**Panel de administración**

**🔹 Plan de desarrollo:**

1️⃣ **Verificar sesión y rol** → Si el usuario no es admin, redirigirlo fuera.  
2️⃣ **Conectar con la base de datos y obtener los usuarios**.  
3️⃣ **Mostrar los usuarios en una tabla** con botones de acción.  
4️⃣ **Agregar estilos básicos para la tabla** (opcional).

### ✅ **Buenas prácticas para** admin\_dashboard.php

Para mantener un código limpio y modular, es mejor separar las partes así:

1️⃣ **Encabezado (header.php)**

* Contiene <head>, <nav>, estilos y scripts comunes.
* <!DOCTYPE html>
* <html lang="es">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>CRM - Panel Admin</title>
* <link rel="stylesheet" href="../estilos.css">
* </head>
* <body>
* <nav>
* <a href="admin\_dashboard.php">Inicio</a>
* <a href="../logout.php">Cerrar Sesión</a>
* </nav>

2️⃣ **Dashboard (admin\_dashboard.php)**

* Solo tiene la lógica para mostrar la lista de usuarios.
* Incluye el header.php y footer.php.
* <?php
* include 'includes/header.php'; // Agrega la cabecera
* session\_start();
* include 'conexion.php';
* // Verificar si el usuario es administrador
* if ($\_SESSION["usuario\_rol"] !== "admin") {
* header("Location: ../index.php");
* exit();
* }
* // Obtener usuarios de la base de datos
* $sql = "SELECT id, nombre, email, rol, estado FROM usuarios";
* $result = $conexion->query($sql);
* ?>
* <h2>Panel de Administración</h2>
* <table>
* <tr>
* <th>ID</th>
* <th>Nombre</th>
* <th>Email</th>
* <th>Rol</th>
* <th>Estado</th>
* <th>Acciones</th>
* </tr>
* <?php while ($usuario = $result->fetch\_assoc()): ?>
* <tr>
* <td><?= $usuario['id'] ?></td>
* <td><?= $usuario['nombre'] ?></td>
* <td><?= $usuario['email'] ?></td>
* <td><?= $usuario['rol'] ?></td>
* <td><?= $usuario['estado'] ? 'Activo' : 'Inactivo' ?></td>
* <td>
* <a href="editar\_usuario.php?id=<?= $usuario['id'] ?>">Editar</a>
* <a href="eliminar\_usuario.php?id=<?= $usuario['id'] ?>" onclick="return confirm('¿Seguro que deseas eliminar este usuario?')">Eliminar</a>
* </td>
* </tr>
* <?php endwhile; ?>
* </table>
* <?php include 'includes/footer.php'; // Agrega el pie de página ?>

3️⃣ **Pie de página (footer.php)**

* Cierra etiquetas <body> y <html>.
* <footer style="text-align: center; padding: 10px; background-color: #f1f1f1; position: fixed; bottom: 0; width: 100%;">
* <p>&copy; <?php echo date('Y'); ?> CRM - Todos los derechos reservados</p>
* </footer>
* </body>
* </html>

### **¿Siguiente paso?**

1️⃣ **Proteger admin\_dashboard.php** para que solo el admin pueda verlo.  
2️⃣ **Diseñar mejor la tabla con CSS.**  
3️⃣ **Crear editar\_usuario.php y eliminar\_usuario.php.**

Para agregar opciones de edición y eliminación de usuarios, vamos a modificar admin\_dashboard.php y crear dos nuevos archivos:

1️⃣ **Actualizar admin\_dashboard.php** → Agregar botones de editar/eliminar.  
2️⃣ **Crear editar\_usuario.php** → Para actualizar usuarios.  
3️⃣ **Crear eliminar\_usuario.php** → Para eliminar usuarios.  
4️⃣ **Modificar fetch\_usuarios.php** → Para traer la lista de usuarios dinámicamente con fetch().

## ****📂 Estructura y Propósito de los Archivos****

### **1️⃣ Carpeta** css/

* **styles.css** → Contiene los estilos generales del sistema, aplicados a todas las páginas del CRM.

### **2️⃣ Carpeta** includes/

* **footer.php** → Pie de página reutilizable en todas las vistas.
* **header.php** → Encabezado con la barra de navegación o menú principal.

### **3️⃣ Carpeta** js/

* **admin\_dashboard.js** → Controla la interacción en el panel de administración, incluyendo la carga de usuarios, edición y eliminación usando fetch().
* **login.js** → Maneja la validación y envío del formulario de login mediante fetch(), verificando credenciales en el backend.

### **4️⃣ Carpeta** CRM/ **(Archivos PHP y HTML principales)**

* **actualizar\_usuario.php** → Permite modificar la información de un usuario en la base de datos.
* **admin\_dashboard.php** → Panel de administración donde se gestionan los usuarios (listar, editar, eliminar).
* **agregar\_usuario.php** → Formulario para agregar nuevos usuarios al sistema.
* **conexion.php** → Establece la conexión con la base de datos MySQL.
* **editar\_usuario.php** → Página para editar la información de un usuario específico.
* **eliminar\_usuario.php** → Script para eliminar un usuario de la base de datos.
* **fetch\_usuario.php** → Obtiene y devuelve la lista de usuarios en formato JSON para ser utilizada por fetch().
* **index.html** → Página de inicio (posiblemente el formulario de login).
* **insert\_admin.php** → Permite registrar nuevos administradores en la base de datos.
* **login.php** → Procesa la autenticación de usuarios y administra sesiones.
* **usuarios.php** → Devuelve los datos de los usuarios en formato JSON para ser consumidos en el dashboard.

## ****🔄 Flujo del Login****

1. **El usuario ingresa su correo y contraseña en index.html.**
2. **login.js captura el evento del formulario y envía los datos a login.php usando fetch().**
3. **En login.php**, se valida el usuario:
   * Si las credenciales son correctas, se inicia una sesión ($\_SESSION).
   * Si son incorrectas, se devuelve un mensaje de error.
4. **Si el login es exitoso**, el usuario es redirigido a admin\_dashboard.php.
5. **Si falla, login.js muestra un mensaje de error sin recargar la página.**

## ****📊 Flujo del Panel de Administración (****admin\_dashboard.php****)****

1. **Cuando admin\_dashboard.php carga**, ejecuta admin\_dashboard.js.
2. **admin\_dashboard.js usa fetch() para obtener la lista de usuarios desde fetch\_usuario.php.**
3. **Los datos se muestran en una <table> dentro de admin\_dashboard.php.**
4. **Opciones del panel:**
   * **Editar usuario** → Redirige a editar\_usuario.php?id=ID para modificar los datos.
   * **Eliminar usuario** → Llama a eliminar\_usuario.php?id=ID y luego actualiza la lista.