

Ejercicio de Evaluación - Fullstack



"Facilitando superpoderes IA a la fuerza de trabajo humana"

© Cex 2024

Título	Ejercicio de Evaluación
Proyecto	ANY.AI
Versión	1.0.0
Fecha de creación	Jun 24, 2024
Creador	@Daniel Espinosa Rodriguez
Revisores	@Chema Alvarez @Nayelli Colin
Fecha revisión	Jun 24, 2024

Descripción

Desarrolla una aplicación que permita a los usuarios subir documentos, extraer y almacenar metadatos, y visualizar estos metadatos a través de una interfaz web. La aplicación se divide en dos partes:

1. **Backend:** API en Node.js para manejar la carga de documentos y almacenamiento de metadatos.
2. **Frontend:** Componente en React para interactuar con la API y mostrar los metadatos.

Requerimientos

- **Editor de Código:** Un editor de código moderno como Visual Studio Code, WebStorm, Sublime Text.
- **Terminal:** Acceso a una terminal o línea de comandos para ejecutar comandos de desarrollo (integrada en el editor o independiente).
- **Node.js y NPM:** Debe tener instalado Node.js (versión recomendada: 18.x o superior) y NPM (incluido con Node.js).
- **Postman o Insomnia:** Para probar los endpoints de la API y verificar que funcionan correctamente.
- **Git:** Para el control de versiones y gestión del código fuente.
- **Acceso a Object Storage:** Se proporcionará acceso a un API de Object Storage durante la sesión.
- **Acceso a MongoDB:** Se proporcionará acceso a una base de datos MongoDB durante la sesión de entrevista.

Parte 1: Backend

Requerimientos:

1. **Endpoint para Subir Documentos:** Desarrolla un endpoint que permita subir documentos en formatos PDF, Excel y Word.
 - **Almacenamiento:** Almacena los documentos en una carpeta local. (**EXTRA:** Implementar almacenamiento en el Object Storage proporcionado).
 - **Extracción de Metadatos:** Extrae metadatos como nombre del archivo, tamaño, tipo, fecha de carga, y cualquier otro dato relevante.
 - **Almacenamiento de Metadatos:** Almacena los metadatos para su posterior consulta. (**EXTRA:** Utilizar la base de datos MongoDB proporcionada).
2. **Endpoint para Consultar Metadatos:** Desarrolla un endpoint que permita consultar los metadatos almacenados.

- **Filtros:** Permite aplicar filtros básicos como nombre de archivo y tamaño.
3. **(EXTRA) Endpoint para Descargar Archivos:** Desarrolla un endpoint que permita descargar archivos utilizando su ID o nombre.
- **Manejo de Errores:** Maneja casos en los que el archivo no se encuentra.

Parte 2: Frontend

Requerimientos:

1. **Formulario de Subida de Documentos:** Desarrolla un formulario en React que permita a los usuarios seleccionar y subir documentos.
 - **Interacción con la API:** El formulario debe enviar el archivo al endpoint creado en Parte 1 del ejercicio.
2. **Visualización de Metadatos:** Desarrolla una sección en React que muestre en una lista o tabla, los documentos junto con sus metadatos almacenados.
 - **Interacción con la API:** Obtén los metadatos desde el endpoint Parte 1 del ejercicio.
 - **Filtros:** Permite aplicar filtros básicos como nombre de archivo y tamaño.
3. **(EXTRA) Descarga de Archivos:** Añade un botón o enlace que permita descargar archivos utilizando su ID o nombre.
 - **Interacción con la API:** Utiliza el endpoint generado en Parte 1 del ejercicio.

Puntos Extra

1. Integración con Object Storage
2. Integración con MongoDB
3. Implementación del Endpoint de Descarga

Instrucciones de Entrega

Repositorio en GitHub

- **Crea un repositorio en GitHub:** Inicializa un nuevo repositorio en GitHub y sube tu código fuente allí.
- **Sube tu código:** Asegúrate de que el código esté completo y funcional. Incluye tanto el código del backend como del frontend.
- **(OPCIONAL) README.md:** Incluye un archivo `README.md` con instrucciones claras sobre cómo ejecutar el backend y el frontend. Explica cómo configurar cualquier variable de entorno necesaria y cómo probar la aplicación.

Instrucciones de Ejecución

- **Backend:** Explica cómo iniciar el servidor backend, incluyendo comandos para instalar dependencias y ejecutar el servidor.
- **Frontend:** Explica cómo iniciar el servidor frontend, incluyendo comandos para instalar dependencias y ejecutar el frontend.

Demostración

- **Verificación de Funcionalidad:** Asegúrate de que la aplicación cumpla con los requerimientos especificados. Verifica que se pueda cargar documentos, visualizar metadatos con filtros básicos y descargar archivos.

Enlace al Repositorio

- **Proporciona el enlace al repositorio:** Una vez que hayas completado el ejercicio, comparte el enlace al repositorio en GitHub con el equipo de evaluación.

Criterios de Evaluación

- **Funcionalidad:** Verifica que la API y el frontend cumplan con los requerimientos especificados.
- **Calidad del Código:** Evalúa la claridad, organización y buenas prácticas del código.
- **Manejo de Errores:** Observa cómo manejan errores y validaciones en ambas partes (backend y frontend).
- **Puntos Adicionales:** Considera la integración con Object Storage, MongoDB, y la implementación del endpoint de descarga como puntos adicionales.

