🏳 PASO 5 — Plantilla (layout), helpers y seguridad básica

Objetivo:
Añadir una capa de presentación (layout), helpers reutilizables (escapar HTML y tokens CSRF), validación básica de formularios y uso de output buffering para inyectar vistas en una plantilla (layout.php). Mantiene router/front controller y estructura public/ + app/.

Estructura final (debe aparecer exactamente en el footer):

```
paso5/
 app/
  bootstrap.php
  helpers.php
  controllers/
   ClienteController.php
  models/
   ClienteModel.php
  views/
   layout.php
   header.php
   footer.php
   clientes/
    list.php
    form.php
 public/
  index.php
  .htaccess
 README.md
```

---

1) paso5/app/bootstrap.php

```
<?php
// paso5/app/bootstrap.php
// Inicialización: sesiones, BD, autoload, helpers
```

```php
// iniciar sesión para CSRF y flashes
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

// Conexión PDO
$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());
}

// helpers y funciones comunes
require_once __DIR__ . '/helpers.php';

// Autoload simple
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;
        }
    }
});
```

---

2) paso5/app/helpers.php

```
<?php
// paso5/app/helpers.php

// Escapar HTML
function esc($s) {
    return htmlspecialchars($s ?? '', ENT_QUOTES, 'UTF-8');
}

// Mensajes flash simples (guardados en sesión)
function flash_set($key, $msg) {
    $_SESSION['flash'][$key] = $msg;
}
function flash_get($key) {
    $v = $_SESSION['flash'][$key] ?? null;
    if (isset($_SESSION['flash'][$key])) unset($_SESSION['flash'][$key]);
    return $v;
}

// CSRF: generar y comprobar token
function csrf_token() {
    if (empty($_SESSION['csrf_token'])) {
        $_SESSION['csrf_token'] = bin2hex(random_bytes(16));
```

```
    }
    return $_SESSION['csrf_token'];
}
function csrf_field_html() {
    $t = csrf_token();
    return '<input type="hidden" name="csrf_token" value="' . esc($t) . '">';
}
function csrf_check($token) {
    if (empty($_SESSION['csrf_token'])) return false;
    return hash_equals($_SESSION['csrf_token'], (string)$token);
}

// Render: renderizar una vista dentro de layout
function render_view($viewPath, $data = [], $title = '') {
    extract($data, EXTR_SKIP);
    // la vista debe producir $content
    ob_start();
    include $viewPath;
    $content = ob_get_clean();
    // incluir layout
    include __DIR__ . '/views/layout.php';
}
```

---

3) paso5/app/models/ClienteModel.php

(Mismo que antes; lo incluyo por completitud)

```
<?php
// paso5/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]);
    }
}
```

---

4) paso5/app/controllers/ClienteController.php

Controller con validación básica y uso de CSRF + flash messages. Usa render_view() para mostrar vistas dentro del layout.

```
<?php
// paso5/app/controllers/ClienteController.php

class ClienteController
{
    private $model;
    private $viewsPath;

    public function __construct(PDO $pdo) {
        $this->model = new ClienteModel($pdo);
        $this->viewsPath = __DIR__ . '/../views';
    }

    // Listado
    public function index() {
        $clientes = $this->model->getAll();
        render_view($this->viewsPath . '/clientes/list.php', ['clientes' => $clientes], 'Paso 5 — Lista de clientes');
```

```
}

// Añadir
public function add() {
    $errors = [];
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        // CSRF
        if (!csrf_check($_POST['csrf_token'] ?? '')) {
            $errors[] = 'Token CSRF inválido.';
        } else {
            $nombre = trim($_POST['nombre'] ?? '');
            $email  = trim($_POST['email'] ?? '');
            // Validaciones simples
            if ($nombre === '') $errors[] = 'El nombre es obligatorio.';
            if ($email === '' || !filter_var($email, FILTER_VALIDATE_EMAIL)) $errors[] = 'Email inválido.';
            if (empty($errors)) {
                $data = [
                    'nombre' => $nombre,
                    'email' => $email,
                    'telefono' => trim($_POST['telefono'] ?? ''),
                    'direccion' => trim($_POST['direccion'] ?? '')
                ];
                $this->model->create($data);
                flash_set('success', 'Cliente creado correctamente.');
                header('Location: /clientes');
                exit;
            }
        }
    }
    // Mostrar formulario (GET o POST con errores)
    $csrf = csrf_token();
```

```
    render_view($this->viewsPath . '/clientes/form.php', [
        'action' => 'add',
        'cliente' => null,
        'errors' => $errors,
        'csrf' => $csrf
    ], 'Paso 5 — Añadir cliente');
}

// Editar
public function edit(array $params = []) {
    $errors = [];
    $id = isset($params[0]) ? (int)$params[0] : (int)($_REQUEST['id'] ?? 0);
    $cliente = $this->model->getById($id);
    if (!$cliente) {
        flash_set('error', 'Cliente no encontrado.');
        header('Location: /clientes');
        exit;
    }

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        if (!csrf_check($_POST['csrf_token'] ?? '')) {
            $errors[] = 'Token CSRF inválido.';
        } else {
            $nombre = trim($_POST['nombre'] ?? '');
            $email  = trim($_POST['email'] ?? '');
            if ($nombre === '') $errors[] = 'El nombre es obligatorio.';
            if ($email === '' || !filter_var($email, FILTER_VALIDATE_EMAIL)) $errors[] = 'Email inválido.';
            if (empty($errors)) {
                $data = [
                    'nombre' => $nombre,
                    'email' => $email,
```

```
                'telefono' => trim($_POST['telefono'] ?? ''),
                'direccion' => trim($_POST['direccion'] ?? '')
            ];
            $this->model->update($id, $data);
            flash_set('success', 'Cliente actualizado.');
            header('Location: /clientes');
            exit;
        }
      }
    }

    $csrf = csrf_token();
    render_view($this->viewsPath . '/clientes/form.php', [
      'action' => 'edit',
      'cliente' => $cliente,
      'errors' => $errors,
      'csrf' => $csrf
    ], 'Paso 5 — Editar cliente');
  }

  // Borrar (en este ejemplo via GET; en producción usar POST con CSRF)
  public function delete(array $params = []) {
    $id = isset($params[0]) ? (int)$params[0] : (int)($_REQUEST['id'] ?? 0);
    if ($id) {
      $this->model->delete($id);
      flash_set('success', 'Cliente eliminado.');
    }
    header('Location: /clientes');
    exit;
  }
}
```

5) paso5/app/views/layout.php

Plantilla que recibirá $title y $content (la función render_view lo incrusta). Incluye header.php y footer.php dentro del layout o directamente muestra el layout content. Debe mostrar el título del paso y la estructura en footer.

```
<?php
// paso5/app/views/layout.php
// Variables disponible: $title, $content
?>
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title><?= esc($title ?: 'Aplicación') ?></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:#f2f7ff;padding:10px;border-radius:6px}
    .flash{padding:8px;margin:8px 0;border-radius:4px}
    .flash-success{background:#e6ffef;border:1px solid #b8f0c8}
    .flash-error{background:#ffecec;border:1px solid #f0b8b8}
    main{margin-top:10px}
    footer{margin-top:18px;font-size:0.9em;color:#666}
    .errors{color:#b00}
  </style>
</head>
<body>
<header>
  <h1><?= esc($title ?: 'Paso 5 — Plantilla') ?></h1>
  <p class="small">Layout: vistas incrustadas dentro de <code>views/layout.php</code>. Helpers: CSRF, esc, flash.</p>
</header>
```

```
<?php
// mostrar flashes
if ($msg = flash_get('success')) {
    echo '<div class="flash flash-success">' . esc($msg) . '</div>';
}
if ($msg = flash_get('error')) {
    echo '<div class="flash flash-error">' . esc($msg) . '</div>';
}
?>

<main>
<?= $content // contenido de la vista ?>
</main>

<footer>
  <hr>
  <p> Estructura de este paso:</p>
  <pre>
paso5/
  app/
    bootstrap.php
    helpers.php
    controllers/
      ClienteController.php
    models/
      ClienteModel.php
    views/
      layout.php
      header.php
      footer.php
```

```
    clientes/
      list.php
      form.php
  public/
    index.php
    .htaccess
  README.md
```

```
<p class="small">En producción: usar POST para borrado, tokens CSRF y validación más exhaustiva.</p>
</footer>
</body>
</html>
```

---

6) paso5/app/views/clientes/list.php

Vista usada por render_view. Debe asumir $clientes en scope; render_view incluirá esta vista y capturará su contenido como $content.

```
<?php
// paso5/app/views/clientes/list.php
// Variables esperadas: $clientes
?>
<p>
  <a class="btn" href="/clientes/add">Añadir cliente</a>
  <a class="btn" href="/clientes">Refrescar</a>
</p>

<?php if (empty($clientes)): ?>
  <p>No hay clientes.</p>
<?php else: ?>
  <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 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/<?= $c['id'] ?>" onclick="return confirm('¿Borrar cliente #<?= esc($c['id']) ?>?')">Borrar</a>
       </td>
     </tr>
   <?php endforeach; ?>
  </tbody>
 </table>
<?php endif; ?>
```

---

7) paso5/app/views/clientes/form.php

Formulario para add/edit. Usa $action, $cliente, $errors, $csrf pasados por el controlador. Muestra errores en pantalla.

```
<?php
// paso5/app/views/clientes/form.php
// Variables: action, cliente, errors, csrf
?>
<?php if (!empty($errors)): ?>
```

```
 <div class="errors">
  <ul>
   <?php foreach ($errors as $e): ?>
    <li><?= esc($e) ?></li>
   <?php endforeach; ?>
  </ul>
 </div>
<?php endif; ?>

<form method="post" action="" style="max-width:600px">
 <?= csrf_field_html() // campo CSRF ?>
 <?php if ($action === 'edit'): ?>
  <input type="hidden" name="id" value="<?= esc($cliente['id']) ?>">
 <?php endif; ?>

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

 <button type="submit"><?= $action === 'edit' ? 'Guardar' : 'Crear' ?></button>
 <a class="btn" href="/clientes">Cancelar</a>
</form>
```

---

8) paso5/public/index.php

Front controller (router) para Paso 5. Incluye bootstrap y despacha a ClienteController. Igual que en paso4 pero con bootstrap nuevos.

```
<?php
// paso5/public/index.php
```

```php
require __DIR__ . '/../app/bootstrap.php';

// parsear la URL
$uri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
$scriptName = str_replace('\\', '/', dirname($_SERVER['SCRIPT_NAME']));
$base = rtrim($scriptName, '/');

if ($base && strpos($uri, $base) === 0) {
    $path = substr($uri, strlen($base));
} else {
    $path = $uri;
}
$path = trim($path, '/');
$segments = $path === '' ? [] : explode('/', $path);

$resource = $segments[0] ?? 'clientes';
$action = $segments[1] ?? 'index';
$params = array_slice($segments, 2);

// dispatch
switch ($resource) {
    case 'clientes':
        $controller = new ClienteController($pdo);
        if ($action === 'add') $controller->add();
        elseif ($action === 'edit') $controller->edit($params);
        elseif ($action === 'delete') $controller->delete($params);
        else $controller->index();
        break;
    default:
        header("HTTP/1.0 404 Not Found");
        echo "404 - Recurso no encontrado";
```

```
    break;
}
```

---

9) paso5/public/.htaccess (opcional)

```
# paso5/public/.htaccess
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^ index.php [QSA,L]
```

---

10) paso5/README.md

```
# Paso 5 — Layout, helpers y seguridad básica

## Qué se ha hecho

## Estructura
```
