JavaUnitTestRAG

Patricio Escudeiro, Felipe Hiba, Jose Burgos

Dataset

Tipo de datos

¿De dónde salen?

https://github.com/DanAg278/Java-Unit-Testing

https://github.com/christian-kesler/junit-testing-java

https://github.com/rieckpil/java-testing-toolbox

https://github.com/fhiba/paw-medics

https://github.com/fhiba/gamer-grove

Pre procesamiento

- Claude 3.7 Sonnet & o4-mini-high
- descripciones de las clases

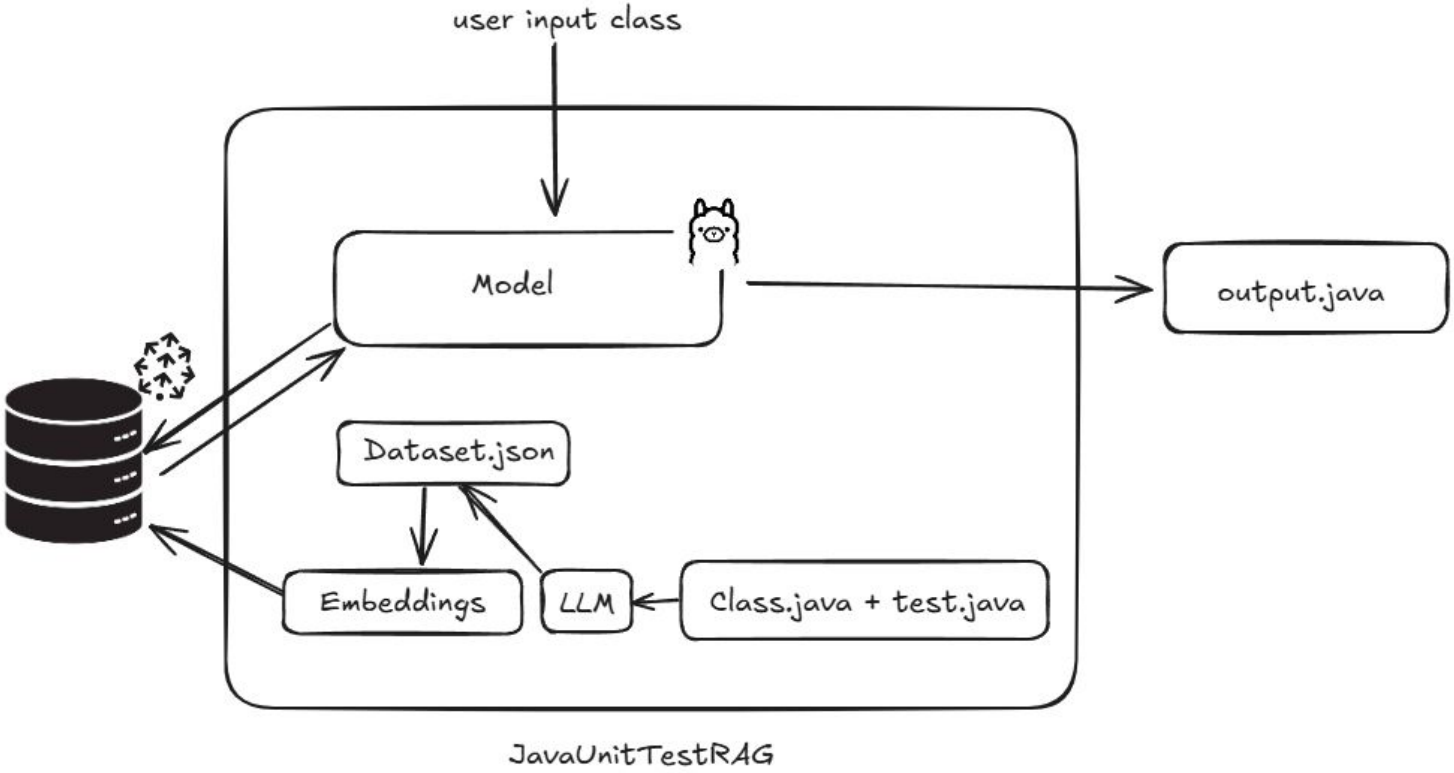
public class Adder {...} -> description: "A simple class that adds two integers together and returns the result."

Text splitting, embeddings and vector DB

- Chunk Size : 750
- Overlap : 150
- Para embeddings: 'sentence-transformers/all-MiniLM-L6-v2'
  - de los endpoints de hugging face
- Utilizamos pinecone DB
  - debimos setear las dimensiones a 384 ya que el default 1024 no era compatible con nuestro RAG

Arquitectura

Arquitectura



Prompt

You are an autonomous code generation agent. Your task is to write **new** unit tests using **JUnit 5** for a Java class provided.
IMPORTANT RULES:

1. You MUST return only valid Java test code using `@Test` and assertion methods such as `assertEquals`, `assertTrue`, `assertThrows`.

2. You MUST write tests that are relevant to the user's request.

3. DO NOT include `import` statements, comments, class headers, or method explanations.

4. NEVER repeat code from the original class or documentation.

5. You MUST generate at least one passing test and one failing test **if applicable**.

6. You may assume the test class is already defined and has access to the class under test.

RESPONSE FORMAT: Only raw Java methods

CONTEXT INFORMATION (retrieved from knowledge base):

{info}

USER REQUEST:

{prompt}

Generate the test methods now.

Desafíos

- Bajo rendimiento de modelos locales
- Generación del dataset
- Dificultades con RAGAs

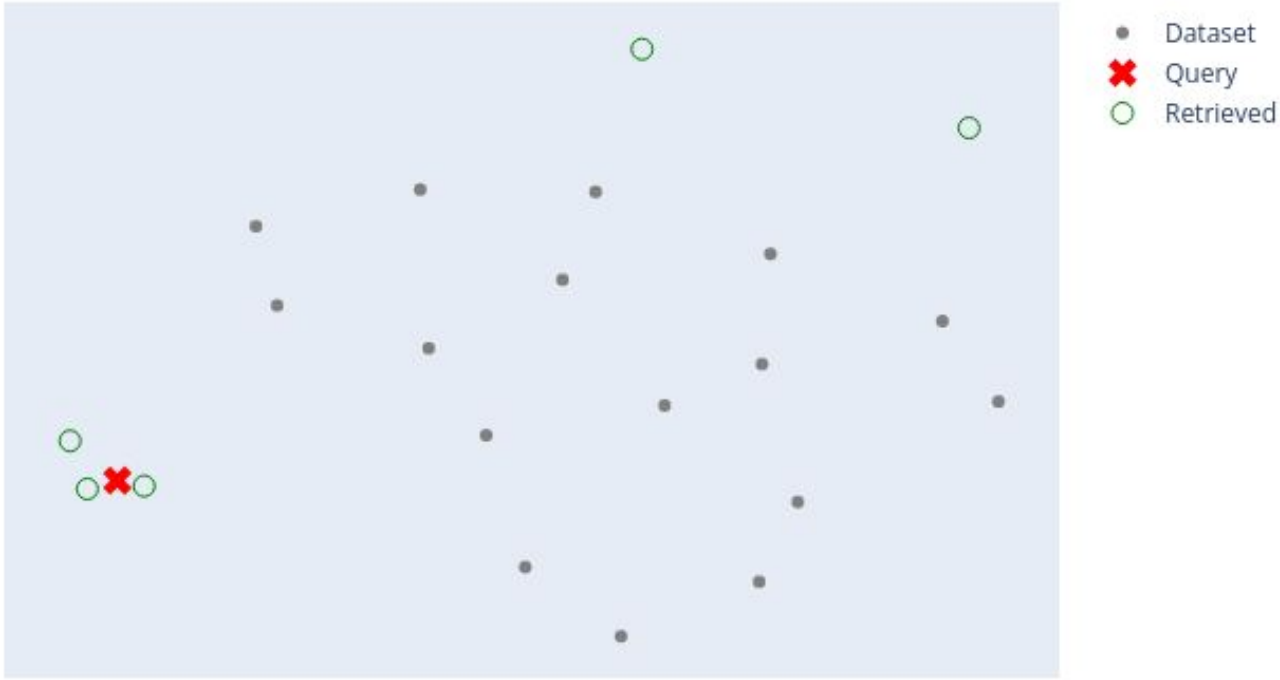
Demo

Resultados

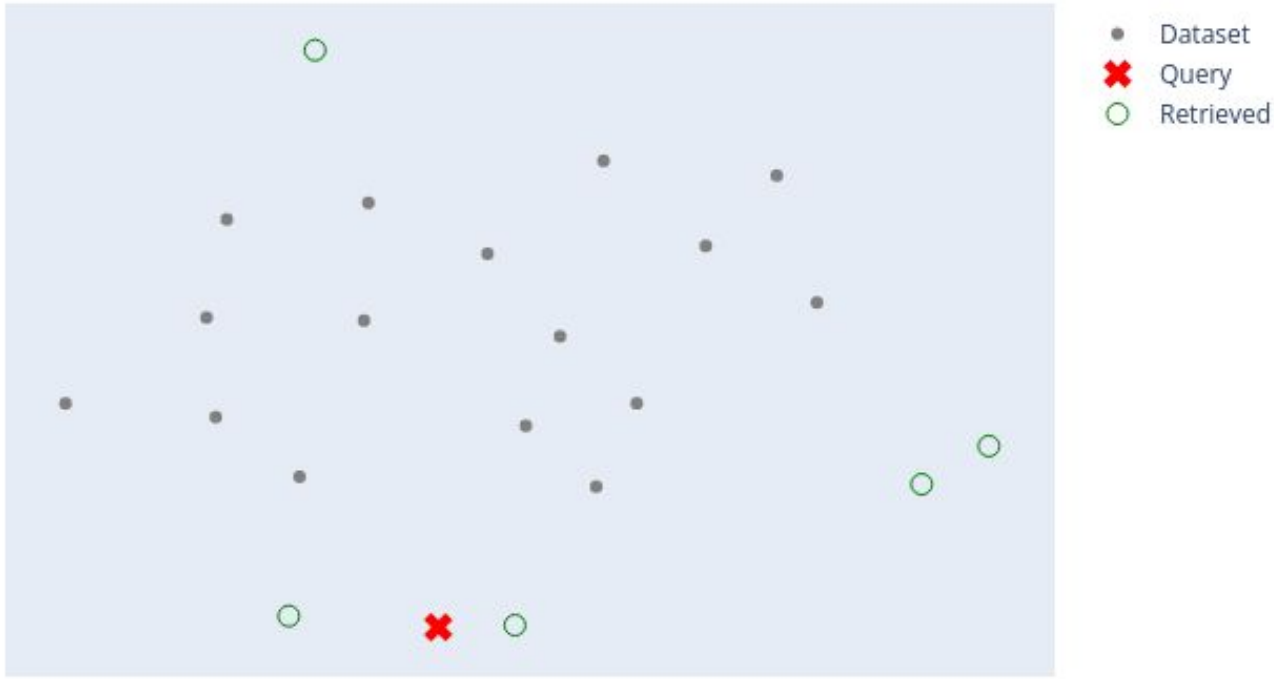
Métricas

- Cantidad de Test Pasados
- Cantidad / Tipos de errores de test
- Calidad de los test

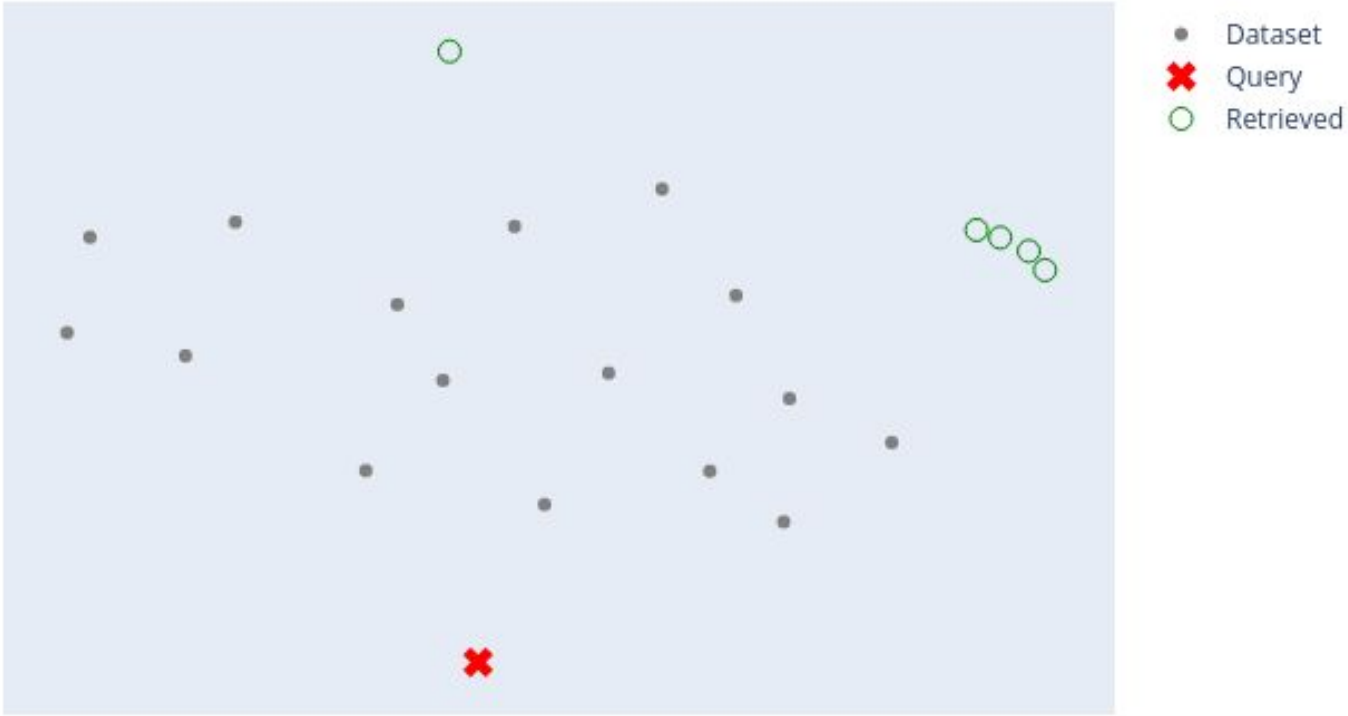
OrderProcessor RAG Projection



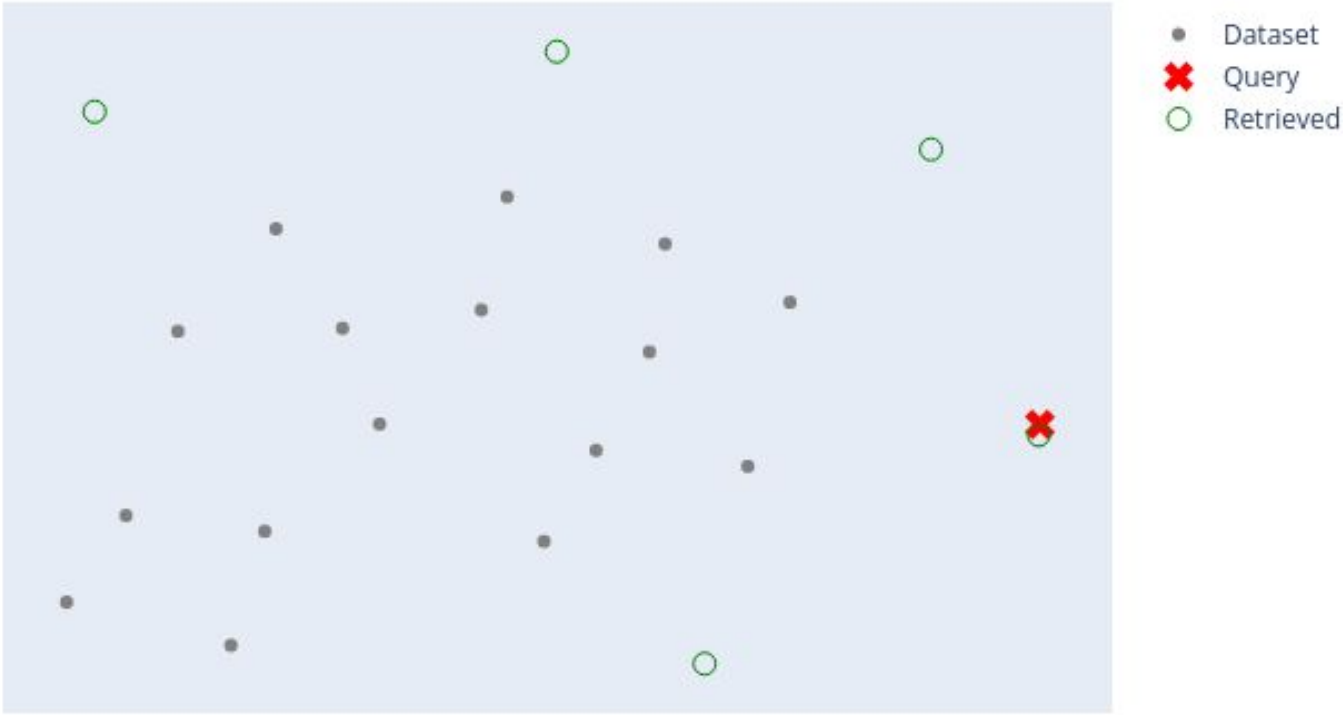
MatrixUtils RAG Projection



UserManager RAG Projection

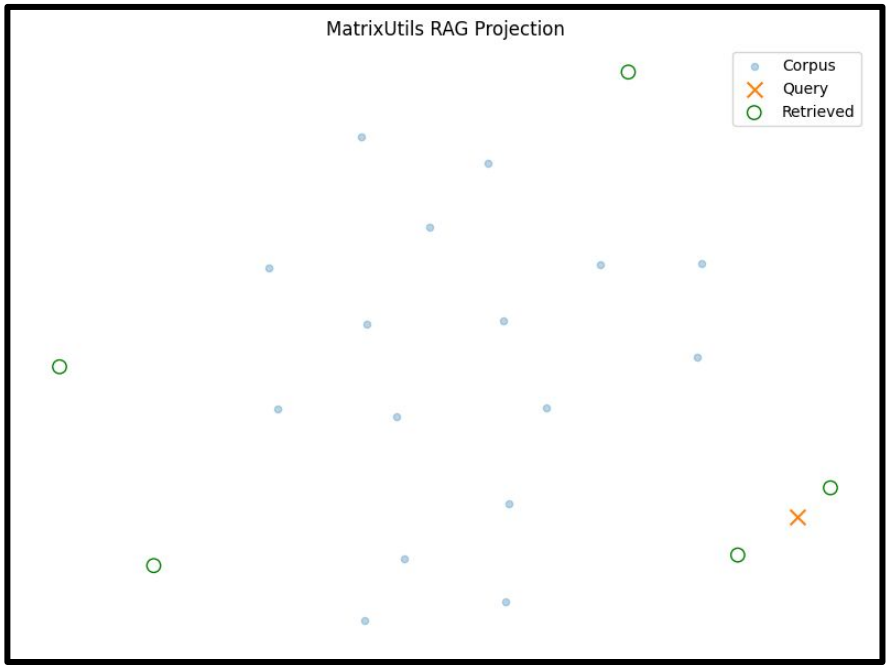


CsvParser RAG Projection

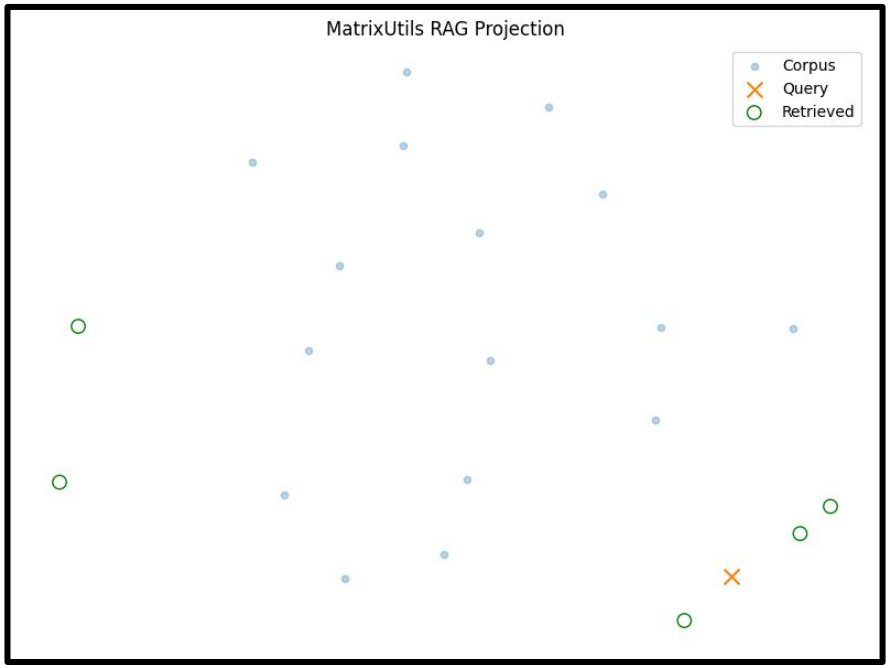


RAGAS

No Reranking

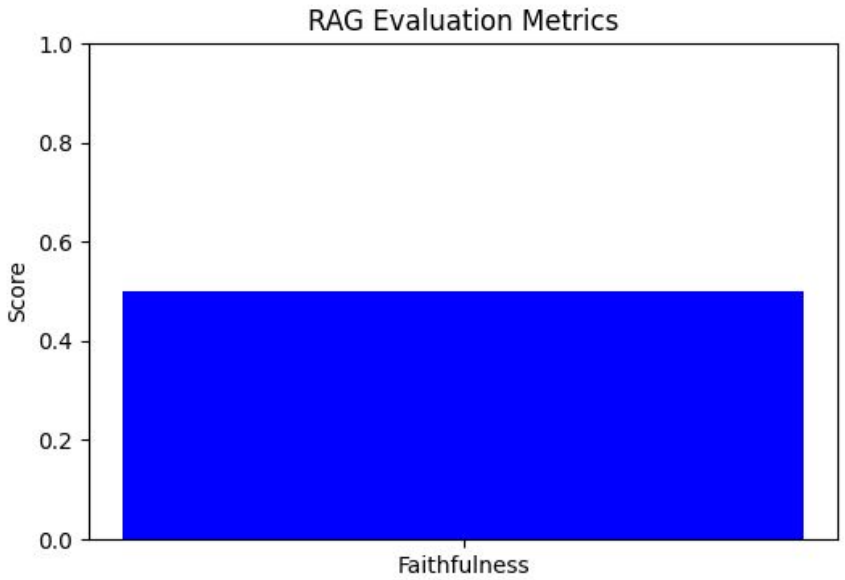


Reranking

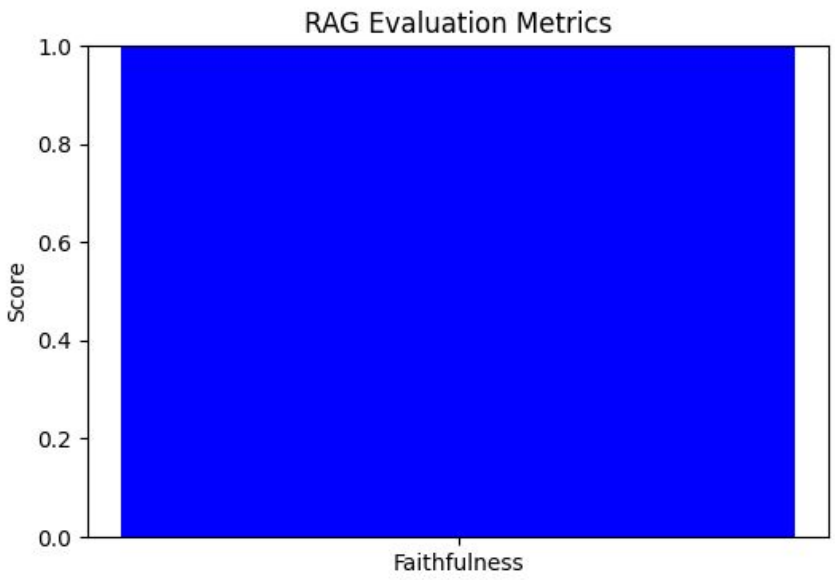


Faithfulness

No Reranking

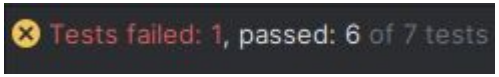


Reranking



RAG Assessment

Order Processor (6 funciones)

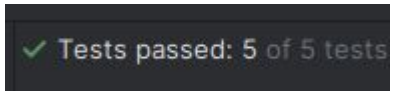


Error de Exception equivocada

Matrix Útil (2 funciones)

Error de Exception equivocada
Error del assert del test

UserManager (3 funciones)



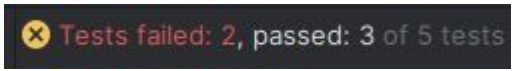
NO RAG Assessment

2 Errores de Assert

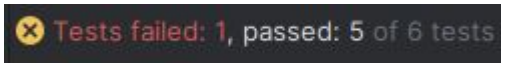
1 Error de Assert

Repite Test

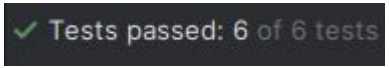
Order Processor NO RAG



Matrix Útil NO RAG



User Manager NO RAG



Errores Comunes

- Error de Excepción en un assert
- Mal acceso de una variable
- Errores en las cuentas

Conclusiones

- Cuanto más alejado están los documentos de la query, peor es el test
- El código de los test con RAG es más prolijo que el de sin RAG
- Es importante tener un buen prompt para que el RAG se comporte de la manera esperada

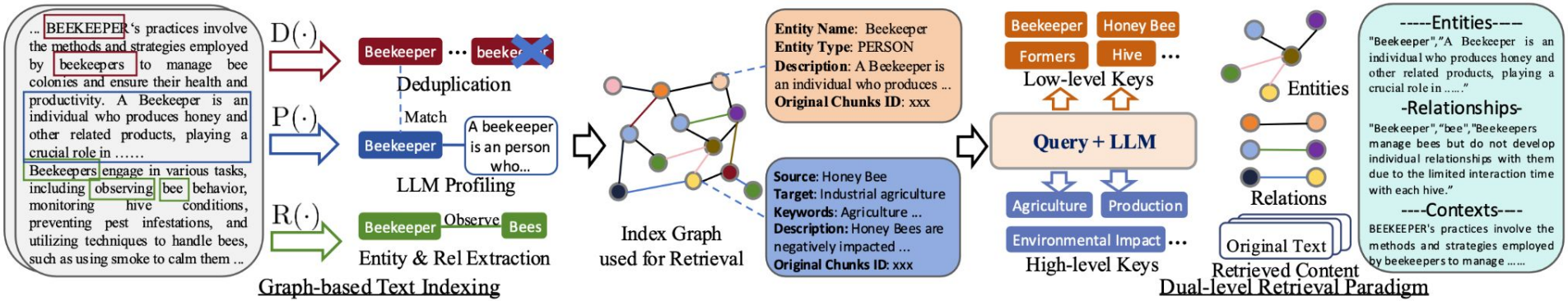


Figure 1: Overall architecture of the proposed LightRAG framework.

Gracias Por Escuchar!