

ProjectPro - Sistema de Gestión de Proyectos SaaS

ProjectPro es una aplicación web completa desarrollada como Software as a Service (SaaS) para la gestión eficiente de proyectos, tareas, equipos y plazos. Esta aplicación ha sido desarrollada utilizando Python con el framework Flask para el backend y HTML, CSS y JavaScript para el frontend.

Características Principales

- **Sistema de Autenticación:** Registro, inicio de sesión y gestión de sesiones con JWT.
- **Gestión de Proyectos:** Creación, edición y seguimiento de proyectos con fechas, participantes y presupuestos.
- **Jerarquía de Tareas:** Permite crear tareas y subtareas con seguimiento de progreso.
- **Gestión de Equipos:** Asignación de roles específicos a los miembros del equipo.
- **Seguimiento de Hitos:** Marcadores de puntos importantes en la línea de tiempo del proyecto.
- **Dashboard Interactivo:** Visualización de datos relevantes de manera resumida.
- **Diseño Responsivo:** Adaptado a diferentes dispositivos.
- **API RESTful:** Para integración con otros sistemas.

Estructura del Proyecto

```

project-management-saas/
|
├─ backend/
|   │   ├── app.py           # Punto de entrada de la aplicación Flask
|   │   ├── config.py        # Configuraciones de la aplicación
|   │   └─ models/          # Modelos de datos
|   │       ├── __init__.py
|   │       ├── user.py      # Modelo de usuario
|   │       ├── project.py   # Modelo de proyecto
|   │       ├── participant.py # Modelo de participante
|   │       ├── task.py      # Modelo de tarea
|   │       ├── subtask.py   # Modelo de subtarea
|   │       └─ milestone.py  # Modelo de hito
|   └─ routes/              # Rutas y endpoints de la API
|       ├── __init__.py
|       ├── auth.py         # Rutas de autenticación
|       ├── projects.py     # Rutas de proyectos
|       ├── participants.py # Rutas de participantes
|       ├── tasks.py        # Rutas de tareas
|       └─ milestones.py    # Rutas de hitos
|   └─ services/            # Lógica de negocio
|       ├── __init__.py
|       ├── auth_service.py # Servicio de autenticación
|       └─ project_service.py # Servicio de proyectos
|   └─ utils/               # Utilidades
|       ├── __init__.py
|       └─ auth_utils.py    # Utilidades de autenticación
|
├─ frontend/
|   │   └─ static/
|   │       ├── css/
|   │           ├── style.css # Estilos principales
|   │           ├── landing.css # Estilos de la landing page
|   │           ├── login.css # Estilos de la página de login
|   │           └─ dashboard.css # Estilos del dashboard
|   │       └─ js/
|   │           ├── auth.js # JavaScript para autenticación
|   │           ├── projects.js # JavaScript para gestión de proyectos
|   │           ├── tasks.js # JavaScript para gestión de tareas
|   │           ├── milestones.js # JavaScript para gestión de hitos
|   │           └─ timeline.js # JavaScript para gráfico de línea de tiempo

```

```
| | └─ images/           # Imágenes para la aplicación
| └─ templates/
|   └─ index.html       # Landing page
|   └─ login.html       # Página de login
|   └─ register.html    # Página de registro
|   └─ dashboard.html   # Dashboard principal
|   └─ project.html     # Vista detallada de proyecto
|   └─ components/     # Componentes HTML reutilizables
|
└─ requirements.txt     # Dependencias del proyecto
└─ README.md           # Documentación del proyecto
```

Requisitos del Sistema

- Python 3.8 o superior
- MySQL (o SQLite para desarrollo)
- Navegador web moderno

Instalación y Configuración

1. Clonar el repositorio

bash

 Copiar

```
git clone https://github.com/tu-usuario/project-management-saas.git
cd project-management-saas
```

2. Crear y activar entorno virtual

bash

 Copiar

```
python -m venv venv
source venv/bin/activate # En Windows: venv\Scripts\activate
```

3. Instalar dependencias

bash

 Copiar

```
pip install -r requirements.txt
```

4. Configurar variables de entorno

Crea un archivo `.env` en la raíz del proyecto con el siguiente contenido:

 Copiar

```
FLASK_APP=backend/app.py  
FLASK_
```