Luis Castelblanco 201910966

Valeria Caro Ramirez 202111040

Nicolas Saavedra Gonzalez 202112963

Documentación del API

1. Introducción

la API desarrollada para el proyecto SaaS RAG. La aplicación permite la gestión, análisis y resumen automatizado de documentos mediante inteligencia artificial generativa. La API está construida con FastAPI y sigue una arquitectura modular, lo que facilita su escalabilidad y mantenimiento.

2. Historias de Usuario

Historia 1: Subir documentos

Como usuario autenticado, quiero subir documentos a la plataforma, para que puedan ser analizados por el modelo de IA.

Criterios de aceptación:

- El usuario debe estar autenticado.
- El sistema debe aceptar archivos en formatos PDF, TXT y DOCX.
- Se debe recibir una confirmación tras la carga exitosa.

Historia 2: Consultar documentos

Como usuario autenticado, quiero listar los documentos que he subido, para acceder a ellos fácilmente.

Criterios de aceptación:

- Solo los documentos del usuario autenticado deben mostrarse.
- La respuesta debe incluir detalles como el nombre, tipo y fecha de carga.

Historia 3: Eliminar documentos

Como usuario autenticado, quiero eliminar un documento que he subido, para gestionar mi espacio de almacenamiento y eliminar información obsoleta.

Criterios de aceptación:

- El usuario debe estar autenticado.
- Solo se pueden eliminar documentos propios.
- Se debe recibir una confirmación tras la eliminación exitosa.

Historia 4: Autenticación de usuario

Como usuario, quiero iniciar sesión en la plataforma, para acceder a mis documentos y funcionalidades personalizadas.

Criterios de aceptación:

- Debe solicitar credenciales de usuario correo y contraseña.
- Si las credenciales son correctas, se debe generar un token JWT.
- Si las credenciales son incorrectas, se debe devolver un mensaje de error adecuado.

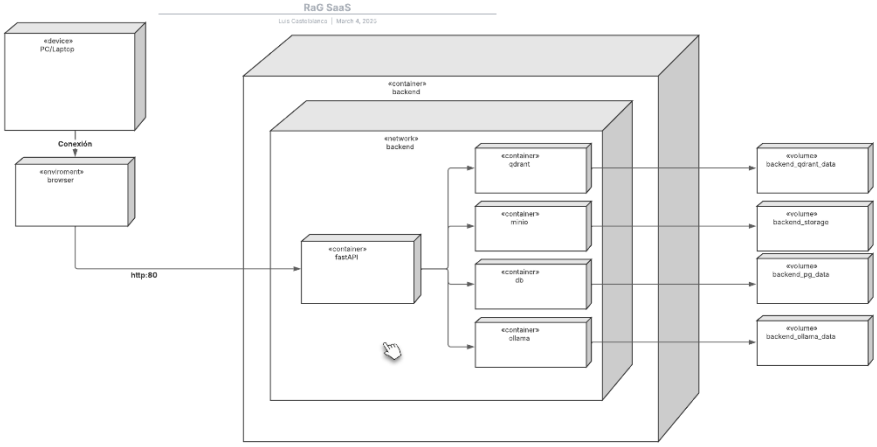
Historia 5: Generación de respuestas con IA

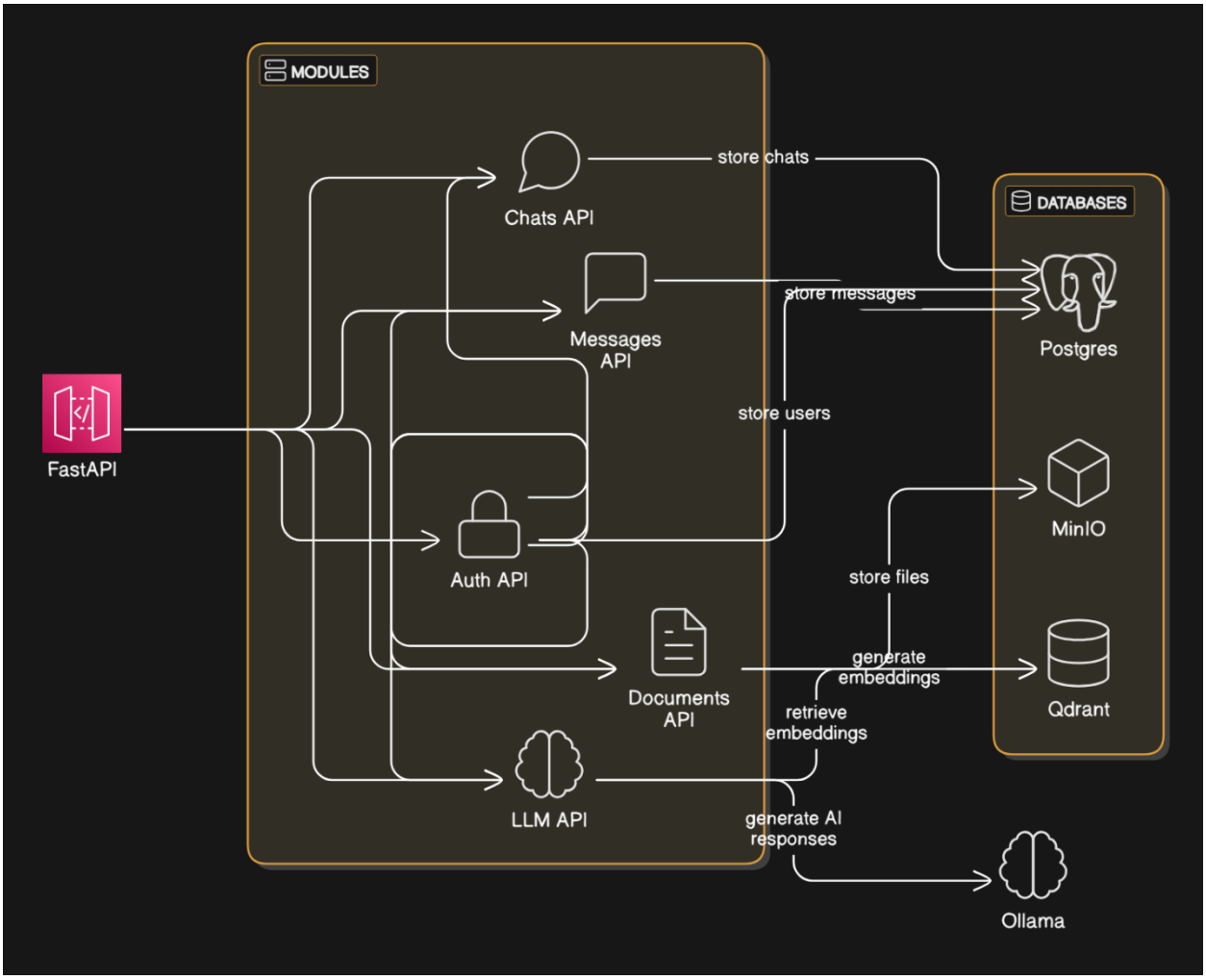
Como usuario autenticado, quiero enviar una consulta a la IA basada en mis documentos, para recibir respuestas contextualizadas.

Criterios de aceptación:

- El usuario debe estar autenticado.
- La consulta debe ser procesada por el modelo Llama 3 8B.

3. Diagramas de Despliegue y Arquitectura


RaG SaaS
«device»
PC/Laptop
Conexión
«environment»
browser
http:80
«container»
backend
«network»
backend
«container»
fastAPI
«container»
qdrant
«container»
minio
«container»
db
«container»
ollama
«volume»
backend_qdrant_data
«volume»
backend_storage
«volume»
backend_pg_data
«volume»
backend_ollama_data


MODULES
Chats API
store chats
Messages API
store messages
store users
FastAPI
Auth API
Documents API
LLM API
DATABASES
Postgres
MinIO
store files
Qdrant
generate embeddings
retrieve embeddings
generate AI responses
Ollama

4. Endpoints

A continuación, listamos los endpoints disponibles en la API:

Gestión de Documentos *(/users/{user_id}/documents/)*

- *GET /users/{user_id}/documents/:* Obtener los documentos de un usuario.
- *POST /users/{user_id}/documents/:* Subir un nuevo documento asociado a un usuario.
- *DELETE /users/{user_id}/documents/{document_id*}: Eliminar un documento específico.

Gestión de Chats *(/users/{user_id}/chats/)*

- *GET /users/{user_id}/chats/:* Obtener los chats de un usuario.
- *POST /users/{user_id}/chats/:* Crear un nuevo chat asociado a un usuario.
- *GET /users/{user_id}/chats/{chat_id}:* Obtener un chat específico de un usuario.
- *DELETE /users/{user_id}/chats/{chat_id}:* Eliminar un chat específico de un usuario.

Mensajes en Chats *(/chats/{chat_id}/messages/)*

- *GET /chats/{chat_id}/messages/:* Obtener los mensajes de un chat.
- *POST /chats/{chat_id}/messages/:* Enviar un nuevo mensaje a un chat.

Procesamiento con LLM (/llm/generate)

- *POST /llm/generate:* Generar una respuesta basada en una consulta de usuario.

5. Instrucciones para la Ejecución y Despliegue

Para ejecutar la API localmente utilizando Docker, usa el siguiente comando:

docker-compose up --build