

PASO 3 — Controlador + Vistas (MVC básico)

Objetivo:

Separar el *controlador* (encargado de procesar la petición y coordinar modelo + vista) y las *vistas* (solo presentación). El controlador usará ClienteModel para datos y cargará vistas que no contienen lógica SQL.

Estructura final del paso3 (debe aparecer exactamente en el footer):

```
paso3/  
  db.php  
  models/  
    ClienteModel.php  
  controllers/  
    ClienteController.php  
  views/  
    header.php  
    footer.php  
  clientes/  
    list.php  
    form.php  
  index.php  
README.md
```

1) paso3/db.php

(igual que en pasos anteriores — centraliza la conexión)

```
<?php  
// paso3/db.php  
$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,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
    ]);
} catch (PDOException $e) {
    die("Error de conexión a la base de datos: " . $e->getMessage());
}

```

2) paso3/models/ClienteModel.php

(Mismo modelo que en paso2 — lo incluyo de nuevo por completitud)

```

<?php

// paso3/models/ClienteModel.php

class ClienteModel
{
    private $pdo;

    public function __construct(PDO $pdo)
    {
        $this->pdo = $pdo;
    }

    public function getAll(): array
    {
        $stmt = $this->pdo->query("SELECT * FROM clientes ORDER BY id DESC");
        return $stmt->fetchAll();
    }

    public function getById(int $id): ?array

```

```

{

$stmt = $this->pdo->prepare("SELECT * FROM clientes WHERE id = ?");

$stmt->execute([\${id}]);

$r = $stmt->fetch();

return \$r ?: null;

}

public function create(array \$data): bool

{

$stmt = $this->pdo->prepare("INSERT INTO clientes (nombre,email,telefono,direccion) VALUES (?,?,?,?,?)");

return \$stmt->execute([

\$data['nombre'] ?? '',

\$data['email'] ?? '',

\$data['telefono'] ?? null,

\$data['direccion'] ?? null

]);


}

public function update(int \$id, array \$data): bool

{

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

return \$stmt->execute([

\$data['nombre'] ?? '',

\$data['email'] ?? '',

\$data['telefono'] ?? null,

\$data['direccion'] ?? null,

\$id

]);


}

```

```

public function delete(int $id): bool
{
    $stmt = $this->pdo->prepare("DELETE FROM clientes WHERE id = ?");

    return $stmt->execute([$id]);
}

}

```

3) paso3/controllers/ClienteController.php

Crea la carpeta `paso3/controllers/` y guarda este archivo. El controlador prepara variables (como `$clientes` o `$cliente`) y luego incluye la vista correspondiente. Las vistas incluyen el `header.php` y `footer.php` para mostrar título y estructura.

```

<?php

// paso3/controllers/ClienteController.php

require_once __DIR__ . '/../models/ClienteModel.php';

class ClienteController
{
    private $model;
    private $basePath;

    public function __construct(PDO $pdo)
    {
        $this->model = new ClienteModel($pdo);
        // ruta base del paso (útil para links relativos)
        $this->basePath = dirname(__DIR__);
    }

    /**
     * Mostrar listado */
    public function index()
    {
        $clientes = $this->model->getAll();

```

```

// la vista se encargará de incluir header/footer
include $this->basePath . '/views/clientes/list.php';

}

/** Mostrar formulario y procesar creación */
public function add()
{
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $data = [
            'nombre' => trim($_POST['nombre'] ?? ''),
            'email' => trim($_POST['email'] ?? ''),
            'telefono' => trim($_POST['telefono'] ?? ''),
            'direccion' => trim($_POST['direccion'] ?? '')
        ];
        $this->model->create($data);
        header('Location: index.php');
        exit;
    }

    $action = 'add';
    $cliente = null;
    include $this->basePath . '/views/clientes/form.php';
}

/** Mostrar formulario y procesar edición */
public function edit()
{
    $id = (int)($_GET['id'] ?? 0);

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $id = (int)($_POST['id'] ?? $id);

```

```

$data = [
    'nombre' => trim($_POST['nombre'] ?? ""),
    'email' => trim($_POST['email'] ?? ""),
    'telefono' => trim($_POST['telefono'] ?? ""),
    'direccion' => trim($_POST['direccion'] ?? "")
];

$this->model->update($id, $data);
header('Location: index.php');
exit;
}

$cliente = $this->model->getById($id);
$action = 'edit';
include $this->basePath . '/views/clientes/form.php';
}

/** Borrar */
public function delete()
{
    $id = (int)($_GET['id'] ?? 0);
    if ($id) {
        $this->model->delete($id);
    }
    header('Location: index.php');
    exit;
}

```

4) paso3/views/header.php

Guarda en `paso3/views/header.php`. Debe mostrar el **título del paso** (requisito del profesor) y abrir el layout.

```
<?php
```

```

// paso3/views/header.php

?>

<!doctype html>
<html lang="es">
<head>
<meta charset="utf-8">
<title>Paso 3 — Controlador + Vistas</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:#fff9e6;padding:10px;border-radius:6px}
h1{margin:0}
.small{color:#666;font-size:0.95em}
.btn{display:inline-block;padding:6px 10px;margin-right:6px;border:1px solid #bbb;border-radius:4px;text-decoration:none;background:#fff}
</style>
</head>
<body>
<header>
<h1>Paso 3 — Controlador + Vistas (MVC básico)</h1>
<p class="small">Las vistas son ahora puras y el controlador orquesta modelo + vistas.</p>
</header>
<main>

```

5) paso3/views/footer.php

Guarda en paso3/views/footer.php. Debe mostrar la **estructura de ficheros** del paso (exacta).

```

<?php
// paso3/views/footer.php
?>
</main>
<hr>
```

```

<footer>
  <p class="small">Estructura de este paso:</p>
  <pre>
paso3/
  db.php
  models/
    ClienteModel.php
  controllers/
    ClienteController.php
  views/
    header.php
    footer.php
  clientes/
    list.php
    form.php
  index.php
  README.md
</pre>
  <p class="small">Siguiente: Paso 4 → Front controller / Router.</p>
</footer>
</body>
</html>

```

6) paso3/views/clientes/list.php (vista pura)

Guarda en paso3/views/clientes/list.php. **La vista incluye header y footer** para cumplir el requisito de mostrar título y estructura en pantalla.

```

<?php
// paso3/views/clientes/list.php
// Esta vista espera la variable $clientes (proporcionada por el controlador).
require __DIR__ . '/../header.php';
?>

```

```

<p><a class="btn" href="index.php?action=add">Añadir cliente</a></p>

<table border="1" cellpadding="6" cellspacing="0" style="width:100%;border-collapse:collapse">
  <thead>

    <tr><th>ID</th><th>Nombre</th><th>Email</th><th>Teléfono</th><th>Dirección</th><th>Creado</th><th>Acciones</th></tr>

  </thead>
  <tbody>
    <?php if (empty($clientes)): ?>
    <tr><td colspan="7">No hay clientes.</td></tr>
    <?php else: ?>
    <?php foreach ($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>
          <a href="index.php?action=edit&id=<?= $c['id'] ?>">Editar</a> |
          <a href="index.php?action=delete&id=<?= $c['id'] ?>" onclick="return confirm('¿Borrar cliente #<?= $c['id'] ?>?')">Borrar</a>
        </td>
      </tr>
    <?php endforeach; ?>
    <?php endif; ?>
  </tbody>
</table>

```

```
<?php  
require __DIR__ . '/../footer.php';
```

7) paso3/views/clientes/form.php (vista pura)

Guarda en paso3/views/clientes/form.php. Espera \$action y \$cliente (cuando edita). Incluye header/footer.

```
<?php  
// paso3/views/clientes/form.php  
require __DIR__ . '/../header.php';  
  
$editing = (isset($action) && $action === 'edit' && !empty($cliente));  
?  
<h2><?= $editing ? 'Editar cliente #' . htmlspecialchars($cliente['id']) : 'Añadir cliente' ?></h2>  
  
<form method="post" action="index.php?action=<?= $editing ? 'edit' : 'add' ?>" style="max-width:600px">  
<?php if ($editing): ?>  
  <input type="hidden" name="id" value="<?= htmlspecialchars($cliente['id']) ?>">  
<?php endif; ?>  
  
  <label>Nombre<br><input name="nombre" required style="width:100%" value="<?= $editing ?  
    htmlspecialchars($cliente['nombre']) : " ?>"></label><br><br>  
  <label>Email<br><input type="email" name="email" required style="width:100%" value="<?= $editing ?  
    htmlspecialchars($cliente['email']) : " ?>"></label><br><br>  
  <label>Teléfono<br><input name="telefono" style="width:100%" value="<?= $editing ?  
    htmlspecialchars($cliente['telefono']) : " ?>"></label><br><br>  
  <label>Dirección<br><input name="direccion" style="width:100%" value="<?= $editing ?  
    htmlspecialchars($cliente['direccion']) : " ?>"></label><br><br>  
  
  <button type="submit"><?= $editing ? 'Guardar' : 'Crear' ?></button>  
  <a class="btn" href="index.php">Cancelar</a>  
</form>  
  
<?php
```

```
require __DIR__ . '/../footer.php';
```

8) paso3/index.php (front simple que delega en el controlador)

Guarda en paso3/index.php. Este archivo instancia el controlador y llama al método adecuado según action. No hace consultas SQL — todo lo hace el controlador + modelo.

```
<?php

// paso3/index.php

require __DIR__ . '/db.php';
require __DIR__ . '/controllers/ClienteController.php';

// crear controlador
$ctrl = new ClienteController($pdo);

// decidir acción (index/list | add | edit | delete)
$action = $_GET['action'] ?? 'index';

// mapear acciones a métodos del controlador
switch ($action) {
    case 'add':
        $ctrl->add();
        break;
    case 'edit':
        $ctrl->edit();
        break;
    case 'delete':
        $ctrl->delete();
        break;
    case 'index':
    case 'list':
    default:
        $ctrl->index();
```

```
break;  
}
```

9) paso3/README.md

Guarda en paso3/README.md y pídele al alumno que lo complete con 1–3 frases de reflexión.

Paso 3 — Controlador + Vistas (MVC básico)

Qué se ha hecho

Estructura

Tareas

Notas didácticas / por qué este paso importa

- **Control centralizado:** el controlador maneja la lógica de rutas y decide qué vista cargar.
- **Vistas puras:** las vistas ya no contienen consultas SQL, solo muestran lo que les pasa el controlador.
- **Más escalable:** añadir nuevos controladores (p. ej. ProductoController) ahora es directo.
- **Mejor para el trabajo en equipo:** frontend puede trabajar en vistas sin tocar lógica.