

Propuesta del Proyecto: Gestor de Tareas

1. Introducción

El presente documento describe la propuesta de desarrollo del **Gestor de Tareas**, una aplicación web que permite a los usuarios gestionar sus tareas de manera eficiente.

El sistema contará con una interfaz intuitiva desarrollada en **Angular**, un backend con **Node.js y Express**, y una base de datos gestionada con **MySQL** a través del ORM **Prisma**.

2. Justificación

En la actualidad, la gestión del tiempo y la organización de tareas son fundamentales tanto en el ámbito personal como profesional.

Existen diversas herramientas en el mercado, pero muchas de ellas son complejas o requieren suscripciones.

Este proyecto busca ofrecer una solución **sencilla, gratuita y de fácil uso** para mejorar la productividad de los usuarios.

3. Objetivos

3.1 Objetivo General

Desarrollar una aplicación web que permita la gestión de tareas de manera sencilla, incorporando autenticación de usuarios y control de roles.

3.2 Objetivos Específicos

- Implementar un **frontend en Angular** con una interfaz moderna y fácil de usar.
- Desarrollar un **backend en Node.js con Express** para gestionar la lógica del negocio.
- Utilizar **Prisma** como ORM para interactuar con una base de datos **MySQL**.
- Implementar un **sistema de autenticación y autorización** de usuarios con diferentes roles (usuario y administrador).
- Proteger las rutas para evitar accesos no autorizados.
- Garantizar la seguridad de contraseñas y datos sensibles.

4. Alcance

4.1 Funcionalidades

Gestión de Tareas

- Crear nuevas tareas (solo administradores), incluyendo: título, descripción, fecha de inicio, fecha de fin, prioridad (alta, media, baja) y usuario asignado.
- Cada tarea estará asignada a un solo usuario.
- Marcar tareas como completadas.
- Eliminar tareas (solo administradores).
- Listar todas las tareas en una interfaz ordenada.

Gestión de Usuarios y Autenticación

- Registro de usuarios con credenciales seguras.
- Inicio de sesión y gestión de sesiones.
- Control de acceso a las tareas según el rol del usuario.
- Visualización de tareas propias para los usuarios estándar.
- Gestión de usuarios y roles desde un panel de administrador.

Interfaz de Usuario

- Interfaz responsiva utilizando **Angular Material**.
- Experiencia de usuario optimizada y moderna.

Seguridad

- Autenticación mediante **JSON Web Tokens (JWT)**.
- Cifrado de contraseñas con **Bcrypt**.
- Confirmación doble de contraseña en registro.

4.2 Exclusiones

- No se implementarán **subtareas** ni **notificaciones automáticas** en esta primera versión.
- No se contemplan integraciones con aplicaciones externas (ejemplo: Google Calendar).
- No se incluirán estadísticas avanzadas ni reportes gráficos en la primera entrega.

5. Tecnologías Utilizadas

- **Frontend:** Angular, Angular Material
- **Backend:** Node.js, Express
- **Base de Datos:** MySQL
- **ORM:** Prisma
- **Autenticación y Seguridad:** JWT, Bcrypt
- **Otros:** TypeScript, JSON para intercambio de datos
- **Despliegue:** Vercel (frontend), Render (backend), Clever Cloud (base de datos)

6. Plan de Trabajo

6.1 Etapas

1. Análisis y diseño

- Definir la estructura del sistema.
- Diseñar la interfaz de usuario.
- Modelar la base de datos con Prisma.

2. Desarrollo del Backend

- Configurar el servidor con Express.
- Implementar las rutas de la API REST.
- Implementar autenticación y autorización con JWT.

- Conectar la base de datos MySQL con Prisma.

3. Desarrollo del Frontend

- Implementar la interfaz en Angular.
- Consumir la API desde Angular.
- Integrar Angular Material para mejorar la experiencia de usuario.
- Implementar el sistema de autenticación en el frontend.

4. Pruebas y despliegue

- Realizar pruebas unitarias y de integración.
- Desplegar en un servidor web.

7. Resultados Esperados

- Aplicación web funcional y estable.
- Implementación de una **API REST** con operaciones CRUD protegidas por roles.
- Base de datos optimizada y conectada mediante Prisma.
- Sistema de autenticación y autorización seguro.
- Interfaz intuitiva y responsiva con Angular Material.

8. Conclusión

Este proyecto permitirá a los usuarios gestionar sus tareas de manera eficiente y sencilla, diferenciando claramente los roles de usuario y administrador.

Se busca proporcionar una herramienta de **código abierto**, ampliable en futuras versiones con nuevas funcionalidades, como:

- Estadísticas avanzadas.

- Gestión colaborativa de tareas.
- Notificaciones automáticas vía mail o WhatsApp.