Práctica 2 — Mini-CRM de clientes y tareas

1. Objetivo de la práctica

Desarrollar un mini-CRM web que permita gestionar clientes y las tareas asociadas a ellos. El sistema debe incluir control de acceso por roles, gestión de datos mediante formularios, etc.

El resultado debe ser una aplicación funcional, clara y bien estructurada.

2. Descripción general de la aplicación

La aplicación debe incluir:

- Un sistema de login con roles diferenciados.
- Un panel principal con acceso a clientes y tareas.
- Un CRUD de clientes (solo para administradores).
- Un CRUD de tareas (para administradores y empleados).
- Listados de clientes y tareas en tablas HTML.
- Se deberán filtrar, ordenar y generar estadísticas.

3. Requisitos funcionales obligatorios

3.1. Autenticación y roles: el sistema debe tener, como mínimo, dos roles:

- **Administrador**
    - CRUD completo de clientes.
    - CRUD completo de tareas.
- **Empleado**
    - Puede ver todos los clientes.
    - Puede crear, editar y cerrar tareas.
    - No puede eliminar clientes.

Requisitos:

- Formulario de login con usuario y contraseña.
- Comprobación de credenciales contra la base de datos.
- Uso de sesiones para mantener la sesión iniciada.
- Páginas protegidas según rol.
- Página de logout que destruya la sesión.

3.2. Gestión de clientes (CRUD completo)

(Solo para administradores)

Cada cliente debe tener:

- Nombre
- Email
- Teléfono
- Empresa (opcional)

El administrador podrá:

- Crear nuevos clientes
- Editar clientes existentes
- Eliminar clientes
- Listar todos los clientes en una tabla HTML

3.3. Gestión de tareas (CRUD completo)

(Administradores y empleados)

Cada tarea debe tener:

- Cliente asociado
- Título
- Descripción
- Fecha de creación
- Fecha límite
- Estado (pendiente, en curso, completada)

El usuario podrá:

- Crear nuevas tareas
- Editar tareas
- Cambiar su estado
- Eliminar tareas (solo admin)
- Listar tareas en una tabla HTML con filtros

3.4. Posibles estructuras de almacenamiento (tienen libertad si consideran otras formas de estructuración)

A) Array de objetos Tarea. Al listar las tareas:

- Cargar todas las tareas desde la BD
- Permitir filtrar por estado (pendiente, en curso, completada)
- Permitir ordenar por fecha límite

B) Estadísticas básicas

Generar estadísticas usando arrays, por ejemplo:

- Número de tareas por estado
- Número de tareas por cliente
- Tareas vencidas (fecha límite pasada)

Estas estadísticas pueden mostrarse en el panel principal.

3.5. Programación orientada a objetos. Debes crear, como mínimo, las clases:

Clase Cliente

- Propiedades básicas
- Constructor
- Métodos getters/setters

Clase Tarea

- Propiedades: cliente, título, descripción, fechas, estado
- Constructor
- Método estaVencida() → true si la fecha límite ya pasó
- Método diasRestantes() → devuelve cuántos días faltan
- Métodos getters/setters

3.6. Validación de formularios

- Nombre del cliente obligatorio
- Email válido
- Teléfono con formato correcto
- Título de tarea obligatorio
- Fecha límite válida y no anterior a la fecha actual
- Estado válido (lista predefinida)

Se pueden usar expresiones regulares donde sea adecuado.

4. Extras opcionales

- Sistema de comentarios dentro de cada tarea
- Buscador de clientes por nombre o email
- Filtro avanzado de tareas por cliente + estado + fecha límite