Buenos Aires aprende
Agencia de Habilidades para el Futuro

NODE JS

Enunciado del Proyecto Final Integrador

Descripción General:

El objetivo del proyecto es diseñar, desarrollar y desplegar una API RESTful funcional que permita gestionar los productos de una tienda en línea o E-Commerce. Este sistema debe permitir a los usuarios autorizados a realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los productos de la tienda y tener la capacidad de almacenar los datos tanto de manera local (JSON) como en la nube (Firebase/Firestore). El desarrollo de este proyecto te permitirá implementar todos los conocimientos adquiridos a lo largo de la cursada y sin dudas plantará algunas semillas para que sigas investigando y aprendiendo sobre el mundo backend y su enorme cantidad de alternativas y aplicaciones en el mundo del desarrollo web.

El proyecto consistirá en:

1. Construir un servidor web utilizando **Node.js** y **Express.js**.
2. Implementar una estructura de proyecto modular y organizada basada en los principios de arquitectura RESTFul.
3. Integrar el manejo de datos mediante archivos JSON y, posteriormente, conectarlo con un servicio de base de datos en la nube (Firestore).
4. Asegurar la correcta comunicación entre cliente y servidor mediante métodos y códigos HTTP.
5. Crear y configurar capas lógicas de seguridad y autenticación.
6. Desplegar el proyecto en un entorno de producción funcional, accesible mediante una URL pública.

Requerimientos Específicos:

1. **Estructura del Proyecto**
    - Crear una estructura clara con las siguientes carpetas principales:
    - /controllers: Contendrá la lógica de negocio.

- /models: Definirá la estructura de los datos.
- /routes: Definirá las rutas de acceso a la API.
- /services: Gestionará el acceso a datos y la interacción con la base de datos.
- /public: Archivos estáticos (opcional).

2. **Funcionalidades**
   - Gestión de productos:
     - Crear nuevos productos para la tienda en línea.
     - Listar, filtrar, buscar y cualquier acción de consulta sobre el listado de productos.
     - Actualizar parcial o totalmente un producto.
     - Eliminar productos mediante id.
   - Manejo de errores: Implementar controladores de errores para manejar respuestas adecuadas en caso de solicitudes inválidas o fallas del servidor.
   - CORS: Configuración para permitir solicitudes desde diferentes dominios.

3. **Seguridad**
   - Autenticación y autorización: validar el acceso de los usuarios registrados mediante el uso de JWT y tokens.

4. **Base de Datos**
   - Acceso inicial a los datos utilizando un archivo JSON.
   - Migrar el manejo de datos a Firebase/Firestore, implementando un servicio para consumir esta base de datos.

5. **Despliegue**
   - Subir la API a un servicio de producción como Vercel, Railway, entre otros.

Funcionalidad Esperada:

- La API debe responder correctamente a los métodos HTTP (`GET`, `POST`, `PUT`, `PATCH`, `DELETE`).
- La API debe devolver los productos o el producto seleccionado.
- Las rutas definidas deben ser claras y tener una responsabilidad única.
- El sistema debe manejar errores comunes (404, 500) y devolver mensajes claros y descriptivos.
- Los datos deben almacenarse y recuperarse correctamente desde la base de datos local (JSON) y la nube (Firestore).
- La API debe permitir el uso de la herramienta solo a usuarios autorizados y autenticados.

Entrega del Proyecto:

- Formato de Entrega: El proyecto debe ser subido a un repositorio de GitHub. Se deberá compartir el enlace del repositorio en el aula virtual antes de la fecha límite.
- Nombre del Repositorio: `proyecto-final-ecommerce-[nombre-apellido]`
- Documentación: El archivo `README.md` debe estar incluido en el repositorio, explicando claramente el objetivo del proyecto, las tecnologías utilizadas, cómo configurarlo y cualquier otro detalle relevante.

Evaluación:

1. **Organización del Código**:
   - Estructura modular y limpieza del código.
   - Uso adecuado de las capas de la aplicación: controladores, modelos, rutas y servicios.
2. **Funcionalidad**:
   - Cumplimiento de los requerimientos funcionales.
   - Correcto manejo de datos en JSON y Firestore.
3. **Manejo de Errores**:
   - Implementación de controladores de errores y códigos HTTP adecuados.
4. **Documentación**:
   - Claridad en la descripción y alcance del proyecto y cómo ejecutarlo en el README principal del repositorio remoto.

5. **Despliegue**:
    - La API debe estar accesible mediante una URL pública.

Notas Adicionales:

- **Originalidad**: Se espera que el proyecto final sea un trabajo original del estudiante. La copia o el uso de proyectos preexistentes será motivo de descalificación.
- **Presentación**: Los estudiantes deben presentar su proyecto, explicando su arquitectura, decisiones técnicas y desafíos enfrentados.
- **Soporte**: Durante el desarrollo del proyecto, los estudiantes podrán hacer preguntas durante las clases destinadas a la resolución de dudas, donde se les proporcionará asistencia técnica y orientación.