

Memoria del Proyecto

Gestor de Tareas Básico

Introducción

Este proyecto consiste en el desarrollo de un gestor de tareas básico utilizando **Node.js** como tecnología principal. El objetivo principal ha sido implementar una API REST con operaciones CRUD (Crear, Leer, Actualizar y Eliminar), gestionando las tareas a través de un archivo JSON.

He querido centrarme en demostrar mi capacidad para aplicar los conocimientos adquiridos en este módulo y sentar una base sólida para futuras mejoras. El plan es ampliarlo en el futuro integrando otras tecnologías con lo que aprenderemos en bases de datos, Express, React y AWS.

Desarrollo

Estructura del Proyecto

El proyecto se ha estructurado de forma modular para facilitar la extensión y el mantenimiento:

- `server.js` : Contiene la lógica del servidor y la API REST.
- `tasks.json` : Archivo utilizado para almacenar las tareas.
- `public/` : Carpeta con la interfaz web.

Durante la configuración inicial, utilicé herramientas como `nodemon` para reiniciar automáticamente el servidor durante el desarrollo, lo que facilitó la iteración y las pruebas.

Implementación de la API REST

La API REST se ha diseñado con las rutas clave:

- **GET /tasks**: Devuelve las tareas existentes o un array vacío si el archivo está vacío.
- **POST /tasks**: Permite agregar tareas con validación del título y asigna un `id` único.
- **PUT /tasks/:id**: Actualiza tareas por `id`, verificando su existencia.
- **DELETE /tasks/:id**: Elimina tareas específicas basándose en su `id`.

Las funcionalidades se han comprobado manualmente utilizando `curl` para garantizar su correcto funcionamiento.

Interfaz Web

He desarrollado una interfaz web interactiva con HTML, CSS y JavaScript. Algunas de sus características son:

- Agregar, editar y eliminar tareas.
 - Marcar tareas como completadas o pendientes.
 - Notificaciones en tiempo real usando la librería **Toastify**.
 - Validaciones para evitar datos inválidos desde el frontend.
-

Problemas Encontrados y Soluciones Implementadas

1. Error al localizar `tasks.json`

- **Problema:** El servidor no encontraba el archivo al iniciarse desde un directorio diferente.
- **Solución:** He utilizado `path.join(__dirname, 'tasks.json')` para asegurar rutas absolutas.

2. Conflictos entre rutas estáticas y API

- **Problema:** Las solicitudes a la API se confundían con peticiones de archivos estáticos.
- **Solución:** He ajustado el orden de las rutas, dando prioridad a las rutas de la API.

3. Falta de validaciones en las operaciones CRUD

- **Problema:** Datos inválidos causaban errores.
- **Solución:** He implementado validaciones tanto en el servidor como en el cliente.

4. Manejo de errores en operaciones de lectura/escritura

- **Problema:** Dificultad para depurar la causa de los errores.
 - **Solución:** He añadido `try-catch` y mensajes descriptivos.
-

Resultados Finales

El proyecto actual cumple con los objetivos del módulo:

- Funcionalidades CRUD implementadas y probadas.
 - Interfaz web funcional y conectada a la API.
 - Problemas resueltos mediante estrategias claras y efectivas.
-

Mejoras Futuras

Planeo convertir este proyecto en una aplicación más completa y escalable:

1. Base de Datos:

- Migrar de un archivo JSON a una base de datos como PostgreSQL.

2. Express:

- Simplificar el manejo de rutas y middlewares con Express.

3. Frontend en React:

- Rediseñar la interfaz web para un diseño modular y dinámico con React

4. Implementación en la Nube:

- Desplegar la aplicación en AWS.

5. Sistema de Usuarios:

- Añadir autenticación para permitir a cada usuario gestionar sus propias tareas.
-

Conclusión

El desarrollo de este proyecto tenía dos propósitos: demostrar lo aprendido durante el módulo y servir de base para el proyecto final. Aunque el gestor de tareas es funcional, lo considero un punto de partida para futuras mejoras y un ejemplo práctico de mis habilidades. Mi intención es seguir ampliándolo a medida que aprenda nuevas tecnologías, convirtiéndolo en un proyecto sólido y escalable para mi portafolio. Por ello, toda crítica y aportación es bienvenida! Muchísimas gracias por todo!

Nota : En la ideación y desarrollo de este proyecto, se ha recurrido al apoyo de la inteligencia artificial para generar ideas, estructurar soluciones y optimizar procesos técnicos. El uso de esta herramienta ha facilitado la exploración de enfoques alternativos y ha contribuido a una implementación más clara y eficiente de las funcionalidades, asegurando el cumplimiento de los objetivos planteados. No obstante, se han tenido en cuenta sus limitaciones para garantizar un equilibrio entre su uso y el desarrollo crítico y personal del proyecto.