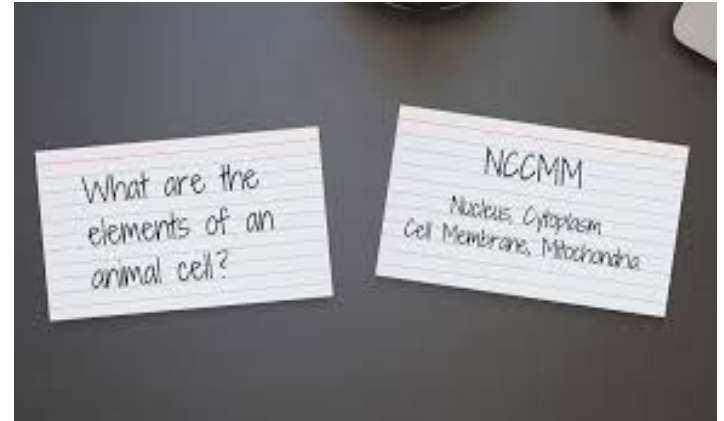


Chat with PDF to create flashcards

- Arriola Aldo
- Castro Carlo
- Barrios Julio
- Fuentes Carpio Patrick Alonso
- Pérez Alvarado Roberth



Caso de Estudio

Este caso consiste en la creación de un pipeline local de **Retrieval Augmented Generation (RAG)**, el cual permite interactuar de manera dinámica con archivos PDF mediante preguntas. Este proyecto combina técnicas de procesamiento de lenguaje natural y recuperación de información para generar respuestas basadas en un documento PDF cargado previamente. A continuación, se describen los pasos principales del proceso:

1. Carga del archivo PDF: Se utiliza la herramienta **UnstructuredPDFLoader** para cargar el contenido del archivo PDF en el sistema. Esta herramienta extrae el texto no estructurado, lo cual es útil para el posterior procesamiento del documento.

2. División del texto en fragmentos: Una vez que el PDF ha sido cargado, el contenido se divide en fragmentos más pequeños mediante **RecursiveCharacterTextSplitter**. Este paso es fundamental para que el modelo pueda manejar el texto de forma más eficiente, ya que los grandes volúmenes de datos no se procesan de manera óptima en una sola instancia.

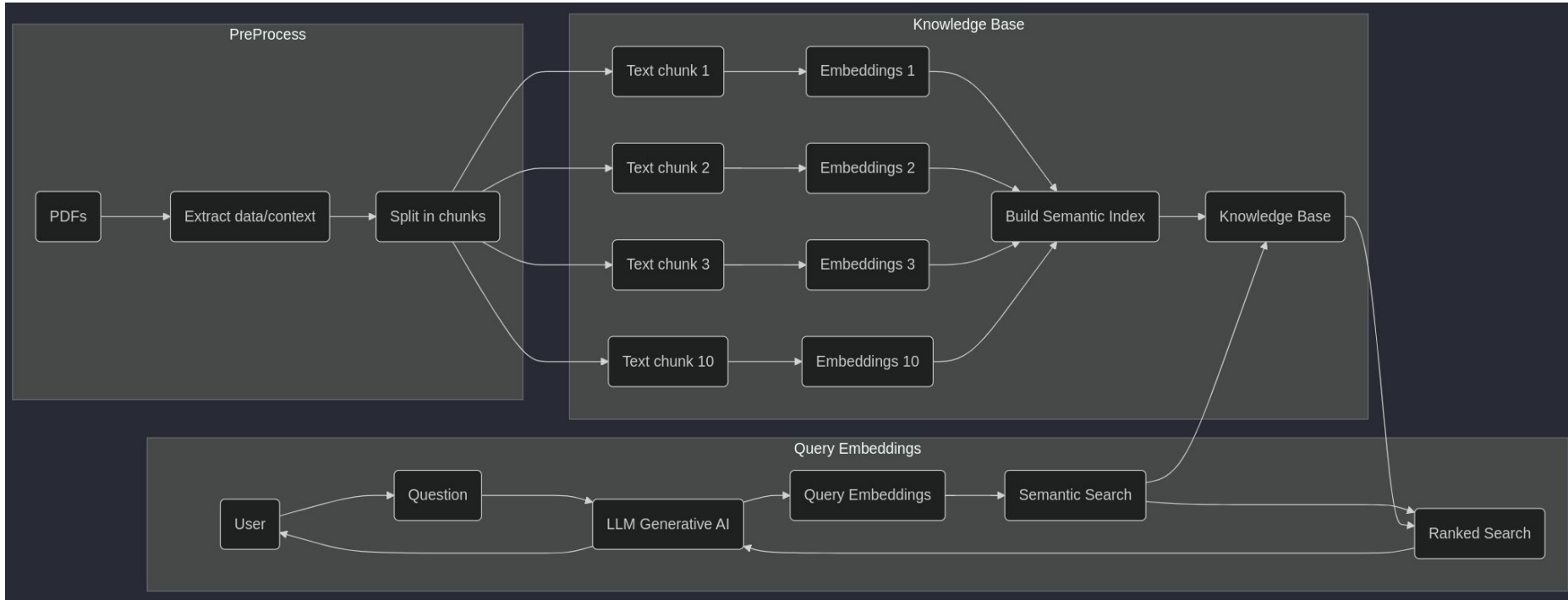
3. Creación de embeddings: Posteriormente, se crean embeddings de cada fragmento del texto usando **OllamaEmbeddings**. Los embeddings son representaciones vectoriales del texto que permiten al modelo comprender mejor el significado semántico de cada fragmento.

4. Creación de una base de datos vectorial: A continuación, se utiliza el método `from_documents` de Chroma para construir una base de datos vectorial, donde cada fragmento de texto se asocia con su embedding correspondiente. Esta base de datos es esencial para la recuperación eficiente de información, ya que permite buscar y comparar rápidamente los fragmentos más relevantes para una consulta específica.

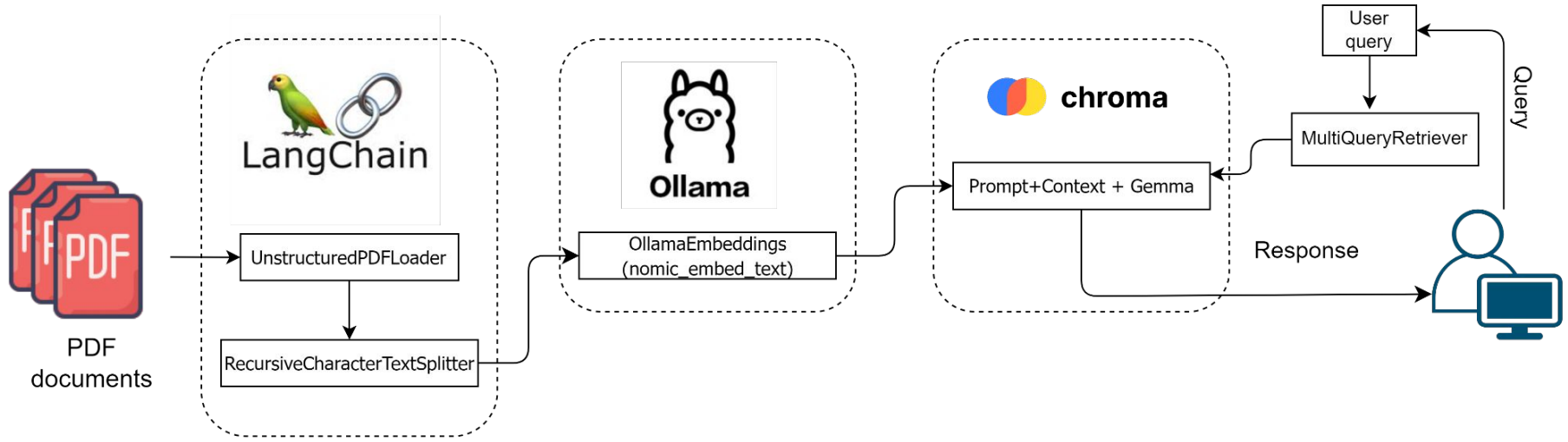
5. Respuesta a preguntas: Finalmente, se implementa el método `chain.invoke`, que permite generar respuestas a preguntas basadas en el contenido del documento PDF. El modelo utiliza la base de datos vectorial para recuperar el contexto más relevante de acuerdo con la pregunta formulada, generando así una respuesta precisa basada en la información del PDF.

En resumen, este caso de estudio demuestra cómo combinar técnicas de procesamiento de lenguaje natural, como la generación de embeddings y la creación de bases de datos vectoriales, para permitir una interacción fluida y eficaz con documentos PDF. El sistema puede responder a preguntas de manera precisa y generar respuestas fundamentadas en el contenido del archivo, lo que facilita el acceso a la información contenida en documentos extensos.

Diagrama de Procesos



Tecnologias



**HERE
WE
CODE**

Resultados

```
1) **Title:** Software Architecture Visualization
   **Question:** What is a software architecture visualization?
   **Answer:** A software architecture visualization is a choice, not a given. It includes certain elements and excludes others, emphasizes some aspects and de-emphasizes others, based on the designer's choices.

2) **Title:** Software Architecture and Physical Constraints
   **Question:** Why do physical constraints matter in software architecture?
   **Answer:** Physical constraints like processor speed, network bandwidth, memory, and storage limit the performance and potential of a software system.

3) **Title:** Shakespeare's Quote on Software Architecture
   **Question:** What does this quote from William Shakespeare imply about software architecture?
   **Answer:** The quote suggests that while our ambitions for a software system may be infinite, the execution is confined and subject to physical limitations.

4) **Title:** Grady Booch's Definition of Architecture
   **Question:** What does Grady Booch define as architecture in software systems?
   **Answer:** According to Grady Booch, architecture represents the significant design decisions that shape a system, where significant is measured by the cost of change.

5) **Title:** Time, Money, and Effort in Software Architecture
   **Question:** What role do time, money, and effort play in determining the quality of software architecture?
   **Answer:** Time, money, and effort help us sort between the large and small, distinguish the architectural stuff from the rest, and determine whether an architecture is good or not by meeting the needs of its users over time.

6) **Title:** Bad Architecture versus Good Architecture
   **Question:** What does Brian Foote and Joseph Yoder say about the cost of bad architecture?
   **Answer:** They argue that if you think good architecture is expensive, try bad architecture.

7) **Title:** Typical Changes in a System's Development
   **Question:** What should typical changes in a system's development not be according to Ralph Johnson?
   **Answer:** According to Ralph Johnson, the kinds of changes a system's development typically experiences should not be the changes that are costly, hard to make, or take managed projects of their own.

8) **Title:** Time Travel and Software Architecture
   **Question:** How do we predict typical changes in a software system without crystal balls and time machines?
   **Answer:** We cannot predict the future precisely; however, we can learn from the past, observe the present, and make informed decisions based on that knowledge.

9) **Title:** Architecture as a Hypothesis
   **Question:** According to Tom Gilb, what is software architecture?
   **Answer:** Tom Gilb defines architecture as a hypothesis that needs to be proven by implementation and measurement.

10) **Title:** Walking the Path of Good Software Architecture
    **Question:** What qualities does walking the path of good software architecture require?
    **Answer:** Walking the path of good software architecture requires care, attention, thought, observation, practice, and principle. It recognizes the softness of software and operates with incomplete knowledge, but is good at discovering and creating things through asking questions and running experiments.
```

Referencias

- Kipkemboi, T. (n.d.). Local Ollama RAG. GitHub.
https://github.com/tonykipkemboi/ollama_pdf_rag/blob/main/local_ollama_rag.ipynb