



PASO 4 — Front Controller / Router (separación public/ y app/)

Objetivo:

Crear un único punto de entrada (public/index.php) que despache peticiones mediante un router simple hacia controladores en app/controllers/. Mantener app/models/ClienteModel.php y app/views/ para vistas. Esto facilita seguridad (archivos sensibles fuera de public/) y routing limpio.

Estructura final (debe mostrarse exactamente en el footer):

```
paso4/  
  
  app/  
    bootstrap.php  
    controllers/  
      ClienteController.php  
    models/  
      ClienteModel.php  
    views/  
      header.php  
      footer.php  
      clientes/  
        list.php  
        form.php  
  public/  
    index.php  
    .htaccess  
  
  README.md
```

Nota: Si tu servidor no permite configurar public/ como document root, accede via <http://localhost/paso4/public/index.php> o usa .htaccess (incluido) para redireccionar todas las peticiones a index.php.

1) paso4/app/bootstrap.php

Crea paso4/app/bootstrap.php. Carga la configuración de BD, arranca \$pdo, registra un autoload mínimo y funciones utilitarias.

```
<?php
```

```

// paso4/app/bootstrap.php

// Inicialización global para la app (BD, autoload, helpers)

// --- Conexión PDO (ajusta usuario/clave si es necesario) ---
$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) {
    // En entorno didáctico mostramos el error
    die("Error de conexión a la base de datos: " . $e->getMessage());
}

// --- Autoload simple para controllers y models ---
spl_autoload_register(function($class) {
    $paths = [
        __DIR__ . '/controllers/' . $class . '.php',
        __DIR__ . '/models/' . $class . '.php',
    ];
    foreach ($paths as $file) {
        if (file_exists($file)) {
            require_once $file;
            return;
        }
    }
}

```

```

    }
});

// función helper para escapar HTML
if (!function_exists('esc')) {
    function esc($s) { return htmlspecialchars($s ?? "", ENT_QUOTES, 'UTF-8'); }
}

```

2) paso4/app/models/ClienteModel.php

Crea paso4/app/models/ClienteModel.php. Mismo modelo robusto con métodos CRUD.

```

<?php

// paso4/app/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) paso4/app/controllers/ClienteController.php

Crea paso4/app/controllers/ClienteController.php. Controller con métodos index, add, edit, delete. Usa rutas limpias: controller recibe parámetros desde router.

```
<?php
```

```
// paso4/app/controllers/ClienteController.php
```

```
class ClienteController
```

```
{
```

```
    private $model;
```

```
    private $viewsPath;
```

```
    public function __construct(PDO $pdo)
```

```
    {
```

```
        $this->model = new ClienteModel($pdo);
```

```
        $this->viewsPath = __DIR__ . '/../views';
```

```
    }
```

```
// lista
```

```
public function index()
```

```
{
```

```
    $clientes = $this->model->getAll();
```

```
    include $this->viewsPath . '/clientes/list.php';
```

```
}
```

```
// crear: mostrar form y procesar POST
```

```
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);

        // redirigir a la lista (ruta relativa a public/)
        header('Location: /clientes');
        exit;
    }

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

// editar: mostrar form y procesar POST
// $params puede incluir el id en $params[0]
public function edit(array $params = [])
{
    $id = isset($params[0]) ? (int)$params[0] : (int)($_REQUEST['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: /clientes');
exit;
}

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

```

```

// borrar: id en params[0]
public function delete(array $params = [])
{
    $id = isset($params[0]) ? (int)$params[0] : (int)($_REQUEST['id'] ?? 0);
    if ($id) $this->model->delete($id);
    header('Location: /clientes');
    exit;
}
}

```

4) paso4/app/views/header.php

Crea paso4/app/views/header.php. Muestra título del paso (requisito) y abre layout.

```

<?php
// paso4/app/views/header.php
?>

<!doctype html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>Paso 4 — Router / Front Controller</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:#e8f6ff;padding:10px;border-radius:6px}

  h1{margin:0}

  .nav{margin-top:8px}

  .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 4 — Router y Front Controller</h1>

  <p class="small">Entrada única en <code>public/index.php</code> que despacha peticiones a
controladores en <code>app/controllers/</code>.</p>

</header>
<main>

```

5) paso4/app/views/footer.php

Crea paso4/app/views/footer.php. Debe mostrar exactamente la estructura del paso.

```

<?php

// paso4/app/views/footer.php

?>

</main>

<hr>

<footer>

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

  <pre>
paso4/
app/
  bootstrap.php
  controllers/

```



```
    ClienteController.php
models/
    ClienteModel.php
views/
    header.php
    footer.php
    clientes/
        list.php
        form.php
public/
    index.php
    .htaccess
README.md
</pre>


>Accede via /clientes, /clientes/add,  

/clientes/edit/{id}, /clientes/delete/{id}</p>
</footer>
</body>
</html>


```

6) paso4/app/views/clientes/list.php

Crea paso4/app/views/clientes/list.php. Vista pura; incluye header/footer.

```
<?php
// paso4/app/views/clientes/list.php
require __DIR__ . '/../header.php';
?>

<p>
    <a class="btn" href="/clientes/add">Añadir cliente</a>
    <a class="btn" href="/clientes">Refrescar</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><?= esc($c['id']) ?></td>

                    <td><?= esc($c['nombre']) ?></td>

                    <td><?= esc($c['email']) ?></td>

                    <td><?= esc($c['telefono']) ?></td>

                    <td><?= esc($c['direccion']) ?></td>

                    <td><?= esc($c['creado_at']) ?></td>

                    <td>

                        <a href="/clientes/edit/<?= $c['id'] ?>">Editar</a> |

                        <a href="/clientes/delete/<?= $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) paso4/app/views/clientes/form.php

Crea paso4/app/views/clientes/form.php. Vista pura para add/edit. Usa rutas limpias y esc().

```
<?php
// paso4/app/views/clientes/form.php
require __DIR__ . '/../header.php';

$editing = (isset($action) && $action === 'edit' && !empty($cliente));
?>

<h2><?= $editing ? 'Editar cliente #' . esc($cliente['id']) : 'Añadir cliente' ?></h2>

<form method="post" action="">
    <?php if ($editing): ?>
        <input type="hidden" name="id" value="<?= esc($cliente['id']) ?>">
    <?php endif; ?>

    <label>Nombre<br><input name="nombre" required style="width:100%" value="<?= $editing ?
esc($cliente['nombre']) : " ?>"></label><br><br>

    <label>Email<br><input type="email" name="email" required style="width:100%" value="<?= $editing ?
esc($cliente['email']) : " ?>"></label><br><br>

    <label>Teléfono<br><input name="telefono" style="width:100%" value="<?= $editing ?
esc($cliente['telefono']) : " ?>"></label><br><br>

    <label>Dirección<br><input name="direccion" style="width:100%" value="<?= $editing ?
esc($cliente['direccion']) : " ?>"></label><br><br>

    <button type="submit"><?= $editing ? 'Guardar' : 'Crear' ?></button>

    <a class="btn" href="/clientes">Cancelar</a>
</form>

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

8) paso4/public/index.php (Front Controller / Router)

Crea paso4/public/index.php. Este es el único archivo público al que apuntará el servidor. Analiza la URL y despacha a ClienteController.

```
<?php

// paso4/public/index.php

// Front controller simple para Paso 4


// Ajusta el path al bootstrap de la app
require __DIR__ . '/../app/bootstrap.php';


// Parsear la ruta

// Se espera que el servidor sirva esta carpeta como base: / (ej: http://localhost/paso4/public/)

// Obtenemos la ruta solicitada: /clientes/edit/5 o /clientes
$url = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);

// Si la aplicación está en un subdirectorío (como /paso4/public), queremos eliminar esa parte.
// Obtenlo calculando el "base" relativo al SCRIPT_NAME.
$scriptName = str_replace('\\', '/', dirname($_SERVER['SCRIPT_NAME']));
$base = rtrim($scriptName, '/');

if ($base && strpos($url, $base) === 0) {
    $path = substr($url, strlen($base));
} else {
    $path = $url;
}

$path = trim($path, '/'); // e.g. "clientes/edit/5" or ""

// Default: clientes

if ($path === '') {
    $segments = [];
```

```

} else {

    $segments = explode('/', $path);
}

// Decide controlador: por ahora solo 'clientes'
$resource = $segments[0] ?? 'clientes';
$action = $segments[1] ?? 'index';
$params = array_slice($segments, 2);

// Mapear rutas simples
switch ($resource) {
    case 'clientes':
        // Aseguramos que la clase ClienteController esté disponible (autoload lo carga)
        $controller = new ClienteController($pdo);
        // mapear acciones a métodos; pasar params cuando sea necesario
        if ($action === 'add') {
            $controller->add();
        } elseif ($action === 'edit') {
            $controller->edit($params);
        } elseif ($action === 'delete') {
            $controller->delete($params);
        } else {
            // 'index' / 'list' / default
            $controller->index();
        }
        break;

    default:
        // 404 simple
        header("HTTP/1.0 404 Not Found");
        echo "<h1>404 - Recurso no encontrado</h1>";

```

```
    echo "<p>Recurso: " . esc($resource) . "</p>";  
    break;  
}
```

9) paso4/public/.htaccess (opcional — para tener URLs limpias)

Si usas Apache y puedes poner .htaccess en public/, este fichero redirige todo a index.php (útil si quieres omitir public/ en la URL o forzar routing).

```
# paso4/public/.htaccess
```

```
RewriteEngine On
```

```
# Si el recurso existe (archivo o directorio) dejar pasar
```

```
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
# Redirigir todo a index.php
```

```
RewriteRule ^ index.php [QSA,L]
```

Nota: el index.php actual ya maneja rutas sin necesidad del .htaccess, pero el .htaccess facilita que rutas como /clientes/add funcionen al apuntar DocumentRoot a public/.

10) paso4/README.md

Copia este README a paso4/README.md.

```
# Paso 4 — Front Controller / Router
```

```
## Qué se ha hecho
```

```
## Estructura
```



Notas didácticas importantes

- **Seguridad:** al mover la lógica y configuración fuera de public/, reduces riesgo de exponer archivos sensibles.
- **URLs limpias:** router facilita crear URLs amigables; opcionalmente se puede mejorar con .htaccess.
- **Extensibilidad:** añadir nuevos controladores (p. ej. ProductoController) es directo; solo hay que crear app/controllers/ProductoController.php y adaptar el router.

- **Recomendación práctica:** configurar DocumentRoot a paso4/public en Apache/Nginx para reproducir el comportamiento real de una app.