

Propuesta del Proyecto: Gestor de Tareas Simple

1. Introducción

El presente documento describe la propuesta de desarrollo del **Gestor de Tareas Simple**, 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, permitiendo crear, visualizar y eliminar tareas.

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**.
- Permitir la eliminación de tareas.
- Implementar un sistema de **autenticación de usuarios** para gestionar sesiones y acceso.
- Proteger las rutas para los accesos sin autorización.

4. Alcance

4.1 Funcionalidades

- **Gestión de tareas:**
 - Crear nuevas tareas con título, descripción, fecha inicio y fecha fin, asignarle una prioridad (alta, media o baja).
 - Cada tarea tiene asignado un solo usuario.
 - Marcar tareas como completadas.
 - Eliminar tareas existentes.
 - Listar todas las tareas en una interfaz ordenada.
- **Autenticación de usuarios:**
 - Registro de usuarios con credenciales seguras.
 - Inicio de sesión y gestión de sesiones.

- Control de acceso a tareas según el usuario autenticado.
- **Interfaz de usuario responsiva** con Angular Material.
- **Persistencia de datos** en una base de datos MySQL.

4.2 Exclusiones

- No se ofrecerán funcionalidades avanzadas como subtareas o notificaciones en la primera versión.
- No se ofrecera la funcionalidad de “Estadísticas” para esta primera versión.

5. Tecnologías Utilizadas

- **Frontend:** Angular, Angular Material
- **Backend:** Node.js, Express
- **Base de Datos:** MySQL
- **ORM:** Prisma
- **Autenticación:** JSON Web Tokens (JWT), Bcrypt para el cifrado de contraseñas
- **Otros:** TypeScript, JSON para intercambio 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.
 - Implementar autenticación y gestión de sesiones con JWT.
 - Conectar la base de datos PostgreSQL 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 (Heroku, Vercel o similar).

7. Resultados Esperados

- Aplicación web funcional.
- Implementación de una API REST con operaciones CRUD.
- Base de datos optimizada y conectada mediante Prisma.
- Sistema de autenticación seguro y funcional.
- Interfaz intuitiva con Angular Material.

8. Conclusión

Este proyecto permitirá a los usuarios gestionar sus tareas de manera eficiente y sencilla. Se busca proporcionar una herramienta de código abierto que pueda ser utilizada y ampliada en futuras versiones, incorporando nuevas funcionalidades como gestión colaborativa de tareas y notificaciones automáticas via mail/whatsapp.