lógica:** decisiones `if ($action === '...')` para `add`, `edit`, `delete`, `list`.
- **Vistas:** HTML mezclado con PHP al final del fichero.
- **Motivo didáctico:** mostrar lo que queremos evitar: mezcla de responsabilidades. En pasos posteriores veremos cómo esto pierde escalabilidad y dificulta trabajo en equipo.

PRUEBAS rápidas que debe hacer el alumno

1. Abrir `http://localhost/paso0/index.php` (ajusta la URL a tu servidor).
 2. Ver la lista de clientes (20 insertados por la SQL).
 3. Crear un cliente nuevo (formulario Add).
 4. Editar un cliente.
 5. Borrar un cliente.
 6. Revisar el código y localizar dónde está la conexión, dónde la lógica y dónde la vista.
-

README sugerido para `paso0/README.md` (pega en la carpeta)

Paso 0 – Código monolítico

Descripción:

Todo en un único fichero ``index.php``. Conexión, lógica y vistas mezcladas.

Cómo ejecutar:

1. Importa la base de datos ``dwes_mvc_ej`` (archivo SQL entregado).
2. Coloca este folder en el servidor (ej. `htdocs/paso0`).
3. Visita `http://localhost/paso0/index.php`

Cambios realizados:

- ...