

PASO 2 — Separar el Modelo (ClienteModel)

Objetivo:

Extraer las consultas a la base de datos en una clase **ClienteModel** (single responsibility). index.php deja de ejecutar SQL directo y usa el modelo. Esto hace más fácil testear y reutilizar la lógica de datos.

Estructura final del paso2 (se mostrará en el footer):

paso2/

db.php

models/

ClienteModel.php

index.php

views/

header.php

footer.php

clientes/

list.php

form.php

README.md

1) paso2/db.php

(Mismo que en paso1 — lo incluyo por completitud)

```
<?php
```

```
// paso2/db.php
```

```
// Conexión PDO centralizada. Ajusta usuario/clave según tu entorno.
```

```
$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) paso2/models/ClienteModel.php

Crea la carpeta paso2/models/ y guarda este archivo. El modelo expone métodos CRUD: getAll, getById, create, update, delete.

```
<?php
// paso2/models/ClienteModel.php

class ClienteModel
{
    private $pdo;

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

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

```
/** Obtener un cliente por id */
```

```
public function getById(int $id): ?array
```

```
{
```

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

```
    $stmt->execute([$id]);
```

```
    $r = $stmt->fetch();
```

```
    return $r ?: null;
```

```
}
```

```
/** Crear cliente. $data debe contener 'nombre','email','telefono','direccion' */
```

```
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
```

```
    ]);
```

```
}
```

```
/** Actualizar cliente por id */
```

```
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
    ));
}

/** Borrar cliente por id */
public function delete(int $id): bool
{
    $stmt = $this->pdo->prepare("DELETE FROM clientes WHERE id = ?");
    return $stmt->execute([$id]);
}
}

```

3) paso2/views/header.php

```

<?php
// paso2/views/header.php
?>

<!doctype html>

<html lang="es">

<head>

    <meta charset="utf-8">

    <title>Paso 2 — Modelo</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:#eefbf0;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 2 — Crear Modelo (ClienteModel)</h1>

  <p class="small">En este paso la lógica de acceso a datos se ha movido al modelo
<code>ClienteModel</code>.</p>

</header>

<main>
```

4) paso2/views/footer.php

```
<?php

// paso2/views/footer.php

?>

</main>

<hr>

<footer>

  <p class="small">Estructura de este paso:</p>

  <pre>

paso2/

db.php

models/

  ClienteModel.php

index.php

views/

  header.php

  footer.php

  clientes/

    list.php

    form.php

README.md

  </pre>
```

<p class="small">Siguiente: Paso 3 → separamos controladores y vistas.</p>

</footer>

</body>

</html>

5) paso2/views/clientes/list.php

<?php

// paso2/views/clientes/list.php

// Espera variable \$clientes en scope

?>

<h2>Listado de clientes</h2>

<p>Añadir cliente</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>

```

6) paso2/views/clientes/form.php

```

<?php
// paso2/views/clientes/form.php
// Variables esperadas en scope: $action, $cliente (opcional)
$editing = ($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>
```

7) paso2/index.php — comportamiento usando el modelo

```
<?php
// paso2/index.php
// Paso 2 — Usamos ClienteModel para todas las operaciones.

// incluir conexión y modelo
require __DIR__ . '/db.php';
require __DIR__ . '/models/ClienteModel.php';

// instanciar modelo
$model = new ClienteModel($pdo);

// acción: list | add | edit | delete
$action = $_REQUEST['action'] ?? 'list';

// --- Crear (ADD)
if ($action === 'add' && $_SERVER['REQUEST_METHOD'] === 'POST') {
    $data = [
        'nombre' => trim($_POST['nombre'] ?? ''),
        'email' => trim($_POST['email'] ?? ''),
        'telefono' => trim($_POST['telefono'] ?? ''),
        'direccion' => trim($_POST['direccion'] ?? '')
    ];
    $model->create($data);
    // PRG: redirigir a la lista
    header('Location: index.php');
    exit;
```



```
}
```

```
// --- Editar (UPDATE)
```

```
if ($action === 'edit' && $_SERVER['REQUEST_METHOD'] === 'POST') {
```

```
    $id = (int)($_POST['id'] ?? 0);
```

```
    $data = [
```

```
        'nombre' => trim($_POST['nombre'] ?? ''),
```

```
        'email' => trim($_POST['email'] ?? ''),
```

```
        'telefono' => trim($_POST['telefono'] ?? ''),
```

```
        'direccion' => trim($_POST['direccion'] ?? '')
```

```
    ];
```

```
    $model->update($id, $data);
```

```
    header('Location: index.php');
```

```
    exit;
```

```
}
```

```
// --- Borrar (DELETE)
```

```
if ($action === 'delete') {
```

```
    $id = (int)($_GET['id'] ?? 0);
```

```
    if ($id) $model->delete($id);
```

```
    header('Location: index.php');
```

```
    exit;
```

```
}
```

```
// --- Obtener cliente para editar (GET)
```

```
$cliente = null;
```

```
if ($action === 'edit' && isset($_GET['id'])) {
```

```
    $cliente = $model->getById((int)$_GET['id']);
```

```
}
```

```
// --- Preparar datos para vistas
```

```

if ($action === 'list') {
    $clientes = $model->getAll();
}

// incluir header (título del paso)
require __DIR__ . '/views/header.php';

// incluir vista correspondiente
if ($action === 'list') {
    include __DIR__ . '/views/clientes/list.php';
} elseif ($action === 'add') {
    include __DIR__ . '/views/clientes/form.php';
} elseif ($action === 'edit') {
    include __DIR__ . '/views/clientes/form.php';
} else {
    echo "<p>Acción desconocida.</p>";
}

// incluir footer (estructura del paso)
require __DIR__ . '/views/footer.php';

```

8) paso2/README.md

Paso 2 — Separar el Modelo

Qué se ha hecho

Estructura

Notas didácticas / por qué este paso importa

- **Separación de responsabilidades:** ahora toda la lógica de acceso a la BD está centralizada en una clase. Si cambiamos la tabla o la consulta solo modificamos el modelo.
- **Reutilización:** cualquier controlador o script puede usar `new ClienteModel($pdo)`.

- **Testabilidad:** más fácil simular datos o crear pruebas unitarias sobre ClienteModel.
- **Legibilidad:** index.php queda más limpio y se centra en el flujo de la aplicación (control).