

Proyecto Banco Guayaquil – Agente de IA Financiero con RAG | Chapter Lead AI / Isaac Reyes

1. Visión General

Este proyecto implementa un **Agente Inteligente Financiero** diseñado para responder preguntas sobre documentos financieros internos (en PDF) mediante una arquitectura **RAG (Retrieval-Augmented Generation)**, complementado con herramientas externas como una **API de conversión de divisas**. El agente está pensado para ayudar a analistas, gerentes financieros o clientes internos de Banco Guayaquil a obtener insights rápidos, precisos y confiables.

2. Caso de Uso

“Un analista financiero desea saber qué pasivos corrientes registró la empresa XYZ en el último trimestre, y cuánto representa esto en USD al tipo de cambio actual.”

3. Arquitectura del Sistema

Componentes Principales

- **Frontend Web:** Interfaz minimalista en React/Next.js.
- **Backend API:** FastAPI como servidor central.
- **Agente RAG:** Construido con LangChain.
- **Vector DB:** FAISS o Chroma con embeddings OpenAI.
- **Tool Externa:** Conversión EUR→USD usando una API de tasas de cambio.
- **PDF Loader:** Procesador de PDFs con PyMuPDF.

Flujo RAG

1. El usuario hace una consulta desde el frontend.
2. El agente decide:
 - ¿Consultar documentos vectorizados?
 - ¿Llamar la tool de conversión?
3. La respuesta generada combina contexto financiero y funciones auxiliares.

4. Detalle Técnico del Agente

Revisar github IGRA27

5. Tool Externa Integrada

API de Conversión de Divisas

```
import requests

def eur_to_usd():
    response = requests.get("https://api.exchangerate.host/latest?base=EUR&symbols=USD")
    return response.json()["rates"]["USD"]
```

La función se utiliza directamente como `tool` dentro del agente, permitiendo respuestas dinámicas.

6. Arquitectura Visual

```
flowchart TD
    User[Usuario] --> Front[Frontend Web (Next.js)]
    Front --> FastAPI
    FastAPI --> Agent[Agente LangChain]
    Agent --> VectorDB[Vector Store (FAISS)]
```

Agent → FXAPI[API Tasa de Cambio]
VectorDB → Docs[Documentos PDF Financiero]

7. Mejores Futuras

- Añadir autenticación (OAuth).
- Añadir logging estructurado y trazabilidad de herramientas usadas.
- Soporte para múltiples monedas y rangos temporales.
- Razonamiento avanzado con LangGraph.

8. Repositorio y Ejecución

- GitHub: <https://github.com/BancoGuayaquil/ai-financial-agent>
- Frontend local: `npm run dev`
- Backend local: `uvicorn app.main:app --reload`

9. Conclusión

Este agente permite acelerar la toma de decisiones financieras internas al integrar la búsqueda semántica de documentos con herramientas útiles como la conversión monetaria en tiempo real. Es una muestra clara de cómo la IA puede integrarse de forma segura, útil y extensible en el ecosistema del Banco Guayaquil.

Este archivo lo hice con Notion