

Evaluación: Sistema de Archivos

Entrega: 28 de Noviembre de 2024

Se necesita desarrollar un sistema de gestión de documentos para una empresa. Esta empresa necesita que los usuarios puedan almacenar archivos de manera estructurada y segura, permitiendo que sus documentos sean accesibles mediante una URL.

SERVICIOS A CONSTRUIR

1. Cargar Documentos:

Endpoint: **POST** `/documentos/:rut_usuario`

Permitir que se puedan cargar uno o más documentos en la misma transacción a un usuario. Los archivos subidos deben ser procesados asignándoles un UUID (versión 4) antes de almacenarlos (ese será el nuevo nombre, dejando la misma extensión). Si el directorio donde se va a almacenar el archivo no existe se debe crear automáticamente. El directorio donde deben organizarse los archivos debe tener la siguiente estructura de carpetas basada en la fecha y hora de subida:

`(YYYY/MM/DD/HH/mm/UUID.ext)`. **Ejemplos:**

`2024/11/14/18/30/210d0f1f-4a29-4617-a95b-c4e2aec93f1e.pdf`

`2024/10/20/08/59/8ebd9ecb-51ab-4557-b2c8-b3aaec20c2a7.jpg`

`2025/01/08/12/01/89a6e090-75a4-4dd1-9b5f-72246cca3cde.pdf`

Luego de almacenar el archivo, se debe guardar en base de datos la siguiente información (por cada archivo):

- Rut del Usuario.
- Nombre original del archivo.
- Nombre asignado (UUID).
- Ruta de acceso al archivo en el servidor.
- Fecha y hora de carga.

2. Consultar Documentos:

Endpoint: **GET** **/documentos/:rut_usuario**

Dado el rut de un usuario se debe mostrar la información de todos los archivos que tiene asignados. La ruta de acceso al archivo debe permitir que al completarla con host y puerto, se pueda consultar el documento subido.

3. Eliminar Documentos:

Endpoint: **DELETE** **/documentos/:uuid_archivo**

Se debe eliminar el registro de la base de datos, y con el UUID eliminar también el archivo de la ruta donde se encuentra. En caso de que algún directorio quede sin archivos se debe eliminar para evitar directorios vacíos.

SERVIDOR DE ESTÁTICOS

Se debe configurar un servidor de estáticos para permitir el acceso a los archivos mediante una URL. Este servidor debe publicar todos los archivos estáticos que se hayan subido.

TAREAS A REALIZAR

1. Crear base de datos (Elegir la que quieran: MySQL o MongoDB).
Dejar las instrucciones y/o scripts en la carpeta **scripts_db**.
2. Crear un proyecto NestJS, configurar el Swagger y conectarlo a la BD (crear el docker-compose.yml para ambiente de desarrollo).
3. Implementar los 3 servicios solicitados.
4. Publicar los archivos con el servidor de estáticos.
5. Agregar validaciones a los servicios para evitar errores.

ENTREGABLES

Crear un repositorio en github. Este repositorio debe ser privado con permisos de acceso a jorgemparrah.

En el repositorio se debe subir el proyecto con lo solicitado anteriormente.

En u-cursos, se debe entregar el link al repositorio git y el id del commit donde están los cambios que se evaluarán.