



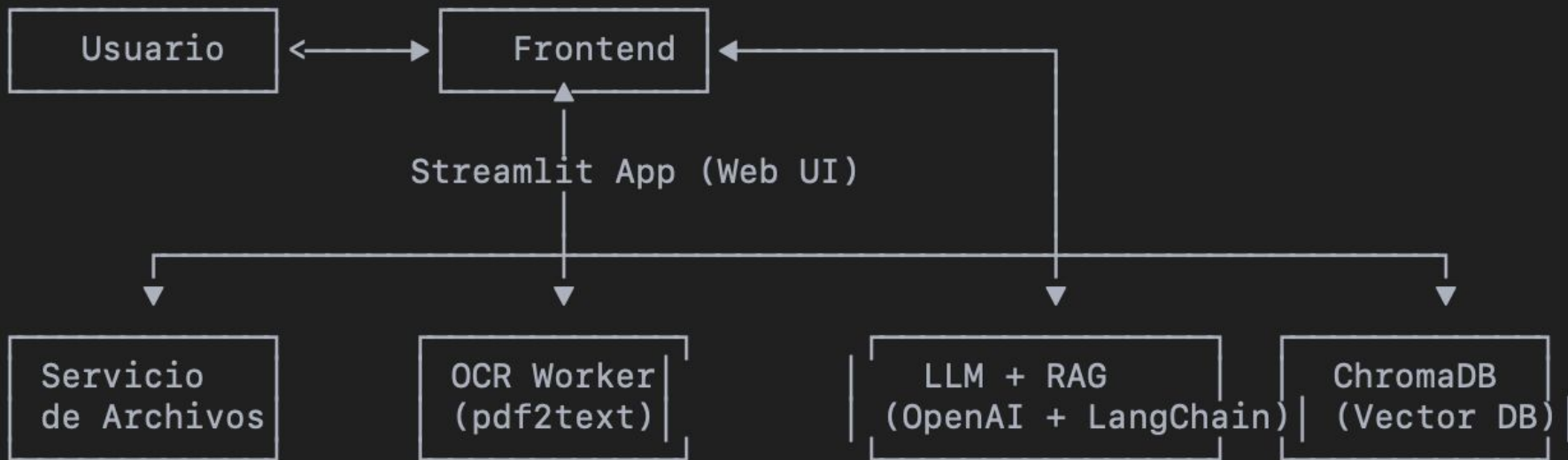
PRESENTACIÓN SISTEMA

Ixent Cornella, Oliver Chan, Juan Jose Acevedo, Carlos Rodríguez y Carla Claramunt



Problema

Arquitectura General





Contenedores Principales

- Frontend: interfaz de usuario (Streamlit).
- OCR: servicio que convierte los PDFs en texto usando Tesseract OCR.
- RAG: core del sistema que maneja la lógica de recuperación y generación de respuestas.
- Chromadb: base de datos vectorial que guarda los embeddings generados a partir del texto.



Flujo de Datos

1. El usuario carga un PDF desde el frontend.
2. Se envía al contenedor ocr, que extrae el texto.
3. El texto se pasa al contenedor rag, que:
 - Genera embeddings con LangChain.
 - Los almacena en chromadb.
 - Genera una respuesta usando OpenAI (OPENAI_API_KEY desde .env).
4. La respuesta se muestra al usuario en la interfaz.

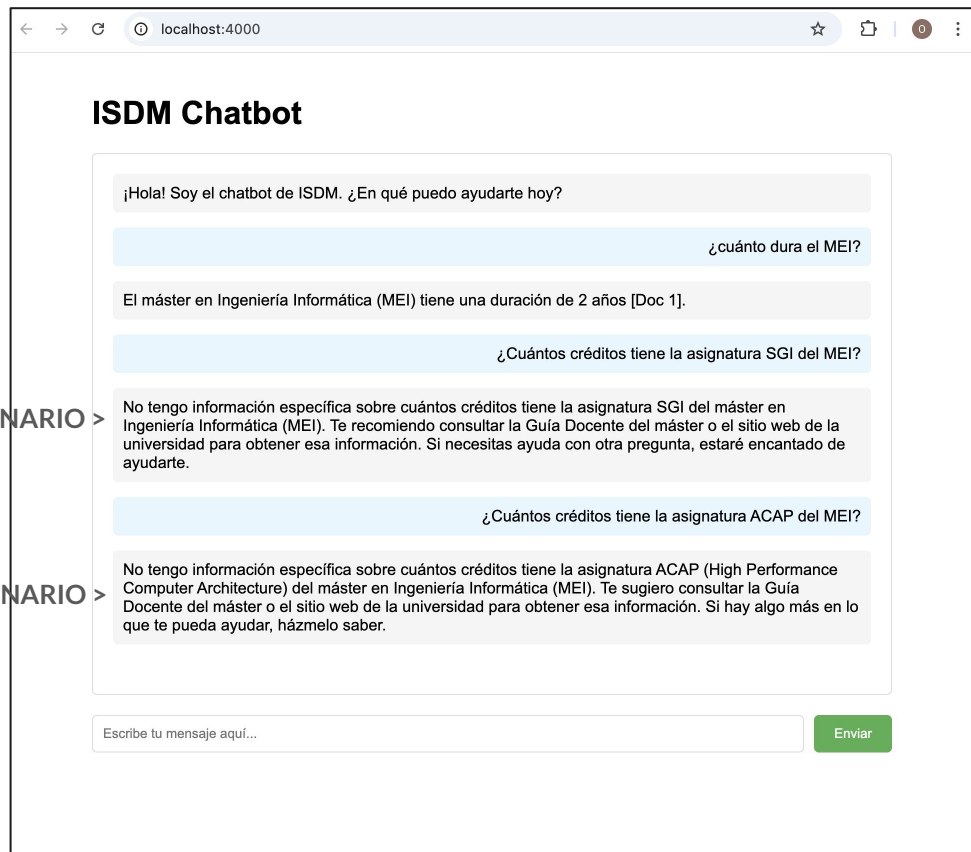
Resultados

- El sistema permite hacer preguntas complejas basadas en documentos cargados.
- Usa OCR para transformar documentos escaneados en texto útil.
- Integra recuperación semántica con generación vía OpenAI.

SIN USO DE DICCIONARIO >

CON USO DE DICCIONARIO >

```
{
  "MIRI": "Master in Innovation and Research in Informatics",
  "MDS": "Master in Data Science",
  "MUI": "Master in User Interface",
  "MOSIG": "Master of Science in Informatics at Grenoble",
  "MOS": "Master of Science",
  "MOSIG": "Master of Science in Informatics at Grenoble",
  "MXX": "Master of Science in XX",
  "MY": "Master of Science in YY",
  "MZZ": "Master of Science in ZZ",
  "ACAP": "High Performance Computer Architecture"
}
```





Mejoras

- Soporte multilingüe (OCR y LLM).
- Persistencia de sesiones de usuario.
- Mejora de la interfaz visual.
- Evaluación automática de la calidad de respuesta.



Créditos

- <https://github.com/DGSI-UPC/llm-chat-rag>
- <https://chatgpt.com>