



**Diseño y Programación de Software Multiplataforma
DPS941 G01T
Ciclo 01-2025**

PROYECTO 1

Sistema de Gestion de Proyecto

Docente: Alexander Alberto Siguenza Campos

Nombre de integrantes:

Apellidos	Nombres	Carné
Guevara Martínez	Carlos David	GM172474

Fecha de entrega:

30 de Marzo de 2025 (vía Aula Digital)

Introduccion

Este proyecto consiste en el desarrollo de una aplicación web utilizando react y nextjs para la gestión de proyectos y tareas, con las funcionalidades de control de acceso, asignación de responsabilidades y seguimiento del avance de trabajo. Está orientado a facilitar la organización colaborativa de equipos, con roles específicos y una experiencia de usuario clara e intuitiva.

Objetivo General

Desarrollar un sistema web funcional que permita a diferentes tipos de usuarios (Administrador, Gerente, Colaborador) gestionar proyectos, asignar tareas, comentar avances y mantener un control ordenado del flujo de trabajo.

Metodologia

Para el desarrollo del sistema de gestión de proyectos, se utilizó una metodología ágil, enfocada en ciclos iterativos e incrementales. Se siguieron los siguientes pasos:

1. **Levantamiento de Requisitos:** Se definieron las necesidades funcionales y no funcionales del sistema, como autenticación de usuarios, asignación de roles, gestión de tareas y proyectos, entre otros.
2. **Diseño de la arquitectura:** Se estableció una estructura basada en Next.js para el frontend, Sequelize y PostgreSQL en el backend, con autenticación personalizada y gestión de estados con React Query.
3. **Desarrollo modular:** Se dividió el sistema por módulos: autenticación, proyectos, tareas, usuarios, roles, etc., aplicando buenas prácticas de separación de responsabilidades y reutilización de componentes.
4. **Pruebas locales y despliegue:** El sistema se probó localmente y luego fue desplegado en Vercel, utilizando Supabase como servicio de base de datos en la nube.
5. **Mejora continua:** Se realizaron mejoras visuales, validaciones, manejo de errores y optimización de experiencia de usuario de forma iterativa.

Análisis de Resultados

- Se logró construir un sistema funcional donde los usuarios pueden:
 1. Registrarse e iniciar sesión.
 2. Crear, editar y eliminar proyectos.
 3. Asignar usuarios a proyectos y tareas.
 4. Comentar tareas y visualizar su avance.
- Se implementó un panel de administración para cambiar roles y editar correos electrónicos de los usuarios.
- Se incluyó una interfaz intuitiva con dashboard visual, estadísticas y diseño responsivo.
- El sistema fue conectado exitosamente con Supabase para pruebas de producción.

Conclusiones

El sistema cumple con los objetivos planteados: facilitar la gestión de proyectos, asignación de tareas y colaboración entre usuarios; La adopción de herramientas modernas permitió una implementación ágil y escalable.

El sistema se encuentra listo para futuras mejoras como notificaciones por correo, notificaciones de asignaciones de manera interna o de comentarios, control de permisos más completo o integración con servicios externos.

Referencias

[Next.js Documentation](#) Documentacion de next Js

[React Query Documentation](#) Documentacion de React Query

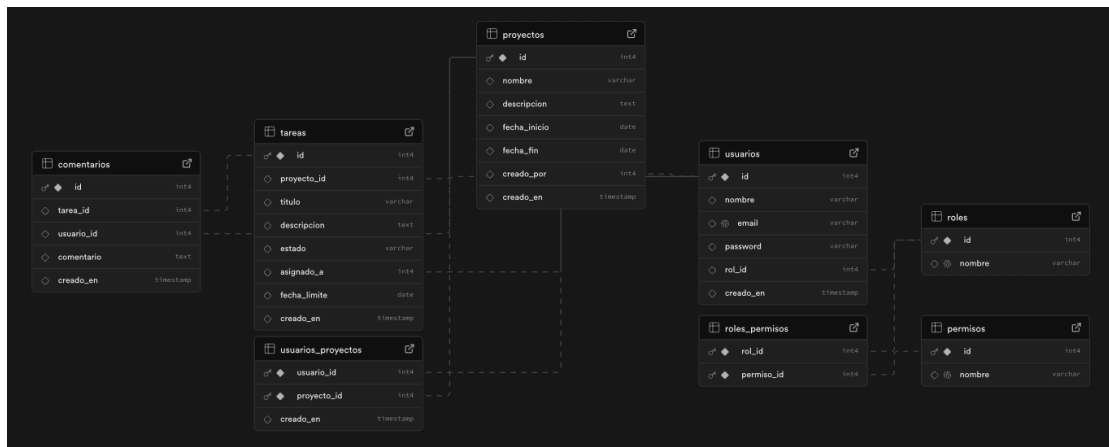
[Tailwind CSS Documentation](#) Documentacion de Tailwind Css para estilo Global

[Sequelize ORM](#) Documentacion Sequelize que sirvio como nuestro ORM

[Supabase](#) Base de datos en linea (paquete gratuito)

[Vercel Deployment](#) Hosting gratuito en el que se encuentra desplegada nuestra web.

Anexos



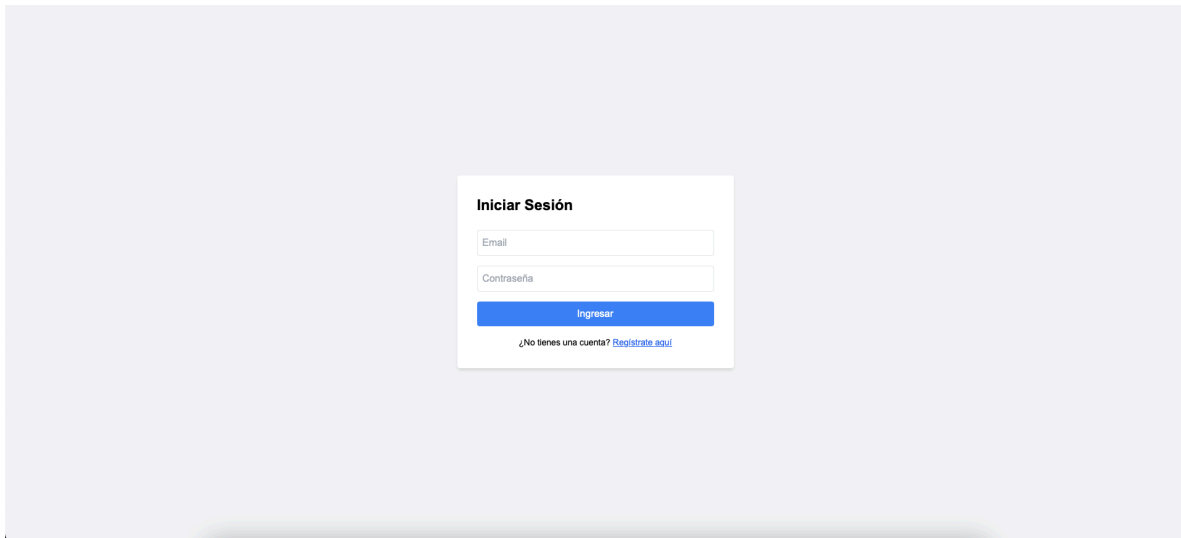
Esquema de base de datos.
Imagen 1, Supabase Schema Visualizer.

```

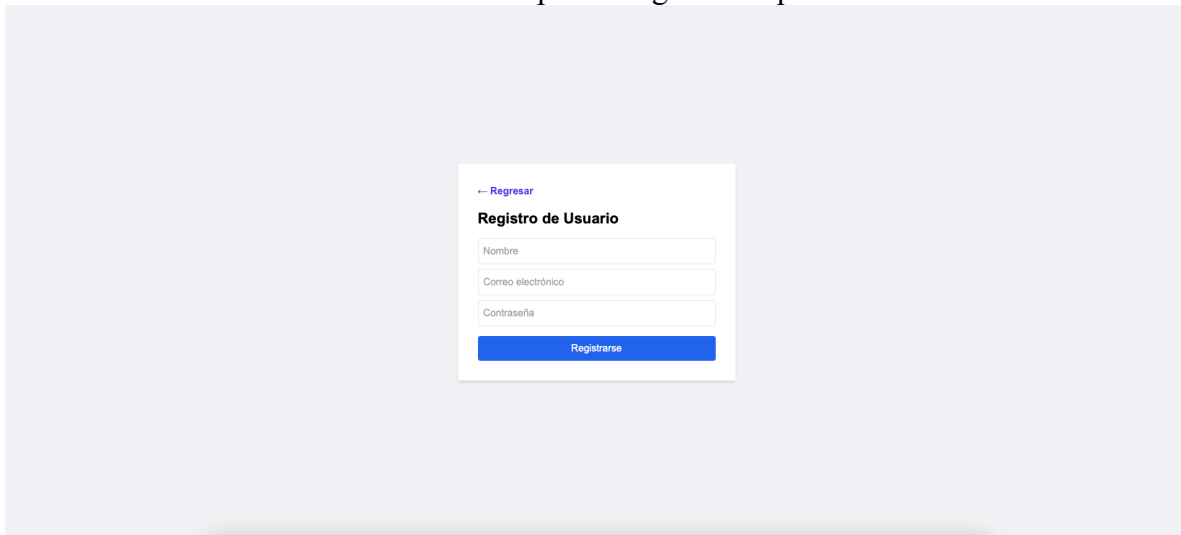
1  sistema-gestion-proyectos/
2  ├── public/                # Archivos estáticos (imágenes, íconos, etc.)
3  ├── src/
4  │   ├── components/       # Componentes reutilizables (Botones, Modales, etc.)
5  │   │   ├── ui/
6  │   │   │   ├── BackButton.js
7  │   │   │   ├── InputField.js
8  │   │   │   ├── Loading.js
9  │   │   │   └── Modal.js
10  │   ├── context/          # Contextos globales
11  │   │   └── AuthContext.js
12  │   ├── hooks/            # Custom hooks
13  │   │   └── useRequireAuth.js
14  │   ├── layouts/          # Layouts como Sidebar o Navbar
15  │   │   └── SidebarLayout.js
16  │   ├── models/           # Modelos Sequelize
17  │   │   ├── index.js
18  │   │   ├── usuario.js
19  │   │   ├── proyecto.js
20  │   │   ├── tarea.js
21  │   │   ├── rol.js
22  │   │   ├── permiso.js
23  │   │   └── comentario.js
24  │   ├── pages/            # Rutas de la app (Next.js)
25  │   │   ├── api/
26  │   │   │   ├── auth/
27  │   │   │   │   ├── login.js
28  │   │   │   │   └── registro.js
29  │   │   │   ├── proyectos/
30  │   │   │   │   ├── index.js
31  │   │   │   │   ├── [id].js
32  │   │   │   │   └── [id]/desasignar.js
33  │   │   │   ├── tareas/
34  │   │   │   │   ├── index.js
35  │   │   │   │   └── [id].js
36  │   │   │   ├── comentarios/
37  │   │   │   │   ├── index.js
38  │   │   │   │   └── [id].js
39  │   │   │   ├── usuarios/
40  │   │   │   │   └── index.js
41  │   │   ├── admin/
42  │   │   │   └── gestion-roles.js
43  │   │   ├── dashboard.js
44  │   │   ├── proyectos/
45  │   │   │   ├── [id].js
46  │   │   │   └── crear.js
47  │   │   ├── tareas/
48  │   │   │   └── [id].js
49  │   │   ├── index.js      # Login
50  │   │   └── registro.js   # Registro
51  │   ├── services/         # Llamadas a la API (axios)
52  │   │   ├── auth.js
53  │   │   ├── proyectos.js
54  │   │   ├── tareas.js
55  │   │   ├── comentarios.js
56  │   │   └── usuarios.js
57  │   ├── styles/           # Estilos globales
58  │   │   └── globals.css
59  ├── .env.local            # Variables de entorno locales
60  ├── .gitignore
61  ├── next.config.js
62  ├── package.json
63  ├── README.md             # Documentación del proyecto
64  └── tailwind.config.js
65

```

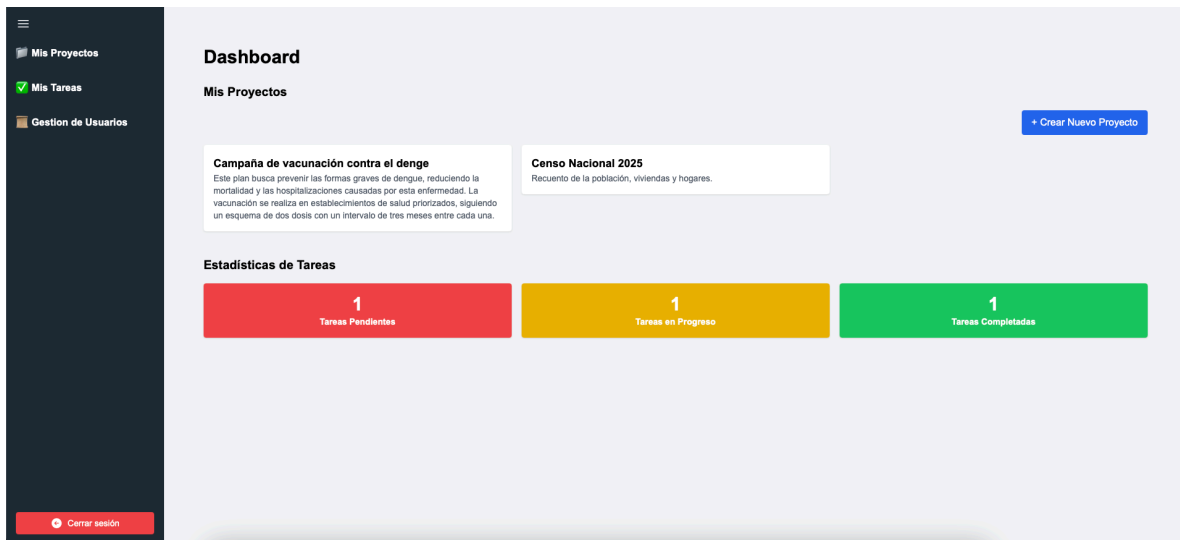
Estructura proyecto
Imagen 2, CodeSnap Visual Studio Code.



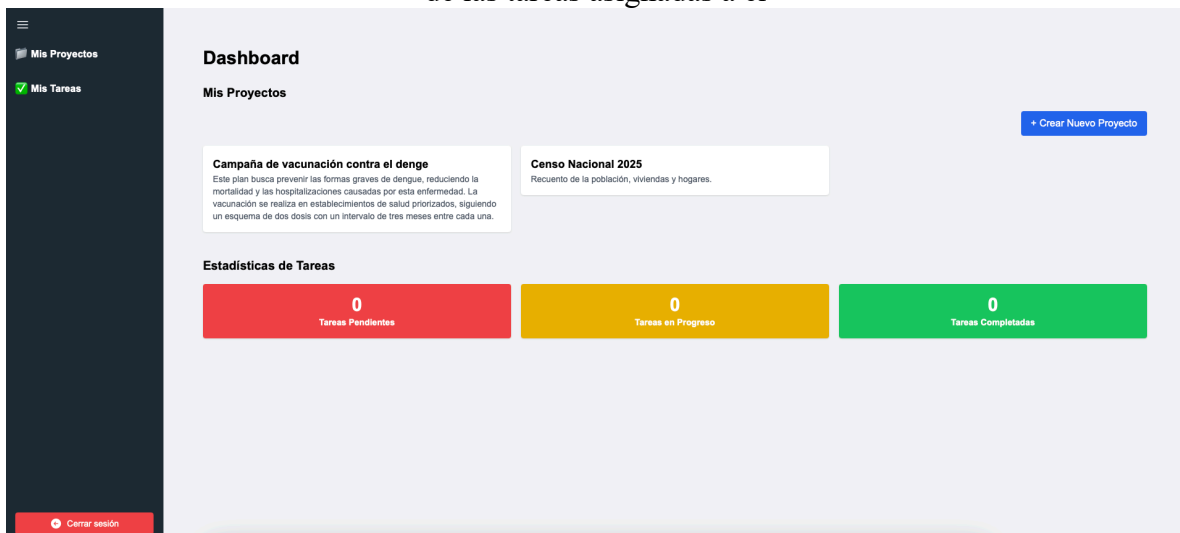
Primer vistazo al entrar en la pagina web, nos encontramos con el login y podemos observar la opcion Regístrate aquí.



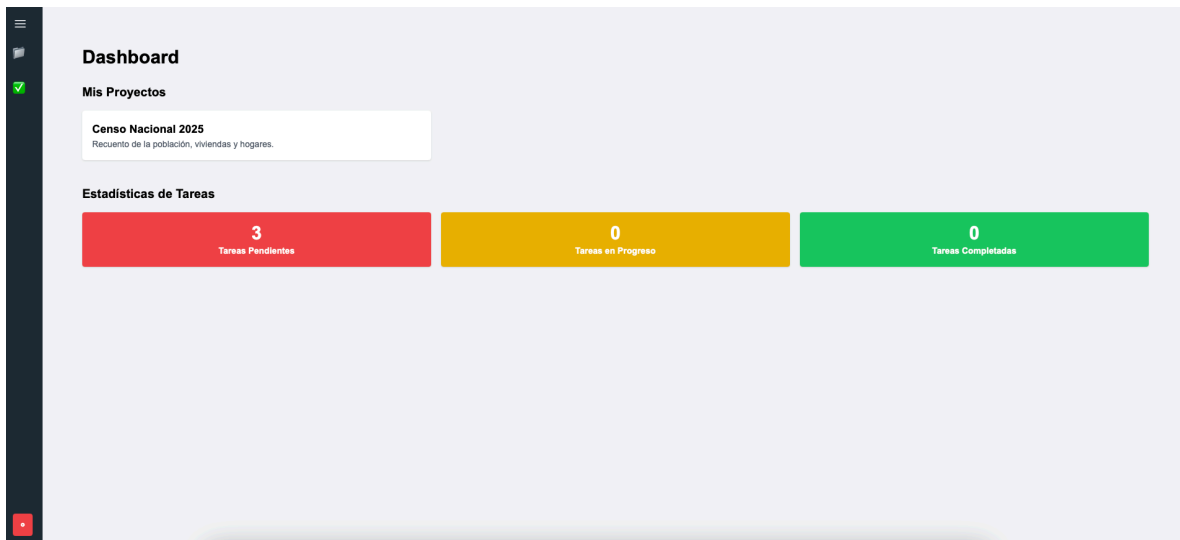
Pantalla de registro de usuario, cualquier usuario puede registrarse al sistema, y desde esta vista si se coloca cualquier otra ruta, no puede acceder ya que existe la proteccion de ellas.



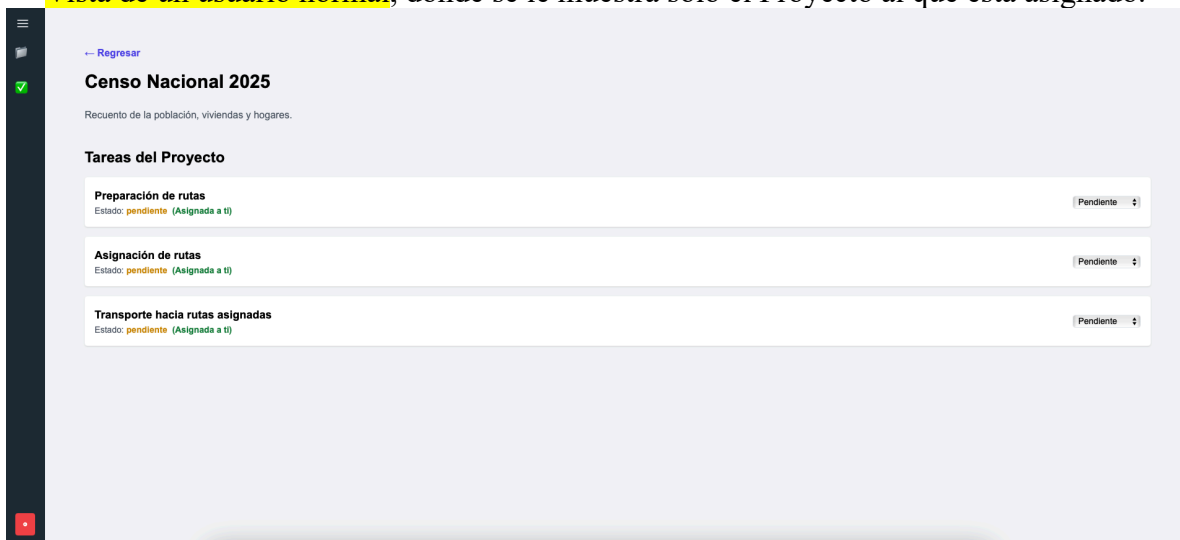
Vista de Administrador, Se puede observar un sidebar con los procesos que puede realizar el administrador, y también en el dashboard podemos ver los proyectos y unas estadísticas de las tareas asignadas a el



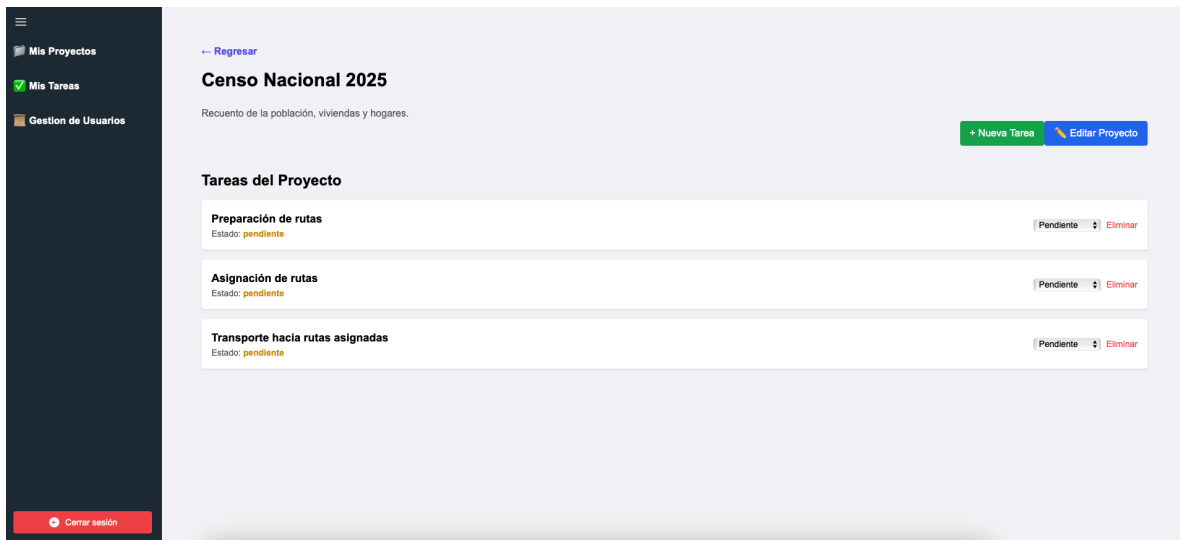
Vista de Gerente, tiene prácticamente la misma vista pero tiene oculta la parte de gestión de usuarios que es lo que diferencia al usuario.



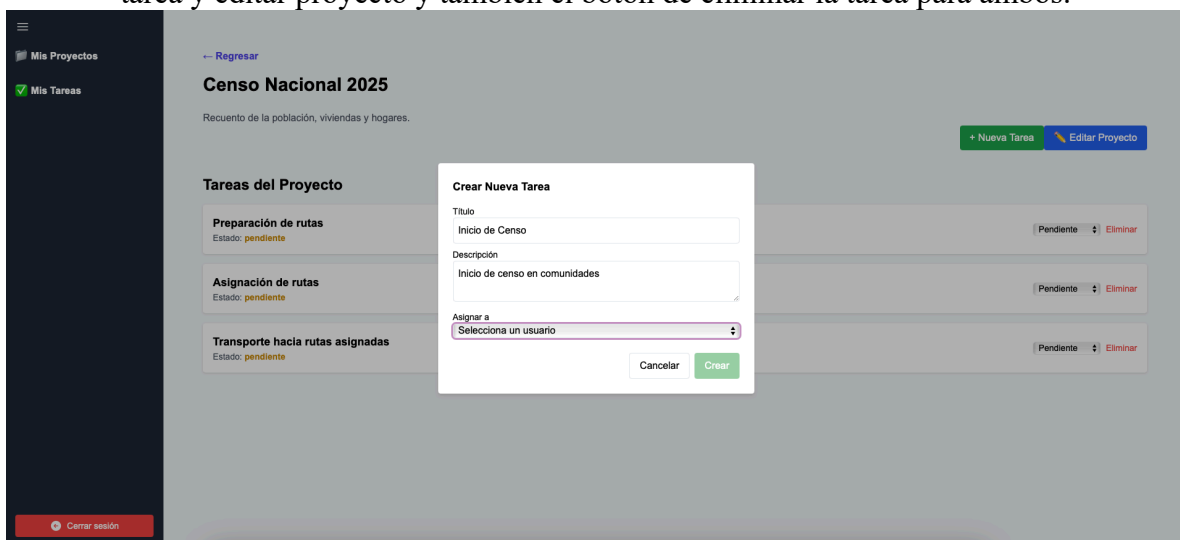
Vista de un usuario normal, donde se le muestra solo el Proyecto al que esta asignado.



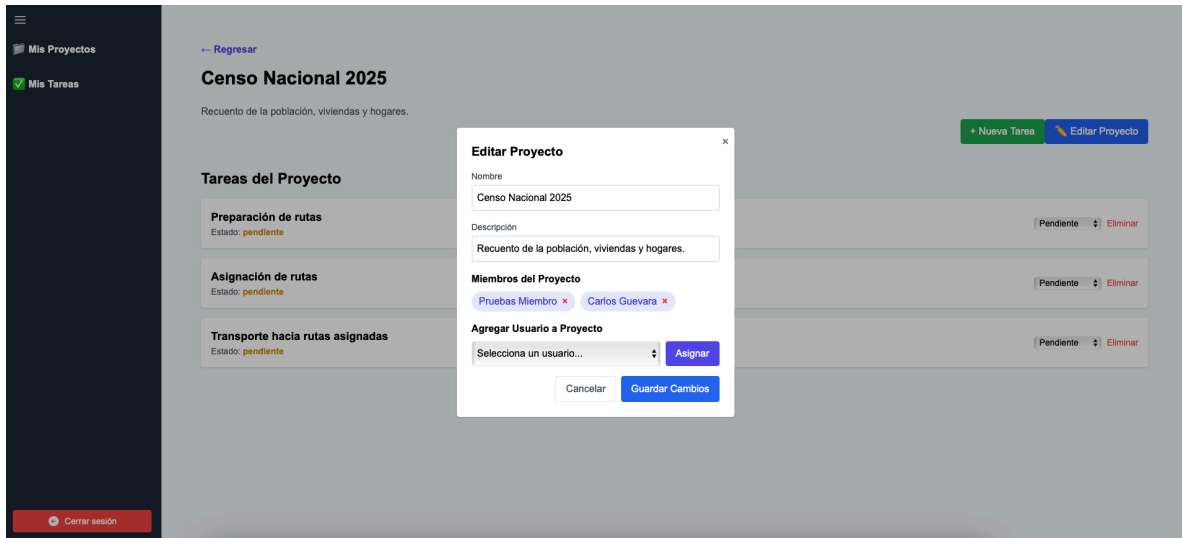
Vista de usuario normal, Al dar click en el proyecto se pueden observar todas las tareas del proyecto en este caso este usuario tiene todas las tareas asignadas y puede cambiar desde ahí el estado de ellas.



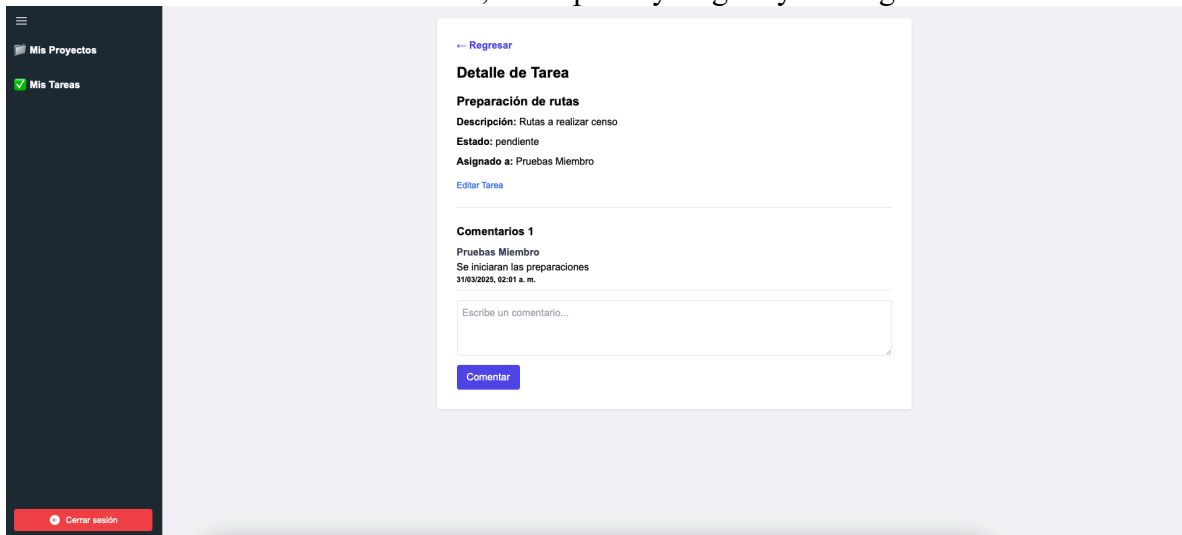
Vista de administrador y Gerente, Esta vista especifica es del administrador lo podemos diferenciar por el sidebar de gestion de usuarios pero practicamente se tienen un par de diferencias significativas a la vista del miembro como se agregan los botones para nueva tarea y editar proyecto y tambien el boton de eliminar la tarea para ambos.



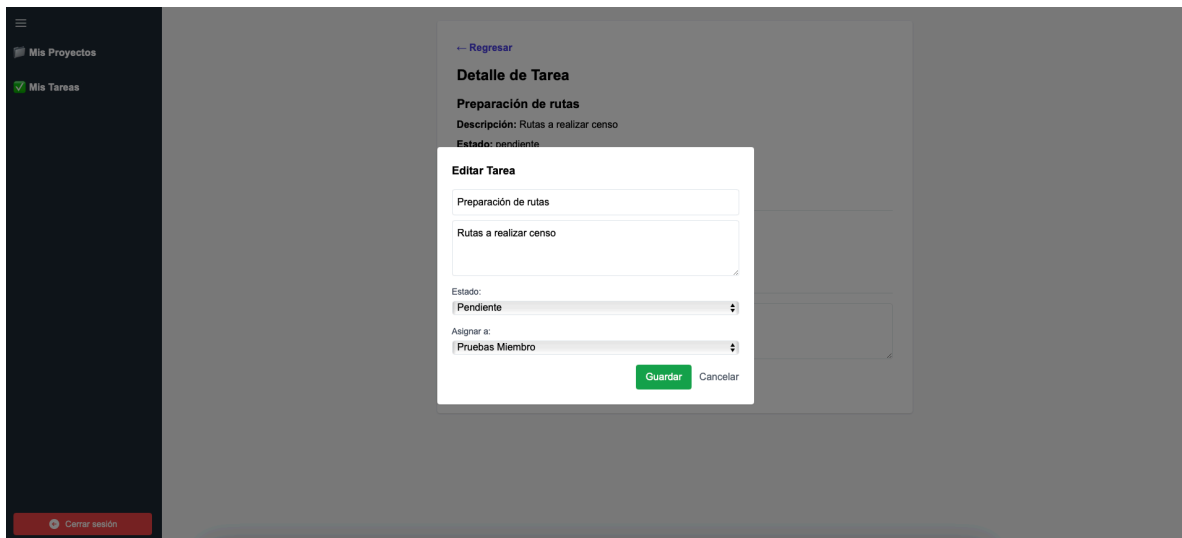
Vista de Gerente y Administrador del modal de creacion de tarea, Podemos observar el modal el cual nos solicita Titulo, Descripción y asignar a donde solo se pueden elegir a las personas que estan asignadas al proyecto.



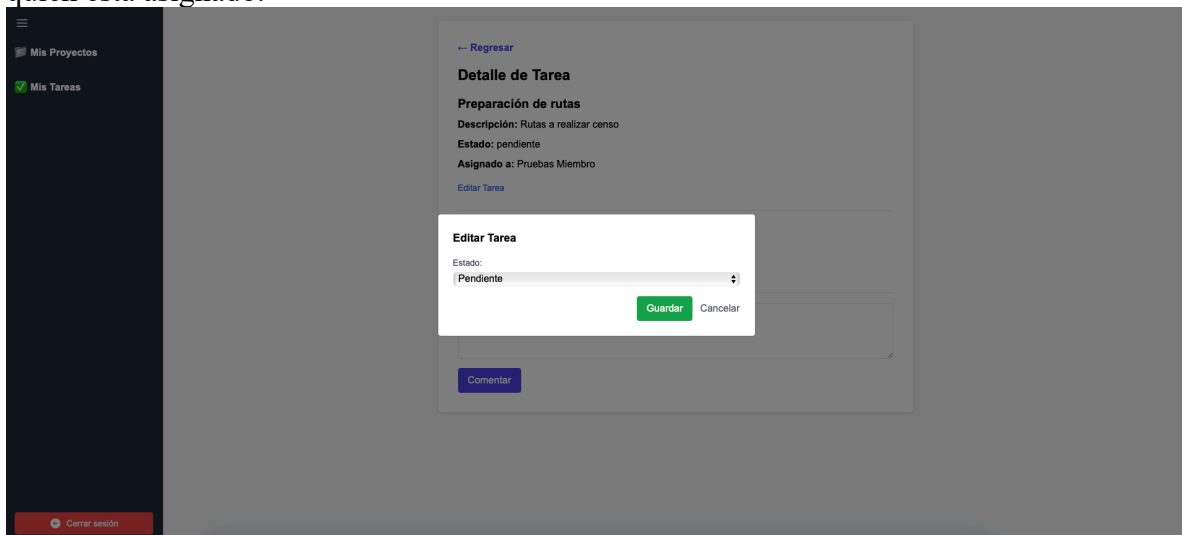
Vista de Gerente y Administrador del modal de Editar Proyecto, Se pueden hacer modificaciones del nombre, descripción y asignar y desasignar usuarios.



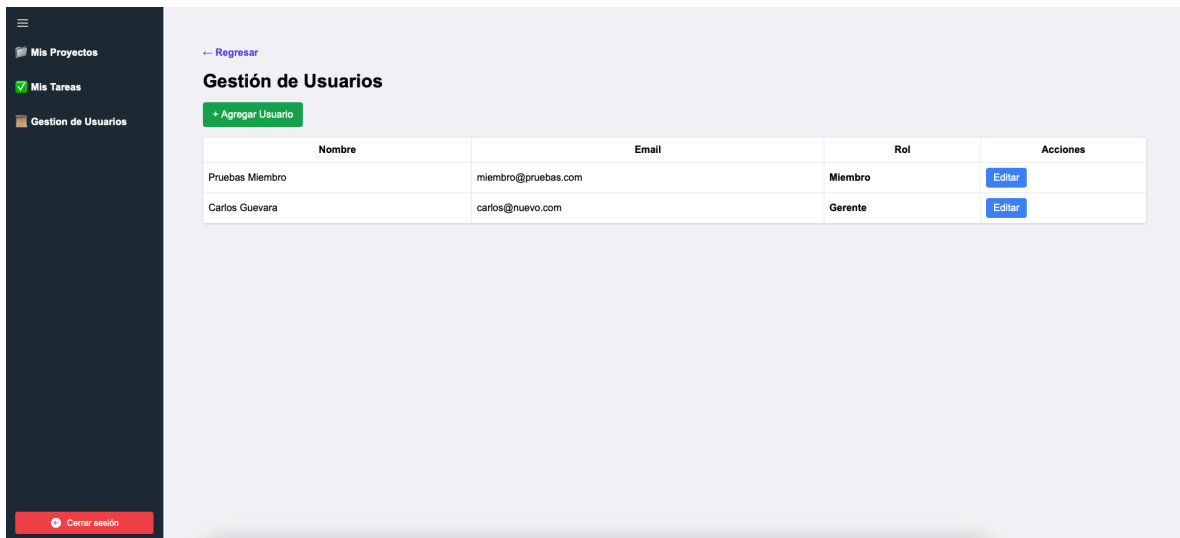
Vista de Gerente y Administrador de la pantalla de detalle de tarea, en este modal podemos ver una tarea que esta asignada al usuario llamado Pruebas Miembro, en el cual el usuario a dejado un comentario.



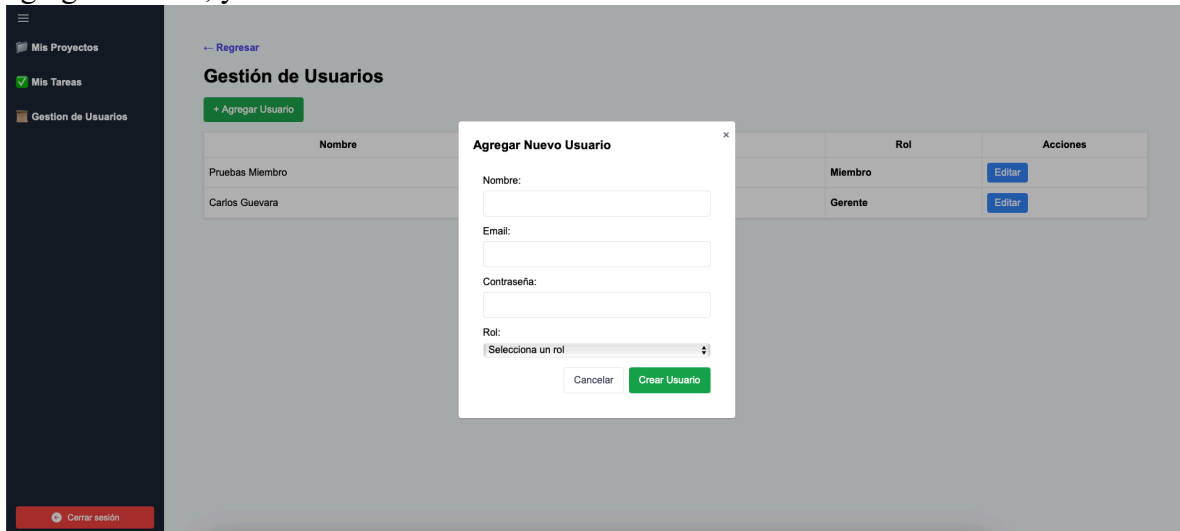
Vista de Gerente y Administrador del modal de editar tarea, en este modal se puede observar que se pueden modificar el título, la descripción, el estado y se puede cambiar a quien esta asignado.



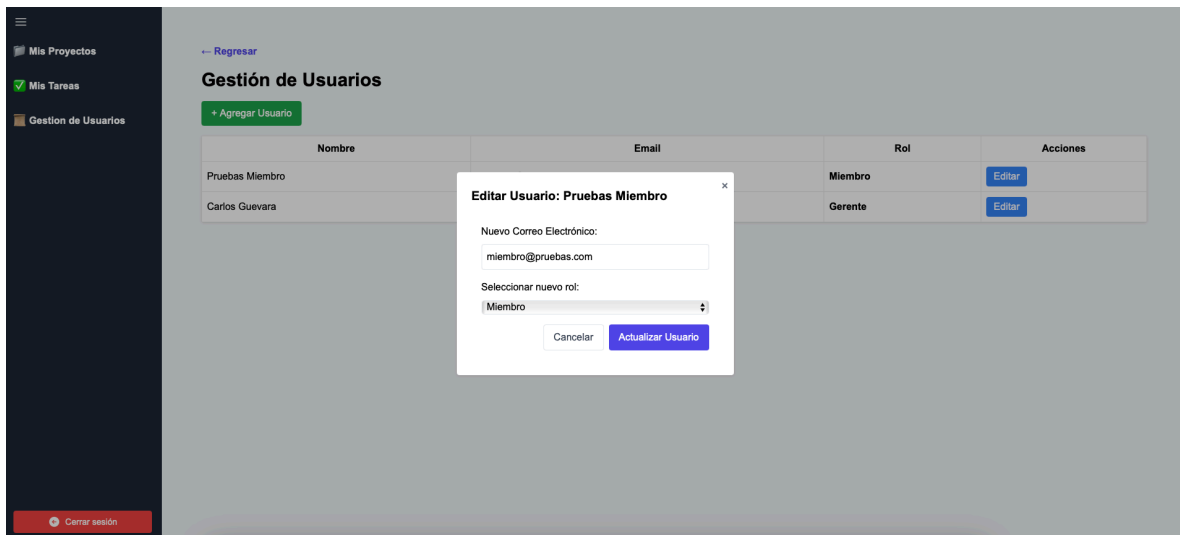
Vista de Miembro del modal de editar tarea, donde el solo puede cambiar el estado de la tarea.



Vista de Administrador, En esta vista tenemos diferentes puntos como lo pueden ser el agregar usuario, y el editar los usuarios existentes.



Vista de Administrador, Modal de agregar nuevo usuario, nos solicita los datos para que el usuario pueda acceder.



Vista de Administrador, Modal de editar usuario, nos solicita los datos para modificar la información del usuario.



Información del archivo `.env.local` instancia de vercel comentada en caso se quieran hacer pruebas locales dejar como esta, si no cambiar a base de datos de supabase.