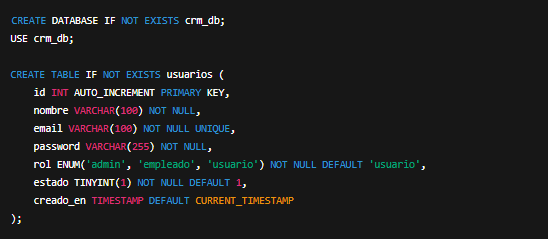
**Sistema de Gestión de Clientes (CRM- Customer Relationship Management) - Implementación del Login**

**1. Estructura de la Base de Datos (Tabla usuarios)**

Crearemos una tabla llamada usuarios con los siguientes campos:

| **Campo** | **Tipo de Dato** | **Descripción** |
| --- | --- | --- |
| id | INT (PK, AI) | Identificador único del usuario |
| nombre | VARCHAR(100) | Nombre completo del usuario |
| email | VARCHAR(100) | Correo electrónico (debe ser único) |
| password | VARCHAR(255) | Contraseña encriptada |
| rol | ENUM('admin', 'empleado', 'usuario') | Tipo de usuario |
| estado | TINYINT(1) | 1 = Activo, 0 = Inactivo |

**Nota:** Usaremos password\_hash() para almacenar las contraseñas de forma segura.



### Explicación:

* **id INT AUTO\_INCREMENT PRIMARY KEY** → Identificador único.
* **nombre VARCHAR(100) NOT NULL** → Nombre del usuario.
* **email VARCHAR(100) NOT NULL UNIQUE** → Correo único para autenticación.
* **password VARCHAR(255) NOT NULL** → Contraseña encriptada.
* **rol ENUM('admin', 'empleado', 'usuario') NOT NULL DEFAULT 'usuario'** → Control de acceso.
* **estado TINYINT(1) NOT NULL DEFAULT 1** → 1 = Activo, 0 = Inactivo.
* **creado\_en TIMESTAMP DEFAULT CURRENT\_TIMESTAMP** → Guarda la fecha y hora de creación.

📌 **Nota:** Recuerda que las contraseñas deben guardarse con password\_hash() para mayor seguridad.

**2. Proceso del Inicio de Sesión (Lógica Completa)**

**Paso 1: Formulario de Login**

* Se solicitará **email** y **contraseña**.
* Validación en **JavaScript** antes de enviarlo al backend.

**Paso 2: Envío de Datos al Servidor (PHP + MySQL)**

* Verificamos si el usuario existe en la base de datos.
* Si el usuario **no existe**, mostramos un mensaje de error.
* Si el usuario **existe**, validamos la contraseña con password\_verify().
* Si la contraseña es incorrecta, mostramos un mensaje de error.
* Si la contraseña es correcta, iniciamos sesión y guardamos los datos del usuario en $\_SESSION.

**Paso 3: Redirección Según el Rol**

* **Administrador:** Redirigimos al panel de administración.
* **Empleado:** Redirigimos a su dashboard con acceso limitado.
* **Usuario estándar:** Accede solo a información permitida.

**Paso 4: Protección de Rutas**

* Si un usuario no está autenticado, lo redirigimos al login.
* Si intenta acceder a un área sin permisos, lo redirigimos según su rol.

**3. Flujo de Lógica en Código (Resumido)**

1. **Usuario ingresa email y contraseña en el formulario.**
2. **JavaScript valida los campos antes de enviarlos al servidor.**
3. **PHP recibe los datos y busca el email en la base de datos.**
4. **Si el usuario existe, compara la contraseña con password\_verify().**
5. **Si es correcta, guarda los datos en $\_SESSION y redirige según su rol.**
6. **Si es incorrecta, muestra un error.**
7. **Si el usuario intenta acceder sin sesión, lo redirige al login.**

**4. Seguridad Adicional**

✅ **Hash de Contraseña:** Usaremos password\_hash() y password\_verify() para proteger las contraseñas. ✅ **Protección de Rutas:** Evitaremos acceso a áreas restringidas con sesiones y roles. ✅ **Medidas Anti-Ataques:** Implementaremos protección contra **SQL Injection** y **XSS**.

Con estos pasos, aseguramos un sistema de login seguro y funcional para nuestro CRM.