

## PRUEBA TÉCNICA APRENDIZ SENA BACKEND DEVELOPER

### Prueba Técnica para Desarrollador Backend (Node.js + Express + TypeScript)

Proyecto: Sistema de Gestión de Tareas

#### Descripción:

Desarrollar una API para la gestión de tareas con CRUD (Crear, Leer, Actualizar, Eliminar) y un sistema de autenticación y autorización, garantizando que los usuarios gestionen únicamente sus propias tareas. La API debe ser consumible desde Postman.

#### Tecnologías:

- Node.js
- Express.js
- TypeScript
- MongoDB (u otra base de datos de elección)

#### Funcionalidades Requeridas:

- 1. Autenticación y Autorización:**
  - Registro e inicio de sesión de usuarios.
  - Middleware para proteger rutas autenticadas.
- 2. Gestión de Tareas:**
  - Crear, leer, actualizar y eliminar tareas.
  - Cada tarea debe tener título, descripción, fecha de vencimiento y estado.
- 3. Validación y Manejo de Errores:**
  - Validar entradas del usuario.
  - Manejar errores y enviar respuestas apropiadas.
- 4. Documentación:**
  - Documentar el API con Swagger (o similar).
  - Instrucciones para ejecutar el proyecto localmente.
  - Comentarios JSDoc detallados en el código.
- 5. Buenas Prácticas:**
  - Código limpio y estructurado siguiendo las mejores prácticas de TypeScript.
  - Uso adecuado de Git con commits descriptivos y regulares.

#### Entregables:

- 1. Código Fuente:**

Bogotá calle 92 #15 - 78

<https://instaleap.io/>

- Repositorio en GitHub con instrucciones claras de instalación y ejecución.
- Commits recurrentes durante el desarrollo.
- Si se usan asistentes de IA, incluir los prompts utilizados.

## 2. Documentación:

- API documentada con Swagger.
- README detallado con instrucciones de configuración, ejecución y descripción de la arquitectura.
- Comentarios JSDoc detallados en el código.

## 3. Video Explicativo:

- Enviar un video explicando el proyecto, su arquitectura y cómo utilizarlo desde Postman.

### Deseables (pero no requeridos):

#### 1. Despliegue:

- Desplegar la aplicación en un servicio online (Heroku, Vercel, AWS, etc.).

#### 2. Pruebas:

- Incluir pruebas unitarias y de integración.

### Evaluación:

1. Funcionalidad y cumplimiento de requisitos.
2. Calidad del código y uso de TypeScript.
3. Claridad y calidad de la documentación.
4. Uso adecuado de Git y control de versiones.
5. Claridad y profundidad del video explicativo.

### Plazo de Entrega:

5 días

### Instrucciones Finales:

Enviar el enlace al repositorio de GitHub y el video explicativo. Asegúrate de que el repositorio sea público o comparte el acceso necesario para su revisión.