



Agenda Médica Inteligente CrudClinic

La empresa **Crudzaso** está desarrollando **CrudClinic**, una aplicación para administrar la agenda médica de su clínica.

Actualmente, la información de las citas se guarda en un archivo Excel con datos redundantes, inconsistentes y sin un modelo de datos definido. Esto genera:

- Dificultad para buscar información.
- Errores en las reservas.
- Información duplicada.
- Falta de control sobre pacientes, médicos y pagos.

El objetivo del proyecto es **normalizar la información, diseñar un modelo de base de datos relacional** y desarrollar una **aplicación web** con **login seguro, CRUD de citas, consultas avanzadas y carga masiva de datos desde CSV**.

Archivo proporcionado

Recibirán un archivo [data_crudclinic.xlsx](#) que contiene **errores e inconsistencias** que deberán identificar y corregir en el proceso de normalización.



Requerimientos Técnicos

1. Trabajo Colaborativo

- El grupo estará conformado por 2 integrantes que participarán en todas las etapas del desarrollo.
- Usar **Azure DevOps Boards** para planificar tareas y dar seguimiento al avance del proyecto.
- Dividir las tareas por funcionalidades, no por roles.

2. Normalización y Base de Datos

- Analizar el archivo y aplicar **hasta 3FN** para obtener un diseño óptimo.
- Crear un **modelo entidad-relación**.
- Generar un **script SQL** para **MySQL** o **PostgreSQL** con:
 - Llaves primarias y foráneas.
 - Tipos de datos adecuados.
 - Restricciones (NOT NULL, UNIQUE, etc.).

3. Backend

- API REST desarrollada con **Node.js + Express**.
- Uso obligatorio de **variables de entorno** (.env) para credenciales y configuración.

Endpoints CRUD para:

- **Citas**
- **Pacientes**
- **Médicos**



4. Frontend

- HTML + Bootstrap (u otro framework CSS permitido).
- Login con validación de credenciales (usuario).
- CRUD de citas que permita:
 - Agregar nuevas citas.
 - Editar citas existentes.
 - Eliminar citas.
 - Listar todas las citas.

5. Carga de Datos desde CSV

- Implementar carga masiva de datos a través de:
 - **Script en Node.js** que lea un archivo CSV y cargue los datos en la base de datos (Manual) o **Endpoint** que reciba un archivo CSV mediante **Multer** y lo procese (**Opcional**).
- Validar datos antes de insertarlos.
- **Punto adicional:** permitir carga desde **XLSX**.

6. Consultas Avanzadas

Implementar endpoints que permitan:

1. Listar todas las citas con:
 - Paciente, médico, especialidad, fecha y hora, estatus.
2. Filtrar citas por médico y rango de fechas.
3. Obtener pacientes con más de 3 citas registradas.
4. Listar médicos con el número de citas atendidas en el último mes.
5. Consultar ingresos generados por método de pago en un rango de fechas.



Entregables

1. Documento con:
 - o Proceso de **normalización**.
 - o Modelo entidad-relación.
 - o Planificación en Azure DevOps.
2. Script SQL de creación de tablas.
3. API en Node.js con CRUD y consultas avanzadas.
4. Frontend con login y CRUD funcional.
5. Implementación de carga desde CSV (y opcional XLSX).
6. Repositorio Git con instrucciones de instalación y ejecución.