Sprint 1: Estructura técnica y conexión a base de datos

🎯 *OBJETIVO DEL SPRINT*

Tener lista la parte técnica inicial del sistema de asistencias:

- *Diseño visual (maqueta)*
- *Diagrama de base de datos (DER)*
- *Carpeta del proyecto funcionando con Flask*
- *Conexión a base de datos MySQL (local o nube)*
- *Pruebas basicas de funcionamiento*

✅ *ENTREGABLES COMPLETOS*

1. 🖼️ Maqueta digital

¿Qué tienes que tener?

- ***Pantalla de inicio** :*
  Título del sistema: "Sistema de Registro de Asistencias"
  Menú: Inicio | Registrador | Informes

- ***Formulario de asistencia** :*
  - *Nombre del alumno*
  - *Apellido*
  - *Fecha*
  - *Curso*
  - *Estado de asistencia (Presente / Ausente)*

Herramientas sugeridas:
Figma
Canva

2. Diagrama de base de datos (DER)

Tablas:

Tabla: alumnos

- *id_alumno(INT, PK, AUTO_INCREMENTO)*
- *nombre(VARCHAR 50)*
- *apellido(VARCHAR 50)*
- *curso(VARCHAR 30)*

Tabla: asistencias

- *id_asistencia(INT, PK, AUTO_INCREMENTO)*
- *fecha(FECHA)*
- *estado(ENUM: 'Presente', 'Ausente')*
- *id_alumno(INT, FK –alumnos.id_alumno)*

Relación:

- *Un alumno tiene muchas asistencias. (1 :N)*

Estructura del proyecto (Flask)

intento

```
/asistencias_app/
│
├── /static/              # CSS, imágenes
│   └── style.css
├── /templates/           # HTML con Jinja
│   ├── index.html
│   └── formulario.html
├── app.py                # Código principal de Flask
├── database.py           # Conexión con MySQL
├── requirements.txt      # Dependencias
└── README.md             # Documentación
```

4. Base de datos en la nube

Opciones:

- ***XAMPP (localhost)***
- ***Infinito libre***
- ***Base de datos Freesql***

Conexión a base MySQL con Python:

```
pitón

# database.py
import mysql.connector

def conectar():
    try:
        conexion = mysql.connector.connect(
            host="sqlXXXX.infinityfree.com",  # o localhost
            user="tu_usuario",
            password="tu_contraseña",
            database="nombre_de_la_base"
        )
        print("✅ Conexión exitosa")
        return conexion
    except mysql.connector.Error as err:
        print("❌ Error al conectar a la base:", err)
        return None
```

□ TAREAS DE PRUEBA

□ Pruebas de conexión

- Probar conexión con usuario o clave incorrecta → mostrar error controlado.
- Verificar si la conexión correcta muestra un mensaje como “Conexión exitosa”.
- Tiempo de respuesta aceptable (menos de 1 segundo).

□ Pruebas de la base de datos

- Ejecutar en phpMyAdmin o desde Python:

```sql
SQL

SELECT * FROM alumnos;
DESCRIBE asistencias;
```

Verificar:

- Que existen las tablas `alumnos`y `asistencias`.
- Que los campos tengan los tipos de datos correctos.
- Que la clave foránea `id_alumno`funcione (JOIN correcto).

□ LISTA DE VERIFICACIÓN FINAL:

TAREA	ESTADO
Conexión exitosa con la base de datos (local o nul	✅
Mockup digital terminado	❌
DER (diagrama de base de datos) finalizado	✅
Carpeta del proyecto organizada	✅
Archivo README.md escrito	✅
Inicio del README en inglés	✅
Link del repositorio compartido con la docente	✅