



**Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Puebla**

Práctica 1

Transformación digital de organizaciones de la sociedad civil

Profesor: David Augusto Céspedes Hernández

Emilio Santiago Castillo Sanchez | A01734624

Carlos Eduardo Osorio González | A01734512

Rolando Rodriguez Santiago | A01734383

Joel Sotres Rodriguez | A01734648

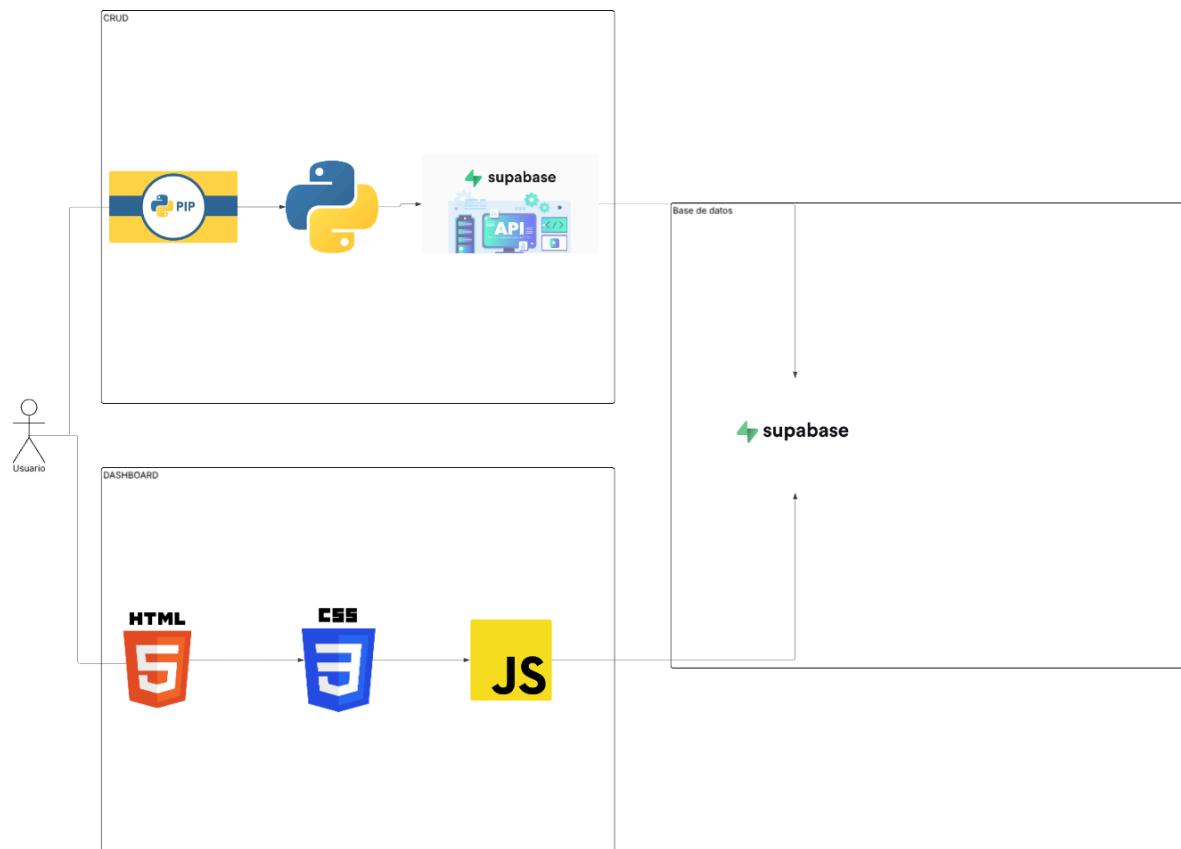
Anhuar Maldonado Aguilar | A01737767

Juan Daniel Salmerón Mora | A01737637

Fecha de entrega:

23 de febrero de 2026

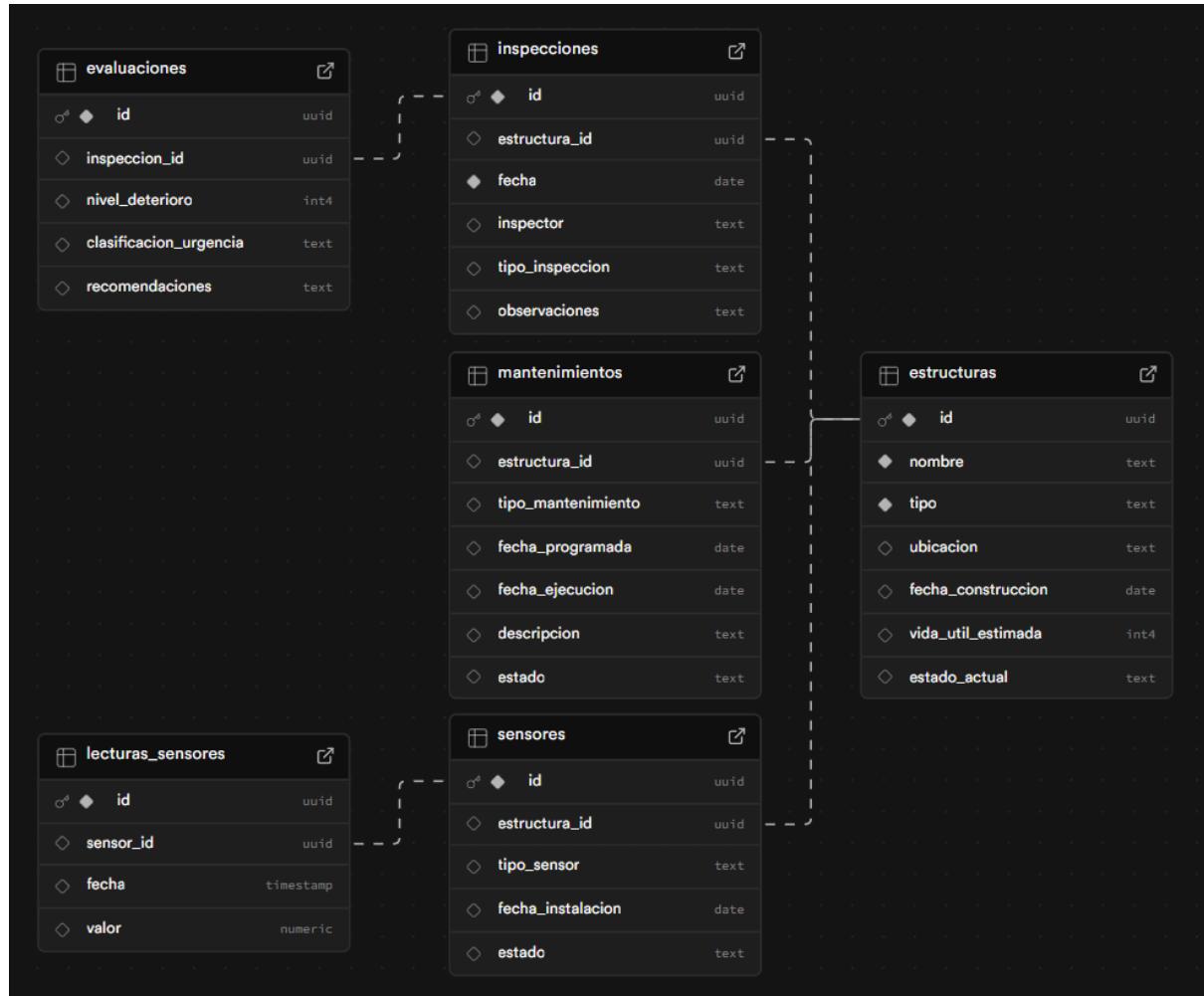
## Diagrama de arquitectura del sistema completo



Link a diagrama:

[https://lucid.app/lucidchart/15b5013e-0827-4655-82ec-6084237d6f43/edit?view\\_items=4Fkyhknr9agx&page=upkyqUCXhKcd&invitationId=inv\\_ab854fc7-023c-40c1-86a3-ccd4e7546d83](https://lucid.app/lucidchart/15b5013e-0827-4655-82ec-6084237d6f43/edit?view_items=4Fkyhknr9agx&page=upkyqUCXhKcd&invitationId=inv_ab854fc7-023c-40c1-86a3-ccd4e7546d83)

## Esquema de base de datos con explicaciones de diseño



El modelo de datos fue diseñado para gestionar información relacionada con infraestructura vial. Su propósito es permitir el registro estructurado de:

- Infraestructuras (estructuras viales)
- Inspecciones realizadas
- Evaluaciones del estado
- Mantenimientos programados y ejecutados
- Sensores instalados
- Lecturas generadas por los sensores

El modelo sigue un enfoque relacional, utilizando claves primarias (PK) para identificar registros de forma única y claves foráneas (FK) para mantener la integridad referencial entre tablas. Se buscó mantener una estructura normalizada que evite duplicidad de información y facilite el análisis posterior mediante consultas y dashboards.

El modelo está organizado de forma que cada tabla cumple una función específica. Se utilizan claves primarias para identificar registros únicos y claves foráneas para

asegurar que las relaciones entre tablas sean válidas. La separación de información en distintas tablas evita duplicidad de datos y mejora la claridad del sistema. Además, el diseño permite almacenar información histórica, lo cual es importante para análisis de mantenimiento, evolución del deterioro y monitoreo por sensores. En conjunto, la base de datos es coherente, organizada y adecuada para implementar operaciones CRUD y visualizaciones en un dashboard.

#### ***Tabla: estructuras***

La tabla estructuras es la tabla principal del sistema. Su clave primaria (PK) es el campo id, de tipo UUID, el cual identifica de forma única cada estructura. Esta tabla no contiene claves foráneas, ya que es la entidad principal de la cual dependen otras tablas.

Sus componentes son los siguientes campos: id, nombre, tipo, ubicación, fecha construcción, vida útil estimada y estado actual.

Esta tabla almacena la información general de cada infraestructura vial. Se decidió usar UUID para permitir escalabilidad y evitar duplicados. La vida útil estimada es un número entero para facilitar cálculos futuros. A partir de esta tabla se relacionan inspecciones, mantenimientos y sensores.

#### ***Tabla: inspecciones***

La tabla de inspecciones almacena los registros de revisiones realizadas a las estructuras. Su clave primaria (PK) es id. Su clave foránea (FK) es estructura\_id, la cual se conecta con estructuras.id.

Sus componentes son: id, estructura\_id, fecha, inspector, tipo\_inspeccion y observaciones.

Esta tabla permite que una estructura tenga varias inspecciones a lo largo del tiempo. Se separó de la tabla evaluaciones para mantener mejor organización y permitir un registro más claro del proceso de revisión.

#### ***Tabla: evaluaciones***

La tabla evaluaciones guarda el resultado técnico de cada inspección. Su clave primaria (PK) es id. Su clave foránea (FK) es inspeccion\_id, que se conecta con inspecciones.id.

Sus componentes son: id, inspeccion\_id, nivel\_deterioro, clasificacion\_urgencia y recomendaciones.

Se decidió separar esta tabla de inspecciones para mantener un diseño más ordenado. El campo nivel\_deterioro es un número entero que puede representar una escala de evaluación. Esto facilita comparaciones entre diferentes inspecciones.

### ***Tabla: mantenimientos***

La tabla mantenimientos registra las acciones realizadas o programadas para cada estructura. Su clave primaria (PK) es id. Su clave foránea (FK) es estructura\_id, que se conecta con estructuras.id.

Sus componentes son: id, estructura\_id, tipo\_mantenimiento, fecha\_programada, fecha\_ejecucion, descripcion y estado.

Se incluyen dos fechas para poder analizar si el mantenimiento se realizó conforme a lo planeado. El campo estado permite identificar si el mantenimiento está pendiente, en proceso o completado.

### ***Tabla: sensores***

La tabla sensores registra los dispositivos instalados en cada estructura. Su clave primaria (PK) es id. Su clave foránea (FK) es estructura\_id, que se conecta con estructuras.id.

Sus componentes son: id, estructura\_id, tipo\_sensor, fecha\_instalacion y estado.

Esta tabla permite que una estructura tenga varios sensores. El campo estado ayuda a saber si el sensor está activo o fuera de servicio.

### ***Tabla: lecturas\_sensores***

La tabla lecturas\_sensores almacena los valores registrados por cada sensor. Su clave primaria (PK) es id. Su clave foránea (FK) es sensor\_id, que se conecta con sensores.id.

Sus componentes son: id, sensor\_id, fecha y valor.

El campo fecha es de tipo timestamp para poder hacer análisis por tiempo. El campo valor es numérico para permitir decimales. Esta tabla permite que cada sensor tenga múltiples lecturas, formando un historial de datos.

## Listado de prompts utilizados con evaluaciones

### **Prompt 1:**

*“Genera datos de ejemplo en formato CSV compatibles con Supabase/PostgreSQL.*

*Reglas obligatorias:*

- *No incluyas la columna id si es tipo UUID (la base la genera).*
- *Usa fechas en formato YYYY-MM-DD.*
- *Respeta claves foráneas existentes.*
- *No uses caracteres especiales raros.*
- *Entrega solo texto CSV listo para importar.”*

El prompt funciona muy bien porque establece reglas claras que evitan los errores más comunes al importar datos en Supabase/PostgreSQL, especialmente en temas sensibles como UUID y fechas.

### **Prompt 2:**

*“Genera un INSERT en la tabla con datos realistas de puentes, túneles y carreteras, usando el mismo formato mostrado y estados como bueno, regular o deteriorado para poder añadir más datos a una base de datos” \*Se añadió imagen de la base de datos como ejemplo\**

El prompt fue adecuado y generó un resultado apropiado. Sin embargo, posteriormente fue necesario solicitar a la IA algunos ajustes, específicamente en el formato de las ID, para que fueran diferentes entre sí y al mismo tiempo mantuvieran un patrón similar a los datos ya existentes en la tabla “estructuras” de la base de datos.

Finalmente, se realizó una corrección manual, ya que la IA asignó un nombre distinto a la tabla. Por ello, únicamente se modificó ese detalle de manera directa para mantener coherencia con el esquema original de la base de datos.

### **Prompt 3:**

*“En base a la base de datos de estructuras, genera datos aleatorios pero coherentes para la tabla de mantenimiento. No es necesario que ambas tablas tengan la misma cantidad de registros” \*Se le dio el CSV de la tabla de estructuras y se le dio una imagen de los datos ya existentes para la tabla de mantenimiento\**

Después de ingresar el prompt, aunque sí se generó un INSERT, fue necesario hacer varias correcciones. Las ID de las estructuras no se estaban tomando correctamente, ya que la IA las estaba inventando y esos datos realmente no existían. Luego se le tuvo que pedir que las ID de los mantenimientos fueran consistentes con los ID que ya existían.

#### **Prompt 4:**

*“Voy a realizar un dashboard sencillo de una base de datos, cuál sería la forma más sencilla de realizarlo, dame los frameworks que sean más sencillos para esta tarea.”*

Después de ingresar el prompt, chat nos dió una explicación acerca de los frameworks más sencillos y al final tomamos una decisión de hacerlo con HTML, CSS (bootstrap), JS puro.

#### **Prompt 5:**

*“Ahora bien, en unos momentos voy a tener una base de datos de la cuál voy a obtener datos, estará hecha en Supabase.”*

Esto es una explicación acerca de cómo realizar la conexión de manera un poco más sencilla, para que a la hora de tener las llaves pudiéramos conseguir el objetivo y obtener los datos.

### **Manual de usuario básico con capturas de pantalla**

#### **1. Objetivo del sistema**

- El objetivo del dashboard es:
- Facilitar la toma de decisiones
- Monitorear el estado de las estructuras
- Visualizar tendencias de inspección y mantenimiento
- Analizar el nivel de deterioro

#### **2. Requisitos del sistema**

- El usuario solo necesita:
- Un navegador web moderno:
- Google Chrome
- Microsoft Edge
- Firefox
- Conexión a internet
- No se requiere instalación adicional.

#### **3. Interfaz del Dashboard**

Al ingresar al sistema, el usuario visualizará:

- ◆ 3.1 Sección de KPIs (Indicadores clave)

Se muestran 4 indicadores principales:

KPI	Descripción
 Estructuras	Total de estructuras registradas

-  Inspecciones Número de inspecciones realizadas
  -  Mantenimientos Total de mantenimientos ejecutados
  -  Sensores Sensores instalados en las estructuras
- 3.2 Gráfica: Inspecciones por mes  
Tipo: Gráfica de barras

Muestra el número de inspecciones realizadas por mes

Permite identificar temporadas con mayor actividad

-  3.3 Gráfica: Mantenimientos por estado

Tipo: Gráfica circular (pie)

Clasifica los mantenimientos según su estado:

Ejemplo:

Pendiente

En proceso

Finalizado

-  3.4 Gráfica: Nivel de deterioro promedio

Tipo: Gráfica tipo doughnut

Representa el promedio de deterioro de las estructuras

Escala del 0 al 10

#### 4. Actualización de datos

El dashboard obtiene los datos en tiempo real desde la base de datos de Supabase.

Cada vez que se recarga la página:

- Se consultan los datos actualizados
- Se recalculan los indicadores
- Se actualizan las gráficas

## Listado de dependencias y requisitos del sistema

### CRUD

#### Requisitos previos

Antes de comenzar, asegúrate de tener instalado:

- Python 3.8 o superior
- pip (gestor de paquetes de Python)

Puedes verificarlo con los siguientes comandos:

- python --version
- pip --version

### Instalación de dependencias

Para ejecutar el proyecto es necesario instalar las siguientes librerías:

- pip install rich
- pip install supabase

### *Dashboard*

#### Requisitos previos

Antes de comenzar, asegúrate de contar con lo siguiente:

#### URL del proyecto en Supabase

#### Clave pública (anon key) de Supabase

Estas credenciales son necesarias para establecer la conexión entre la aplicación web y la base de datos.