

# 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

### 1 Frontend (Interfaz de Usuario)

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

### 2 Backend (API y Conexión a Base de Datos)

- **Plataforma:** Next.js API Routes
- **Base de Datos:** Supabase (PostgreSQL)
- **ORM:** Supabase Client ([@supabase/supabase-js](#))(temporal)
- **Autenticación:** Supabase Auth (opcional)

### 3 Almacenamiento y Hosting

- **Base de Datos en la Nube:** Supabase
  - **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/
|   |       |— supabase.ts    # Configuración de Supabase
|   |       |— components/
|   |           |— ConvenioTable.tsx # Componente de tabla interactiva
|   |           |— ConvenioForm.tsx # Formulario para agregar convenios
```

---



## Base de Datos (Supabase - PostgreSQL)

### Tabla: **convenios**

```
CREATE TABLE convenios (
  id SERIAL PRIMARY KEY,
  cooperante TEXT NOT NULL,
  sector TEXT NOT NULL
);

CREATE TABLE registro_procesos (
  id SERIAL PRIMARY KEY,
  convenio_id INT REFERENCES convenios(id) ON DELETE CASCADE,
  autoridad_ministerial TEXT,
  funcionario_emisor TEXT,
  entidad_emisora TEXT,
  funcionario_receptor TEXT,
  entidad_receptora TEXT,
  registro_proceso TEXT,
  fecha_inicio TIMESTAMP,
  fecha_final TIMESTAMP,
  tipo_convenio TEXT
);
```

---



### Conexión a Supabase en **supabase.ts**

```
import { createClient } from "@supabase/supabase-js";

const supabaseUrl = process.env.NEXT_PUBLIC_SUPABASE_URL;
const supabaseKey = process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY;

if (!supabaseUrl || !supabaseKey) {
  throw new Error("Supabase URL y API Key son requeridos.");
}
```

```
}
```

```
export const supabase = createClient(supabaseUrl, supabaseKey);
```

---

## Funcionalidades Clave

- ✓ **Gestión de convenios:** Crear, leer, actualizar y eliminar convenios.
  - ✓ **Visualización interactiva:** Tablas con filtros, ordenamiento y paginación.
  - ✓ **Integración con Supabase:** Conexión directa a PostgreSQL en la nube.
  - ✓ **Autenticación de Usuarios (Futuro):** Login con Supabase Auth.
  - ✓ **Diseño Moderno:** Tailwind CSS.
  - ✓ **Despliegue en Vercel:** Hosting gratuito y escalable. (Alternativa)
- 

## Instalación y Ejecución

### 1 Clonar el repositorio

```
git clone https://github.com/tu-repo/gantt-2.0.git
cd gantt-2.0
```

### 2 Instalar dependencias

```
npm install
```

### 3 Configurar Variables de Entorno

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

```
NEXT_PUBLIC_SUPABASE_URL=https://tuproject.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=tu-api-key-aqui
```

### 4 Ejecutar el servidor de desarrollo

```
npm run dev
```

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

---

## Despliegue

```
npm run build
```

---