Documentación del Proyecto Gantt 2.0

📌 Introducción

Gantt 2.0 es una aplicación web diseñada para gestionar y visualizar convenios en trámite utilizando una interfaz moderna y dinámica. La aplicación permite la administración de convenios con opciones de visualización en tablas interactivas, edición, eliminación y carga de datos de manera sencilla.

🚀 Tecnologías Utilizadas

1️⃣ Frontend (Interfaz de Usuario)

- **Framework:** Next.js (App Router)
- **Lenguaje:** TypeScript
- **Estilos:** Tailwind CSS
- **Componentes UI:** Prime React
- **Manejo de Estado:** React Hooks

2️⃣ Backend (API y Conexión a Base de Datos)

- **Plataforma:** Next.js API Routes
- **Base de Datos:** MariaDB

3️⃣ Almacenamiento y Hosting

- **Hosting:** Cpanel
- **Manejo de Variables de Entorno:** `.env.local`

📂 Estructura del Proyecto

```
/gantt-2.0
 ├── .env.local          # Variables de entorno
 ├── .gitignore          # Archivos ignorados en Git
 ├── package.json          # Dependencias del proyecto
 ├── tsconfig.json        # Configuración de TypeScript
 ├── next.config.ts       # Configuración de Next.js
 ├── tailwind.config.ts    # Configuración de Tailwind CSS
 ├── src/
 │   ├── app/
```

```
│   │   ├── layout.tsx       # Layout global
│   │   ├── page.tsx         # Página principal
│   ├── lib/
│   │   ├── mariadb.ts       # Configuración de Supabase
│   ├── components/
│   │   ├── ConvenioTable.tsx # Componente de tabla interactiva
│   │   ├── ConvenioForm.tsx  # Formulario para agregar convenios
```

📊 Base de Datos (Supabase - PostgreSQL)

Tablas

```sql
-- Tabla convenios
CREATE TABLE convenios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cooperante VARCHAR(255) NOT NULL,
    sector VARCHAR(255) NOT NULL,
    fase_actual VARCHAR(255) NOT NULL,
    fecha_inicio DATE NOT NULL,
    fecha_final DATE NOT NULL,
    nombre VARCHAR(255) NOT NULL,
    firmado BOOLEAN NOT NULL DEFAULT FALSE,
    consecutivo_numerico INT NOT NULL
);

-- Tabla registro_procesos
CREATE TABLE registro_procesos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    convenio_id INT NOT NULL,
    autoridad_ministerial VARCHAR(255) NOT NULL,
    funcionario_emisor VARCHAR(255) NOT NULL,
    entidad_emisora VARCHAR(255) NOT NULL,
    funcionario_receptor VARCHAR(255) NOT NULL,
    entidad_receptora VARCHAR(255) NOT NULL,
    registro_proceso TEXT NOT NULL,
    fecha_inicio DATETIME NOT NULL,
    fecha_final DATETIME NULL,
    tipo_convenio VARCHAR(255) NOT NULL,
    fase_registro VARCHAR(255) NOT NULL,
    entidad_proponente VARCHAR(255) NOT NULL,
    FOREIGN KEY (convenio_id) REFERENCES convenios(id) ON DELETE CASCADE
);

-- Tabla historial_registro_procesos
CREATE TABLE historial_registro_procesos (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
    registro_proceso_id INT NOT NULL,
    evento TEXT NOT NULL,
    fecha DATETIME NOT NULL,
    FOREIGN KEY (registro_proceso_id) REFERENCES registro_procesos(id) ON
DELETE CASCADE
);

-- Tabla usuario
CREATE TABLE usuario (
    id CHAR(36) PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    correo VARCHAR(255) UNIQUE NOT NULL,
    contrasena TEXT NOT NULL,
    rol VARCHAR(50) NOT NULL,
    fecha_creacion DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP
);

-- Tabla inventario
CREATE TABLE inventario (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cooperante VARCHAR(255) NOT NULL,
    contraparte_externa VARCHAR(255) NOT NULL,
    nombre_convenio VARCHAR(255) NOT NULL,
    objeto_convenio TEXT NOT NULL,
    fecha_rige DATE NOT NULL,
    fecha_vencimiento DATE NOT NULL,
    tipo_instrumento VARCHAR(255) NOT NULL,
    instancias_tecnicas TEXT NOT NULL,
    presupuesto DECIMAL(15,2) NOT NULL,
    informe TEXT NOT NULL,
    documento_pdf VARCHAR(512) NULL -- Para almacenar la ruta del archivo PDF
);
```

🔗 Conexión a Supabase en `supabase.ts`

```
import mariadb from "mariadb";

const pool = mariadb.createPool({
  host: process.env.DATABASE_HOST,
  user: process.env.DATABASE_USER,
  password: process.env.DATABASE_PASSWORD,
  database: process.env.DATABASE_NAME,
```

```
  port: Number(process.env.DATABASE_PORT),
  connectionLimit: 5,
});

export async function query(sql: string, values?: any[]) {
  let conn;
  try {
    conn = await pool.getConnection();
    const rows = await conn.query(sql, values);
    return rows;
  } catch (error) {
    console.error("Error en la consulta:", error);
    throw error;
  } finally {
    if (conn) conn.end();
  }
}

export default pool;
```

🔄 Funcionalidades Clave

✅ **Gestión de convenios**: Crear, leer, actualizar y eliminar convenios.
✅ **Visualización interactiva**: Tablas con filtros, ordenamiento y paginación.
✅ **Integración con MariaDB**: Conexión directa.
✅ **Autenticación de Usuarios (Futuro)**: Login con Supabase Auth.
✅ **Diseño Moderno**: Tailwind CSS.

📜 Instalación y Ejecución

1️⃣ Clonar el repositorio

git clone https://github.com/adrian-sanchez12/Gantt-2.0.git
cd gantt-2.0

2️⃣ Instalar dependencias

npm install

3️⃣ Configurar Variables de Entorno

Crear el archivo `.env.local` en la raíz:

```
DATABASE_HOST=127.0.x.x
```

```
DATABASE_USER=user
DATABASE_PASSWORD=******
DATABASE_NAME=gantt
DATABASE_PORT=3306
```

4️⃣ Ejecutar el servidor de desarrollo

npm run dev

📌 **Abrir en el navegador:** http://localhost:3000

🚀 Despliegue

npm run build
