Universitat
de les Illes Balears

TRABAJO DE FIN DE GRADO

DISEÑO Y EVALUACIÓN DE UN SISTEMA DE GENERACIÓN MEJORADA POR RECUPERACIÓN PARA DOCUMENTOS

Lluis Barca Pons

Grado en Ingeniería Informática

Escola Politècnica Superior

Año académico 2023-24

DISEÑO Y EVALUACIÓN DE UN SISTEMA DE GENERACIÓN MEJORADA POR RECUPERACIÓN PARA DOCUMENTOS

Lluis Barca Pons

Trabajo de Fin de Grado

Escola Politècnica Superior

Universitat de les Illes Balears

Año académico 2023-24

Palabras clave del trabajo: Documentos, Embedding Models, Inteligencia Artifical Generativa, Large Language Models, Retrieval Augmented Generation

Tutores: Isaac Lera Castro y Antoni Jaume Capó

Autorizo a la Universidad a incluir este trabajo en el repositorio institucional para consultarlo en acceso abierto y difundirlo en línea, con finalidades exclusivamente académicas y de investigación	Autor/a Sí	Autor/a No	Tutor/a Sí	Tutor/a No
	☑	☐	☑	☐

Agradecer profundamente a mi familia por la paciencia y los recursos necesarios para crecer y desarrollarme como persona. A Lucia, por aguantar los días malos, pero sobre todo por disfrutar los buenos.

También a todas aquellas personas que han pasado por mi vida en estos años y me han marcado de una forma u otra. Acordarme de Anna por creer en mí y marcarme el camino a seguir; Joan por abrirme al mundo de la informática y valorarme cuando poca gente lo hacía; a mis tutores por su magnífico trabajo y a mis compañeros, que me han motivado a siempre dar un poco más. Sin todos ellos, no sería la persona y el informático que soy.

ÍNDICE GENERAL

ACRÓNIMOS

API Aplication Programming Interface

GPT Generative Pre-trained Transformer

IDE Integrated Development Environment

IA Inteligencia Artificial

LLM Large Language Models

ML Machine Learning

PLN Procesamiento del Lenguaje Natural

RAG Retrieval-Augmented Generation

SLM Small Language Models

RESUMEN

Este trabajo final de grado se centra en el diseño, implementación y evaluación de un sistema de Generación Aumentada por Recuperación (o más conocido por su término en inglés como *Retrieval-Augmented Generation*) que opera en un entorno local, utilizando modelos de código abierto disponibles en la plataforma *HuggingFace*. El principal objetivo ha sido garantizar la privacidad y seguridad de los datos al no depender de servicios en la nube, manteniendo un rendimiento competitivo en términos de recuperación y generación de información a partir de documentos en formato PDF.

A lo largo del proyecto, se ha logrado implementar un sistema que permite la ingesta de documentos y la posterior generación de respuestas a consultas sobre estos. Se han evaluado diferentes modelos de *embeddings* y modelos de lenguaje de gran tamaño (o más conocidos por su término en inglés *Large Language Models*) para identificar las combinaciones más eficaces en tareas de preguntas y respuestas. Los experimentos han mostrado que, dependiendo del tipo de documento y la naturaleza de las consultas, los modelos ofrecen una mayor o menor precisión en la recuperación de contextos relevantes. Además, se ha llevado a cabo una evaluación multilingüe del sistema en tres lenguas: inglés, castellano y catalán. Los resultados reflejan la capacidad del sistema para generar respuestas precisas en los tres idiomas, demostrando su versatilidad. No obstante, también se han identificado áreas de mejora, como la optimización del proceso de recuperación de la información en diferentes documentos.

En conclusión, el sistema desarrollado no solo ha cumplido con los objetivos propuestos, sino que también ha mostrado ser un enfoque viable para la recuperación y generación de información en un entorno controlado y seguro, abriendo nuevas oportunidades para su desarrollo futuro en aplicaciones prácticas donde la privacidad y la seguridad son factores clave.

CAPÍTULO 1

INTRODUCCIÓN

Para la elaboración de este trabajo, se han utilizado herramientas de Inteligencia Artificial (IA) Generativa con la finalidad de asistir en la redacción del texto. Sin embargo, todo el contenido ha sido supervisado y revisado por mí para garantizar la precisión y coherencia de la información presentada.

La IA ha experimentado un avance significativo en los últimos años, proporcionando soluciones innovadoras a problemas complejos que tradicionalmente requerían intervención humana. Este progreso se ha visto impulsado por el desarrollo tecnológico en la computación masiva de datos y la implementación de sofisticados algoritmos capaces de simular el intelecto humano [1].

Dentro del campo de la IA encontramos el conocido como Aprendizaje Automático, o más conocido por su versión inglesa *Machine Learning (ML)*. Este campo se centra en el diseño de algoritmos que identifican una serie de patrones y relaciones en los datos, que posteriormente se explota para tomar ciertas decisiones o realizar diversas predicciones. Dentro de esta familia de algoritmos podemos distinguir tres ramas claramente diferenciadas:

- **Aprendizaje Supervisado**: cuando un modelo se entrena utilizando un conjunto de datos etiquetado. Cuando hablamos de datos "etiquetados" nos referimos a que la información con la cual se realiza el entrenamiento del modelo, se asocia a un resultado correcto [2]. Podemos diferenciar entre dos tipos de datos:
  - **Clasificación**: Asignamos una etiqueta a cada dato de entrada. Por ejemplo, con los mensajes de correo electrónico podemos clasificarlo por *spam* o *no spam*.
  - **Regresión**: Se basa en predecir un valor continuo. Por ejemplo, se utiliza para predecir el valor de una casa en función de sus características de mercado.

- **Aprendizaje No Supervisado**: En este caso, los datos no se encuentran etiquetados. El objetivo no es predecir un dato, como en el aprendizaje automático, es encontrar patrones o relaciones entre los datos. Por ejemplo, para la segmentación de clientes en cualquier negocio [3]. Se diferencian en dos tipos:

  - **Agrupamiento (o *clustering* en inglés)**: El *clustering* se basa en dividir un conjunto de datos en grupos (*clusters*) donde los elementos dentro de cada grupo son más similares entre sí que a los elementos de otros grupos. El objetivo es descubrir la estructura subyacente en los datos sin necesidad de tenerlos previamente etiquetados.
  - **Reducción de dimensionalidad (*dimensionality reduction*)**: Esta tarea busca identificar y capturar la estructura interna de los datos de alta dimensión en un espacio de menor dimensión, a menudo revelando patrones subyacentes y facilitando la visualización o el análisis posterior.

- **Aprendizaje por Refuerzo**: Un agente aprende a tomar decisiones mediante la interacción con un entorno, con el objetivo de maximizar una recompensa acumulada a lo largo del tiempo. A diferencia del aprendizaje supervisado, donde el modelo se entrena con pares de entrada/salida etiquetados, en el aprendizaje por refuerzo, el agente aprende a través de prueba y error [4]. A continuación los diferentes tipos que podemos encontrar:

  - **Métodos basados en Valor (*Value-Based Methods*)**: Estos métodos se centran en aprender una función de valor, que es una estimación de la recompensa esperada que puede obtenerse desde un estado dado (o estado-acción) siguiendo una política determinada.
  - **Métodos basados en Política (*Policy-Based Methods*)**: En lugar de aprender valores para las acciones, estos métodos aprenden directamente una política, es decir, una función que mapea estados a acciones. Los métodos basados en política ajustan directamente los parámetros de la política para maximizar la recompensa acumulada.
  - **Métodos basados en Modelos (*Model-Bases Methods*)**: Estos métodos implican la construcción de un modelo del entorno, que incluye la transición de estados y las recompensas. El modelo se utiliza para planificar y predecir futuros estados y recompensas, lo que ayuda al agente a tomar decisiones informadas.

Por otro lado, encontramos el conocido como *Deep Learning* o Aprendizaje Profundo. Este se trata de una subárea del ML que se enfoca en el uso de redes neuronales artificiales con múltiples capas para modelar y aprender representaciones jerárquicas de datos complejos. Estas redes neuronales están inspiradas en la estructura y funcionamiento del cerebro humano, particularmente en la forma en que las neuronas se conectan y procesan la información. Se diferencia con el ML en varios aspectos como las técnicas utilizadas o la complejidad de sus modelos.

En este documento nos centraremos con los dos primeros (el supervisado y no supervisado), a los cuales solemos relacionar con los modelos *Generative Pre-trained*

Transformer (GPT), que se tratan de unos modelos que han sido preentrenados con millones datos. Sin embargo, estos modelos son una mezcla entre el aprendizaje supervisado y no supervisado; conocido como aprendizaje semisupervisado. Este tipo de aprendizaje se basa en el aprendizaje mediante información etiquetada y no etiquetada para entrenar dicho modelo [5].

Un *transformer* [6] o transformador es una arquitectura basada en redes neuronales diseñada principalmente para el procesamiento de secuencias, como el lenguaje natural. Se basa en un mecanismo de atención que permite al modelo enfocarse en diferentes partes de la secuencia de entrada de datos de manera más eficiente que los modelos secuenciales anteriores, como las redes neuronales recurrentes (RNN) o las redes neuronales recurrentes de largo corto plazo. Existen también modelos de redes neuronales profundas, capaces de crear nuevo contenido a partir de grandes volúmenes de datos. Estos modelos nos puede sonar principalmente por el ya conocido ChatGPT de OpenAI [7]. Ambos modelos se engloban en la familia de la IA Generativa o *Generative AI*, que nos ha presentado los avances más revolucionarios estos últimos años.

Entrando más en detalle, nuestra propuesta de Trabajo Final de Grado es analizar el funcionamiento de una propuesta de *Retrieval-Augmented Generation (RAG)*. Un RAG es básicamente un sistema de IA que combina técnicas de recuperación de información y generación de texto, para responder preguntas de manera precisa y contextual.

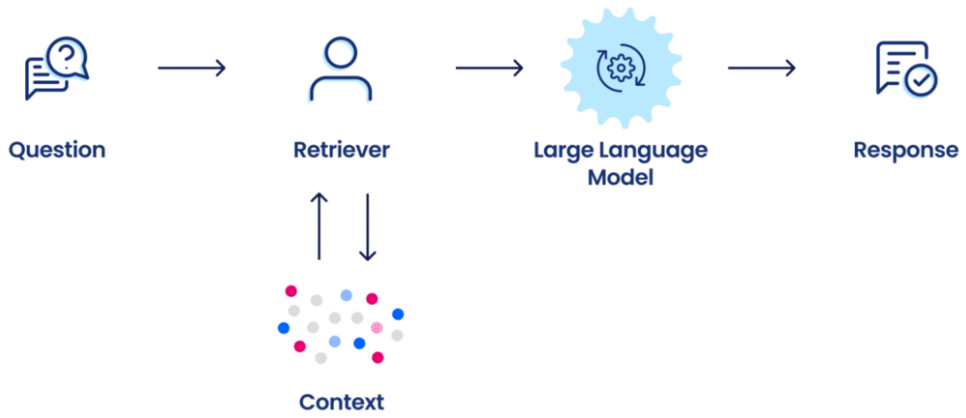

Figura 1.1: Arquitectura simplificada de un RAG [8]

En la figura 1.1 podemos observar como a partir de una *question* (pregunta) formulada por el usuario, se formatea para enviarla al *retriever* (recuperador); un componente encargado de buscar información relevante en una base de datos o conjunto de documentos, que se representa como *context* (contexto). El *retriever* identifica y recupera fragmentos de datos relevantes para responder a la consulta. Estos datos se envían posteriormente al LLM, que utiliza el contexto proporcionado para generar una *response* (respuesta) detallada y coherente. Este proceso permite que el LLM no genere respuestas solo en función de su entrenamiento, sino que integre información relevante y actualizada para ofrecer una respuesta precisa y contextualizada.

Toda esta arquitectura se compone de tres partes fundamentales que a lo largo de la memoria se explicarán con más detalle y son:

1. **Recuperación de la información relevante** → Modelos de Incrustación (o más conocidos como *Embeddings*) [9].
2. **Generación de una respuesta en lenguaje natural** → Modelos de Lenguaje Grande (o más conocidos por su nombre en inglés LLM) [10].
3. **Evaluación del sistema**

1.1. Objetivos

Para llevar a cabo el desarrollo de este Trabajo Final de Grado, se han establecido una serie de objetivos, los cuales serán referenciados a lo largo del documento. Toda la elaboración del mismo se ha realizado aplicando una metodología iterativa y de forma ágil, a medida que se iban incorporando cambios y resolviendo errores. Los objetivos son los siguientes:

O1 Diseño y creación de un RAG en local.

O2 Selección e ingesta de documentos por parte del RAG.

O3 Diseño y evaluación de preguntas para la evaluación sobre el contenido de los documentos.

O4 Analizar la capacidad, tanto de comprensión como de expresión, del sistema en distintas lenguas (inglés, español y catalán).

El enfoque en estos objetivos permitirá llevar a cabo una evaluación exhaustiva y detallada del sistema, garantizando que se aborden todos los aspectos clave necesarios para demostrar su efectividad.

1.2. Requisitos

En esta sección se describen los requisitos fundamentales que guiarán el diseño y desarrollo del RAG para documentos. Estos requisitos abarcan aspectos técnicos y operativos que asegurarán la correcta implementación del sistema, desde la naturaleza y formato de los documentos a procesar, hasta las necesidades de precisión, escalabilidad, y seguridad de los datos. Dichos requisitos han sido establecidos entre mis tutores y yo, a partir de la propuesta inicial de Trabajo Final de Grado.

1. **Formato de los documentos:** El sistema está diseñado específicamente para procesar documentos en formato PDF, que es uno de los formatos más comunes y ampliamente utilizados para la distribución de documentos digitales; especialmente en contextos legales, financieros y académicos. Este enfoque en el formato PDF permite que el sistema se beneficie de las capacidades avanzadas de extracción de texto que esta configuración ofrece. Al centrarse en un solo formato, el sistema puede optimizar su rendimiento en la extracción y procesamiento de información, garantizando una mayor precisión y eficiencia en la generación de respuestas.

2. **Estructura de los documentos:** Los documentos a procesar pueden contener una variedad de estructuras, como:

    - Texto plano
    - Tablas
    - Gráficos
    - Secciones numeradas

El sistema debe ser capaz de identificar y preservar esta estructura durante el procesamiento, asegurando que la información contextual, como las relaciones entre datos en tablas o las jerarquías en secciones, se mantenga intacta y utilizable en la generación de respuestas.

3. **Complejidad del lenguaje:** Dado que los documentos pueden variar en cuanto a la complejidad del lenguaje (desde textos técnicos y legales hasta escritos más coloquiales) y el idioma. El sistema debe estar preparado para manejar diferentes niveles de lenguaje y terminología. Esto implica la utilización de modelos de lenguaje capaces de comprender y procesar tanto lenguaje especializado como general, garantizando respuestas precisas y contextualmente apropiadas; además de realizarlo tanto en inglés, español y catalán.

4. **Precisión en las respuestas:** Es crucial que las respuestas generadas por el sistema sean precisas y estén alineadas con la información contenida en los documentos. Esto requiere que el sistema incorpore mecanismos robustos para verificar la fidelidad de las respuestas, minimizando la generación de información incorrecta o irrelevante. La precisión es especialmente importante en contextos como el análisis de documentos legales o financieros.

5. **Seguridad y privacidad:** El sistema está diseñado para operar completamente de forma local, garantizando que todos los datos procesados se mantengan dentro de la infraestructura del usuario sin necesidad de transmitir información a servidores externos. Esto minimiza los riesgos asociados con la transferencia de datos y garantiza que el control total sobre la información se mantenga en manos de la entidad que utiliza la herramienta. No obstante, el tratamiento de datos desde una perspectiva legal y de cumplimiento de normativas dependerá de la entidad que implemente el sistema.

1.3. Planificación y metodología

El desarrollo de este proyecto se ha realizado siguiendo una metodología ágil, lo que permite iterar sobre los distintos componentes del sistema, integrando cambios y mejoras de forma progresiva. Todo ello con el objetivo de facilitar la incorporación de feedback continuo de mis tutores y permitir adaptaciones rápidas a los problemas técnicos que surgen durante el proceso. A continuación, en la figura 1.2, un diagrama con la planificación a seguir.

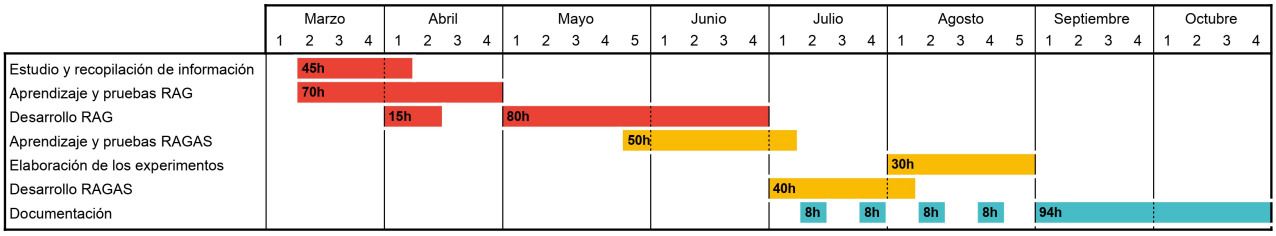

Figura 1.2: Planificación temporal de las tareas

Equivaldría a un reparto de horas como se muestra en la tabla 1.1.

Tarea	Horas
Estudio y recopilación de información	45
Aprendizaje y pruebas RAG	70
Desarrollo RAG	95
Aprendizaje y pruebas RAGAS	50
Elaboración de los experimentos	30
Desarrollo RAGAS	40
Documentación	126
Total	**456**

Cuadro 1.1: Resumen total de horas

- **Estudio y recopilación de información**: Investigación sobre sistemas RAG, artículos científicos y documentación técnica para comprender el estado del arte.
- **Aprendizaje y pruebas RAG**: Experimentación con modelos de RAG y evaluación de su rendimiento.
- **Desarrollo del sistema RAG**: Implementación de la arquitectura del sistema RAG, incluyendo la ingesta de documentos y generación de respuestas.
- **Aprendizaje y pruebas RAGAS**: Familiarización con la herramienta RAGAS y pruebas para medir la calidad de las respuestas generadas.
- **Elaboración de los experimentos**: Diseño de experimentos con preguntas y documentos para evaluar el rendimiento del sistema en distintos casos de estudio.
- **Desarrollo RAGAS**: Personalización e integración de RAGAS para la evaluación de los resultados.
- **Documentación**: Redacción de la memoria del proyecto, incluyendo la arquitectura, experimentos, análisis de resultados y conclusiones.

CAPÍTULO 2

ESTADO DEL ARTE

En este capítulo se presenta una revisión del estado actual de la investigación y el desarrollo en el área de la IA Generativa, concretamente los RAGs. Se abordan definiciones clave, el panorama actual del conocimiento técnico, aspectos técnicos relevantes, casos de estudio y ejemplos prácticos, así como las tendencias actuales y futuras.

2.1. Definición

RAG es el proceso de optimización de la salida de los LLM de modo que haga referencia a una base de conocimientos autorizada fuera de los orígenes de datos de entrenamiento; antes de generar una respuesta [11]. Los LLM se entrenan con grandes volúmenes de datos y usan miles de millones de parámetros para generar resultados originales en tareas como responder preguntas, traducir idiomas y completar frases. Los RAG extienden las capacidades de los LLM a dominios específicos o a la base de conocimientos interna de una organización, todo ello sin la necesidad de volver a entrenar el modelo. Se trata de un método rentable para mejorar los resultados de los LLM de modo que sigan siendo relevantes, precisos y útiles en diversos contextos. [12]

Específicamente la función de un RAG se realiza en dos fases. En la primera fase, la de recuperación de información, funciona recuperando la información relevante de una base de datos vectorial. Este tipo de bases de datos vectoriales son frecuentes en sistemas basados en IA y están diseñadas para manejar vectores en vez de tablas; a diferencia de las bases de datos más comunes. Estos vectores representan una posición en el espacio vectorial, tal y como vemos en la figura 2.1, y se relacionan basándose en su similitud. Para crear dichos vectores, se utilizan los conocidos como modelos de *embeddings* que a partir de fragmentos de texto crea un vector que captura su significado semántico.

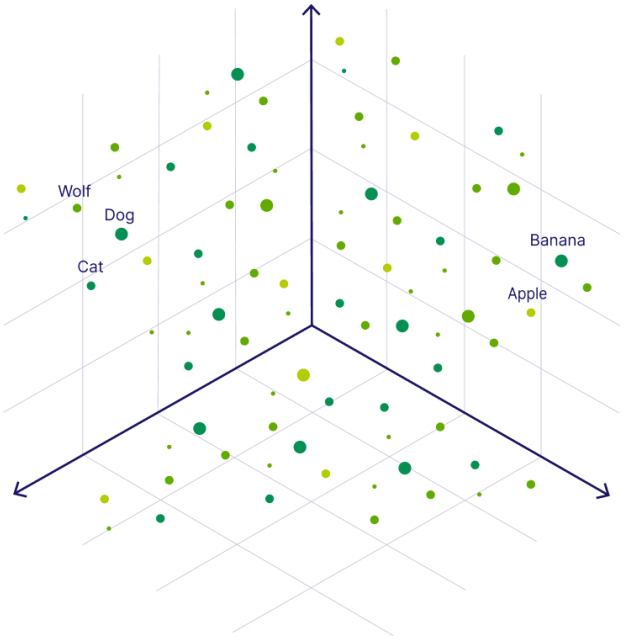

Figura 2.1: Representación vectorial de los datos de una base de datos vectorial [13]

De esta forma, las palabras que tengan mayor similitud, estarán más cerca una de otra. Una vez tenemos nuestra información vectorizada y almacenada, entramos en la segunda fase, donde entran en juego los LLM. Este tipo de modelos de IA están diseñados para comprender y generar texto de manera humana. Están previamente entrenados con enormes volúmenes de datos y a través de arquitecturas neuronales profundas, como los transformadores, aprenden patrones en el lenguaje que después pueden replicar. Un transformador es un tipo de red neuronal que está diseñada, principalmente, para el Procesamiento del Lenguaje Natural (PLN) [14].

2.2. Frontera del conocimiento técnico

La evolución de la IA Generativa ha estado marcada por avances significativos en los LLM. Sin embargo, estos modelos enfrentan limitaciones inherentes debido a su dependencia de datos de entrenamiento estáticos y su tendencia a generar respuestas que pueden ser inexactas o desactualizadas. En este contexto, los RAG han emergido como una solución innovadora que aborda estas limitaciones mediante la integración de fuentes de conocimiento externas autorizadas [15].

En esta sección, se exploran los aspectos positivos y negativos de la frontera del conocimiento técnico en torno a los sistemas RAG. Se discuten los beneficios que RAG aporta a la IA Generativa, como la implementación rentable, la capacidad de proporcionar información actualizada, el aumento de la confianza del usuario y el mayor control para los desarrolladores. Asimismo, se analizan los desafíos técnicos y operativos que aún persisten, como la complejidad en la integración, la necesidad de mantener datos actualizados, la dependencia de la calidad de las fuentes de datos o el riesgo de sobrecarga de información.

2.2.1. Aspectos positivos

La tecnología RAG aporta varios beneficios significativos a los esfuerzos de la IA generativa de una organización [12] [16]:

- **Implementación rentable:** El desarrollo de chatbots y otras aplicaciones de IA normalmente comienza con un modelo básico entrenado en datos generalizados. RAG ofrece un enfoque más rentable para introducir nuevos datos en el LLM, evitando los altos costos computacionales y financieros de volver a entrenar modelos desde cero.
- **Información actual:** Los RAG permiten a los desarrolladores proporcionar las últimas investigaciones, estadísticas o noticias a los modelos generativos. Con la capacidad de conectar el LLM a fuentes de información actualizadas, el modelo puede ofrecer respuestas precisas y contemporáneas.
- **Mayor confianza de los usuarios:** Al presentar información precisa con la atribución de la fuente, aumenta la confianza del usuario en las respuestas generadas. Los usuarios pueden verificar las fuentes y profundizar en los documentos originales si necesitan más detalles.

2.2.2. Aspectos negativos

A pesar de sus beneficios, RAG también presenta ciertos desafíos y limitaciones [17]:

- **Complejidad en la integración:** La integración de sistemas RAG con bases de datos y fuentes de información externa puede ser compleja y requerir un esfuerzo significativo en términos de ingeniería y mantenimiento.
- **Actualización constante de datos:** Mantener los datos externos actualizados es un desafío constante. Los documentos y la información deben ser actualizados regularmente para asegurar que el LLM recupere información precisa y relevante.
- **Dependencia de la calidad de los datos:** La efectividad depende, en gran medida, de la calidad y relevancia de las fuentes de datos externas. Si los datos son incorrectos, o de baja calidad, las respuestas del LLM también se verán afectadas.

2.3. Aspectos Técnicos del RAG

El diseño y la implementación de un RAG también presenta ciertos aspectos técnicos relevantes a tener en cuenta [18] [19]:

1. **Creación de datos externos:** Los datos externos se refiere a nuevos datos fuera del conjunto de datos de entrenamiento original del LLM. Estos pueden provenir de una API, de bases de datos o incluso de repositorios de documentos. Todos estos datos, de distintas fuentes, se pueden convertir en representaciones numéricas mediante modelos de incrustación de lenguaje, almacenándose en una base de datos vectorial.

2. **Recuperación de información relevante:** Utilizando representaciones vectoriales, el sistema realiza una búsqueda de relevancia para encontrar los datos más pertinentes en respuesta a una consulta del usuario. Esta relevancia se calcula mediante comparaciones matemáticas y representaciones vectoriales.

3. **Aumento de la solicitud del LLM:** El modelo RAG aumenta la entrada del usuario agregando los datos recuperados relevantes en el contexto. Técnicas de ingeniería de peticiones permiten que los LLM generen respuestas precisas a las consultas de los usuarios.

4. **Actualización de datos externos:** Para mantener la información actualizada, los documentos externos se actualizan periódicamente y sus representaciones incrustadas se revisan. Esto puede realizarse mediante procesos automatizados en tiempo real o mediante procesamiento por lotes.

2.4. Modelos de embeddings y LLM

Un RAG necesita utilizar dos tipos de modelos de redes neuronales. Por un lado, los modelos LLM para la generación de contenido y, por otro lado, los modelos de *embeddings*, para la generación de representaciones vectoriales de los datos.

Existen infinidad de modelos hoy en día, pero nos hemos decantado por los siguientes debido a varios factores clave. En primer lugar, su popularidad en la comunidad y su implementación en múltiples proyectos exitosos avalan su eficacia y fiabilidad. En segundo lugar, todos los modelos seleccionados son de código abierto, lo que permite un acceso y personalización sin restricciones; cruciales para la implementación de este proyecto. En tercer lugar, estos modelos han demostrado un rendimiento superior en tareas de procesamiento de lenguaje natural, asegurando tanto precisión como eficiencia.

Además, estos modelos también se han elegido por su alta escalabilidad, permitiendo adaptarse a diferentes volúmenes de datos y demandas de procesamiento; lo que resulta clave para garantizar el rendimiento del sistema RAG conforme este crece en complejidad y carga de trabajo. También se ha considerado su compatibilidad multilingüe, ya que todos los modelos están entrenados en múltiples idiomas, lo cual es fundamental para abordar las necesidades multilingües del sistema y finalmente por su extenso soporte y la documentación disponible. Junto con una comunidad activa de desarrolladores, facilitan la integración, optimización y soporte técnico necesario, asegurando la eficiencia y adaptabilidad del sistema a futuro.

Large Language Models

- **Llama 3 8B** [20]

  - **Características:** Llama 3 es la tercera generación de modelos de lenguaje de gran tamaño desarrollados por Meta. Este modelo ha sido preentrenado con ocho mil millones de parámetros y destaca por su capacidad de generar texto con una alta coherencia y relevancia; todo ello gracias a su entrenamiento en un vasto conjunto de datos multilingües y diversos. Incorpora

mejoras en la atención y la capacidad de comprensión contextual profunda, lo que le permite manejar tareas complejas de procesamiento de lenguaje natural con una alta precisión.

- **Enfoque:** Llama 3 está diseñado para aplicaciones que requieren una comprensión profunda y generación precisa de texto en múltiples idiomas, como traducción automática, generación de contenido y asistencia en la escritura.

- **Mistral 7B** [21]

  - **Características:** Mistral es un modelo de lenguaje desarrollado por la empresa Mistral. Este modelo ha sido preentrenado con siete mil millones de parámetros y emplea técnicas avanzadas de compresión y optimización de parámetros para ofrecer un rendimiento rápido sin sacrificar la calidad de las respuestas. También se distingue por su habilidad para integrarse fácilmente en sistemas existentes y su versatilidad en aplicaciones.
  - **Enfoque:** Mistral 2 se utiliza principalmente en aplicaciones donde la velocidad y la eficiencia son críticas, como en chatbots de atención al cliente, motores de búsqueda y asistentes virtuales en dispositivos móviles.

- **Phi3 3.8B** [22]

  - **Características:** Phi3 es un modelo de lenguaje avanzado desarrollado por Microsoft, conocido por su robustez en tareas de procesamiento de lenguaje natural. Este modelo ha sido preentrenado con tres mil ochocientos millones de parámetros y se beneficia de la extensa infraestructura de computación en la nube de Microsoft; lo que le permite escalar de manera eficiente para manejar grandes volúmenes de datos y consultas simultáneas. Phi3 integra mejoras en la comprensión semántica y generación de texto, optimizando la relevancia y precisión de las respuestas.
  - **Enfoque:** Phi3 está enfocado en aplicaciones empresariales y de investigación, incluyendo análisis de texto, generación de informes y soporte en la toma de decisiones basado en datos textuales.

- **Gemma 7B** [23]

  - **Características:** Gemma es un modelo de lenguaje de Google diseñado para ofrecer una alta precisión en la comprensión y generación de lenguaje natural. Este modelo ha sido preentrenado con siete mil millones de parámetros y destaca por su capacidad de aprendizaje continuo y su integración con la extensa base de datos y servicios de Google; permitiendo respuestas contextuales altamente precisas y actualizadas. Utiliza técnicas avanzadas de aprendizaje profundo para mejorar la coherencia y fluidez del texto generado.
  - **Enfoque:** Gemma se emplea en una variedad de aplicaciones que requieren interacciones complejas y personalizadas, como en Google Assistant, análisis de datos textuales y generación automática de contenido.

Modelos de *embeddings*

- **Multilingual Large** [24]
  - **Características:** El modelo intfloat/multilingual-e5-large, desarrollado por IntFloat, es un modelo de *embeddings* multilingüe que genera representaciones vectoriales de alta calidad para texto en múltiples idiomas. Este modelo es especialmente adecuado para tareas que requieren una comprensión precisa del contexto y significado en diversos idiomas.
  - **Enfoque:** El intfloat/multilingual-e5-large se enfoca principalmente en aplicaciones multilingües, como traducción automática, análisis de sentimientos multilingüe, y búsqueda semántica en bases de datos que contienen contenido en diferentes idiomas. Su capacidad para manejar múltiples lenguas lo hace ideal para empresas y organizaciones que operan a nivel global y necesitan procesar datos textuales en varios idiomas.
- **BAAI Large** [25]
  - **Características:** BAAI Large es un modelo de *embeddings* desarrollado por la Academia de Inteligencia Artificial de Beijing (BAAI) [26]. Este modelo está entrenado en un extenso conjunto de datos multilingües y se destaca por su capacidad para manejar tareas complejas de procesamiento de lenguaje natural en diversos idiomas. BAAI Large proporciona *embeddings* de alta calidad que capturan tanto el contexto local como global del texto.
  - **Enfoque:** BAAI Large se enfoca en aplicaciones multilingües y en análisis de texto a gran escala, como en plataformas de medios sociales, sistemas de traducción automática y análisis de sentimientos.
- **MXBAI Large** [27]
  - **Características:** El modelo mxbai-embed-large-v1, desarrollado por Mixedbread, es un potente modelo de *embeddings* diseñado para generar representaciones vectoriales de alta dimensión. Este modelo se destaca por su capacidad de capturar relaciones semánticas complejas y matices contextuales en grandes conjuntos de datos textuales.
  - **Enfoque:** El mxbai-embed-large-v1 se enfoca principalmente en aplicaciones que requieren una comprensión avanzada del lenguaje natural. Se utiliza en sistemas de recomendación, análisis de sentimientos, motores de búsqueda semántica y otras aplicaciones de procesamiento de lenguaje natural que necesitan una representación precisa y rica del texto.
- **BAAI Small** [28]
  - **Características:** BAAI Small es un modelo de *embeddings* más ligero desarrollado por la BAAI [26]. A diferencia de su hermano mayor, BAAI Small está optimizado para entornos con limitaciones de recursos, ofreciendo un equilibrio entre eficiencia y precisión en la generación de *embeddings*.
  - **Enfoque:** BAAI Small se utiliza en aplicaciones donde la eficiencia y el uso de recursos son cruciales, como en dispositivos móviles, aplicaciones de Internet de las Cosas (IoT) y sistemas de chat en tiempo real.

2.5. Tendencias actuales y futuras

En la actualidad, la adopción del RAG está en auge en diversas industrias, desde la salud y la educación hasta el comercio electrónico y los servicios financieros. Este crecimiento se debe a la necesidad apremiante de proporcionar información precisa y actualizada en tiempo real. Por ejemplo, en el sector de la salud, los profesionales pueden beneficiarse de respuestas rápidas y confiables a consultas médicas complejas, mejorando así la atención al paciente. En la educación, los estudiantes pueden acceder a información precisa y actualizada para sus estudios, mientras que en el comercio electrónico, los chatbots mejorados con RAG pueden ofrecer respuestas precisas sobre productos y servicios, mejorando la experiencia del cliente. La integración de RAG con tecnologías emergentes, como la búsqueda semántica y la computación en la nube, está potenciando aún más su capacidad de recuperación y procesamiento de información. La búsqueda semántica, en particular, permite a los RAG escanear extensas bases de datos de información dispar y recuperar datos con mayor precisión, mejorando así la calidad de las respuestas generativas. Por otro lado, la computación en la nube proporciona la infraestructura necesaria para manejar grandes volúmenes de datos y realizar cálculos complejos de manera eficiente, haciendo que las soluciones RAG sean más accesibles y escalables [29] [30].

Otra tendencia notable es la mejora en las técnicas de ingeniería de peticiones, que están optimizando la manera en que los modelos de lenguaje de gran tamaño (de las siglas en inglés LLM) interpretan y responden a las consultas de los usuarios. Estas mejoras permiten que los LLM generen respuestas más precisas y relevantes, aumentando la confianza y satisfacción de los usuarios. La ingeniería de peticiones avanzada facilita la comunicación efectiva con los LLM, asegurando que las respuestas generadas sean coherentes y pertinentes en diversos contextos [30].

Mirando hacia el futuro, se anticipa un desarrollo significativo de modelos de lenguaje de gran tamaño más especializados y modulares [31], para dominios específicos. Estos modelos especializados utilizarán RAG para mejorar aún más la precisión y relevancia de las respuestas generadas, adaptándose mejor a las necesidades particulares de cada industria. Por ejemplo, en el sector financiero, los LLM especializados podrán ofrecer asesoramiento financiero preciso y personalizado, mientras que en la investigación científica, podrán proporcionar análisis detallados y actualizados basados en las últimas investigaciones y datos disponibles.

La automatización avanzada para la actualización de datos externos también se perfila como una tendencia clave. Mantener los datos actualizados es esencial para la eficacia de los sistemas RAG, y la automatización permitirá realizar este proceso de manera continua y en tiempo real. Esto asegurará que los modelos de lenguaje siempre tengan acceso a la información más reciente, mejorando la precisión y relevancia de las respuestas. La automatización también reducirá la carga de trabajo manual asociada con la actualización de datos, permitiendo a los desarrolladores centrarse en mejorar otras áreas del sistema. Además, se espera que las técnicas para garantizar la calidad y relevancia de los datos externos continúen evolucionando. La gestión de la calidad de los datos es crucial para la eficacia de RAG, ya que la precisión de las respuestas genera-

das depende directamente de la calidad de las fuentes de datos. Futuras innovaciones en esta área incluirán mejores métodos para verificar y validar la información, así como técnicas avanzadas de filtrado y limpieza de datos. Estas mejoras garantizarán que los LLM recuperen y utilicen solo la información más precisa y relevante, aumentando así la confianza de los usuarios en las respuestas generadas por la IA [29].

En resumen, los RAG está transformando el campo de la IA generativa, ofreciendo soluciones innovadoras y efectivas en una variedad de sectores. Con la continua evolución y adopción de esta tecnología, es probable que veamos aplicaciones aún más avanzadas y especializadas, así como mejoras significativas en la calidad y relevancia de la información proporcionada por los LLM.

CAPÍTULO 3

ARQUITECTURA Y DISEÑO DEL RAG

En este capítulo se va a analizar las distintas tecnologías utilizadas para la implementación del RAG y el entorno de trabajo diseñado para garantizar un correcto funcionamiento de la herramienta. Todo el trabajo realizado se encuentra publicado en un repositorio de GitHub[1] [33] de forma pública para el fácil acceso a cualquier persona interesada.

3.1. Tecnologías

Para el desarrollo de este proyecto se han utilizado distintas tecnologías seleccionadas para aprovechar el mayor rendimiento de las mismas. Mayoritariamente, todas están relacionadas con el lenguaje de programación Python. Esto se debe a que este lenguaje se ha hecho muy popular entre los desarrolladores de software centrado en la IA y nos ofrece infinidad de librerías que facilitan tanto el diseño, el desarrollo y la implementación.

3.1.1. Entorno de trabajo

Para el entorno de trabajo se ha escogido el Integrated Development Environment (IDE) Visual Studio Code de Microsoft. Este IDE nos ofrece una infinidad de extensiones que nos permiten maniobrar entre distintas tecnologías, en un mismo proyecto, de forma sencilla. Es *Open Source* y eso permite que la comunidad desarrolle constantemente mejoras y actualizaciones que facilitan estar a la orden del día.

Gestionar distintas versiones de Python puede ser una tarea compleja, es por eso que se ha utilizado la librería `Pyenv`, que permite intercambiar entre versiones de forma sencilla. Gracias a estos entornos virtuales, podremos establecer un espacio hermético

[1] GitHub es una plataforma de desarrollo colaborativo basada en la web que utiliza el sistema de control de versiones Git [32] y uno de los sitios más populares para publicar proyectos de software.

en el cual se instalen las versiones, tanto de Python como de sus librerías, que nosotros deseemos. El objetivo de esta implementación es reducir las incompatibilidades y facilitar el desarrollo del proyecto.

3.1.2. Python y librerías

Para el correcto desarrollo del proyecto, se debe elegir una versión de Python y sus respectivas librerías. En este caso se ha escogido la versión 3.12.2, que es la última versión estable. Las librerías implicadas tienen total compatibilidad con esta versión y en los proyectos de software se suele especificar las versiones de estas librerías en un archivo llamado `requirements.txt` disponible en la raíz del repositorio del proyecto [33].

A continuación listamos las principales librerías relacionadas con nuestro proyecto de desarrollo y evaluación del RAG:

- `langchain-community`

  Se trata de una librería que forma parte de la popular librería `langchain`, ampliamente utilizada para construir aplicaciones basadas en LLM. Esta librería nos dará acceso a integraciones con distintas herramientas (sublibrerias) como:

  - **`PyPDFDirectoryLoader`:** Para la carga de documentos en formato PDF y transformación a texto plano.
  - **`RecursiveCharacterSplitter`:** Para realizar la división en *chunks* para su posterior inserción en la base de datos vectorial.
  - **`Chroma`:** Acceso a la tecnología ChromaDB, la cual ofrece un servicio de gestión bases de datos vectoriales a través de esta librería que conecta directamente con su Aplication Programming Interface (API).

- `langchain-huggingface`

  Se trata de una integración de la plataforma HuggingFace [2] y permite acceder a los distintos modelos que ofrece como servicio la plataforma:

  - `HuggingFaceEmbeddings`: Para acceder a los modelos de *embeddings* que se encuentran en la plataforma.

- `Ollama`

  Se trata de la librería que permite la integración del servicio de modelos de LLM que ofrece Ollama [35].

- `streamlit`

  Esta librería nos permite crear interfaces gráficas a base de módulos ya preestablecidos, de forma muy sencilla. Por tanto, Streamlit [36] es de gran ayuda para tener cubierta la parte de diseño gráfico e interacción con el usuario, para centrarse en los aspectos más técnicos.

[2] Esta plataforma es conocida por albergar un inmenso repositorio de modelos de IA/ML y distintos proyectos relacionados con estas tecnologías [34].

- `ragas`

  Esta herramienta es la que va a permitir evaluar de forma exhaustiva el proyecto. Mediante una serie de preguntas, contextos y respuestas correctas, esta librería será capaz de sacar distintas métricas que nos permitirán comprender el rendimiento del RAG en distintos casos de estudio.

Es importante también tener en cuenta que todas estas herramientas se ejecutan en nuestro entorno, con la finalidad de tener un mayor control y seguridad sobre estas.

3.2. Arquitectura e implementación

La arquitectura de un sistema RAG se basa en la combinación de dos componentes principales: el recuperador (*retriever*) y el generador (*generator*). Estos componentes trabajan juntos para producir respuestas precisas y contextualizadas basándose en una consulta dada. Si nos fijamos en la figura 3.1 la parte superior del diagrama se reflejan los pasos referentes a esta primera parte de recuperación de datos y la parte inferior los asociados a la generación de las correspondientes respuestas. Es necesario mencionar que se han incluido en el diagrama, las herramientas utilizadas, así como: *HuggingFace*, *ChromaDB*, *LangChain* o *Ollama*. Estas son las utilizadas en este proyecto, pero podrían utilizarse otros servicios sin modificar la arquitectura general presentada.

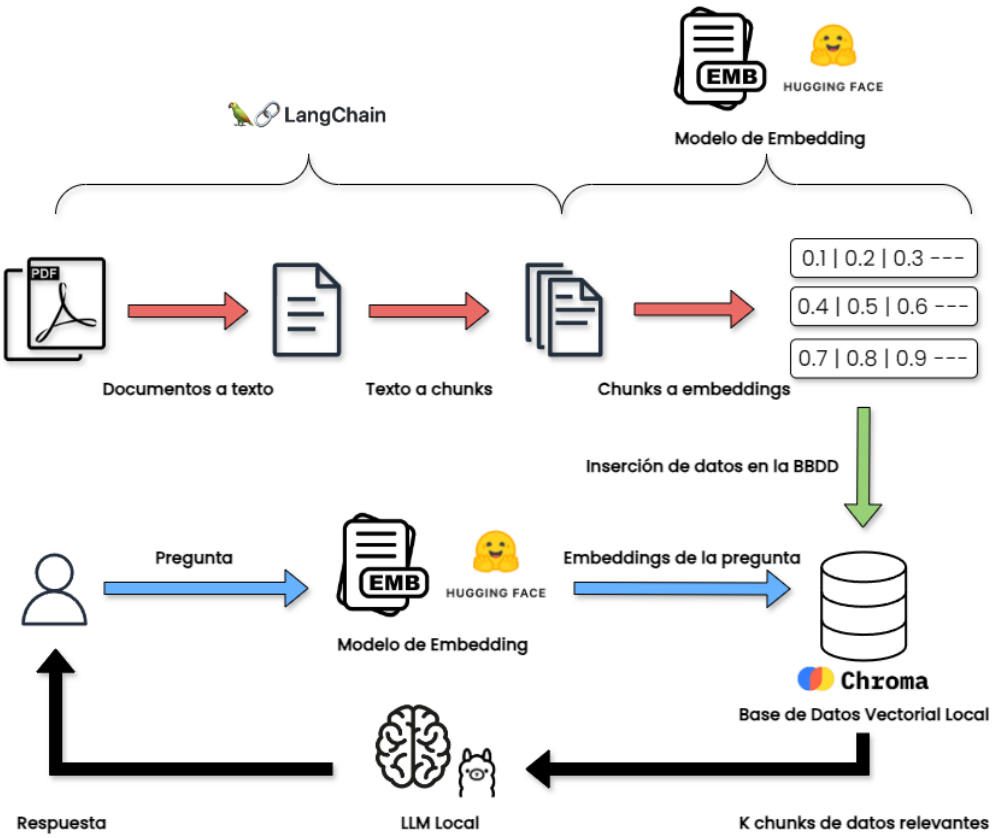

Figura 3.1: Arquitectura de un RAG

3.2.1. Recuperación de la información

Para la extracción inicial de la información de los documentos PDF, se ha utilizado la librería `PyPDFDirectoryLoader`. Esta nos permite cargar todos los PDF de una ruta preestablecida de forma sencilla y convertirlo a texto plano, tal y como observamos en la figura 3.2. Una vez tenemos este texto plano, mediante la función `RecursiveCharacterTextSplitter` podemos dividir el texto inicial *chunks* de un cierto tamaño, en nuestro caso de 512 caracteres. Es importante saber en qué tamaños trabajan los modelos de *embeddings* que se utilicen, ya que será esencial para crear estos *chunks*. Ya que el tamaño de los *chunks* debe ser parejo al que usan los modelos. De esta forma, dichos modelos podrán buscar de forma eficiente la información relevante necesaria. Ahora estos *chunks* podemos incorporarlos a nuestra base de datos vectorial de Chroma.

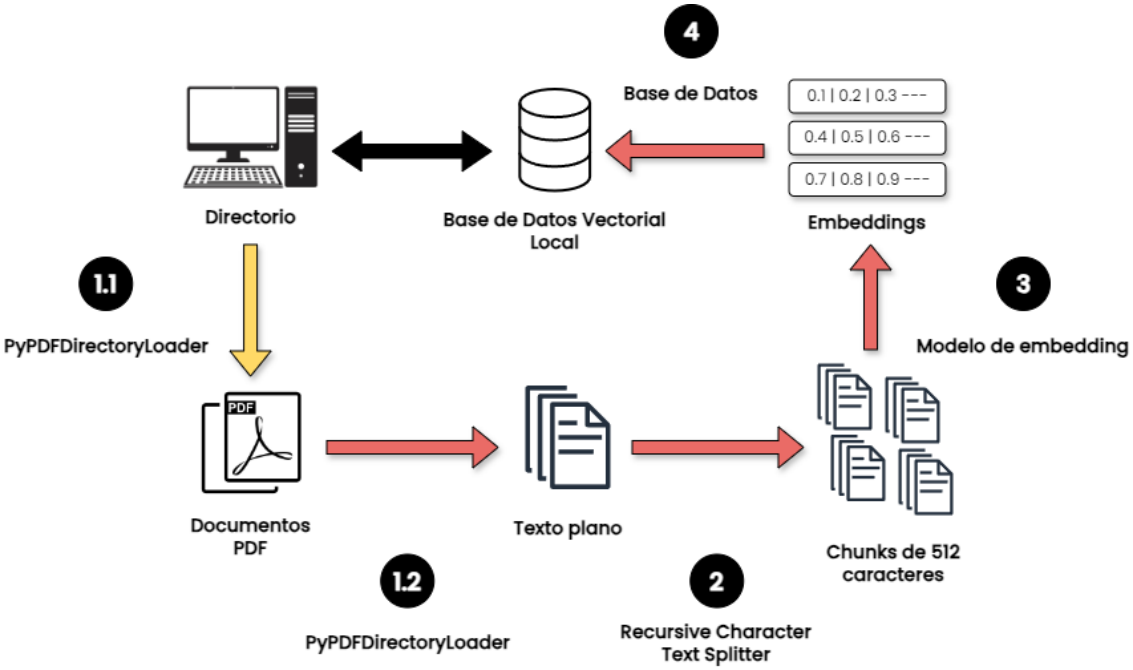

Figura 3.2: Proceso de vectorización de los documentos

Además, previamente se verifica si dichos *chunks* ya aparecen en la base de datos. En caso afirmativo, se descarta la carga de datos a la base de datos. Esto es importante, ya que de esta forma optimizamos el proceso de carga inicial de datos para no tener elementos duplicados; que suele tener mucho coste computacional.

3.2.2. Generación de la respuesta

En esta fase nos disponemos, que podemos observar en la parte inferior de la figura 3.1, nos disponemos a registrar la pregunta del usuario y buscar similitudes en nuestra base de datos; a modo de contexto. Para llevar a cabo esto, se debe buscar los *chunks* con mayor similitud a la pregunta formulada en la base de datos vectorial. Esto se realiza con el método `similarity_search_with_score()` al cual le pasamos por pa-

rámetro la pregunta realizada por el usuario y el número de similitudes que queremos que nos devuelva. A continuación se escoge el *prompt* dependiendo del idioma que haya escogido el usuario y se llama al modelo LLM, que deberá estar ejecutándose en local, en cuestión para realizar la consulta. Un *prompt* es básicamente el texto inicial que se le introduce al modelo para enfocar su generación de respuestas.

Una vez realizados todos estos pasos, el usuario recibe la respuesta generada por el modelo de LLM y se da el proceso por finalizado hasta que el usuario vuelva a realizar otra pregunta.

3.3. Herramientas y métricas para la evaluación

La librería `RAGAS`[37] de Python nos permite realizar diversas pruebas a los diferentes modelos y obtener ciertas métricas, que se explican con más detalle en la siguiente subsección, sobre los diferentes procesos que lleva a cabo el RAG. En particular, obtendremos las métricas mostradas en la figura 3.3, que nos permitirán evaluar dos aspectos clave:

- **Generación de contenido**: Mediremos el rendimiento de los LLM en la creación de respuestas.
- **Recopilación de información relevante**: Evaluaremos la efectividad de los *embeddings* en la extracción de datos relevantes de la base de datos.

3.3.1. Métricas utilizadas

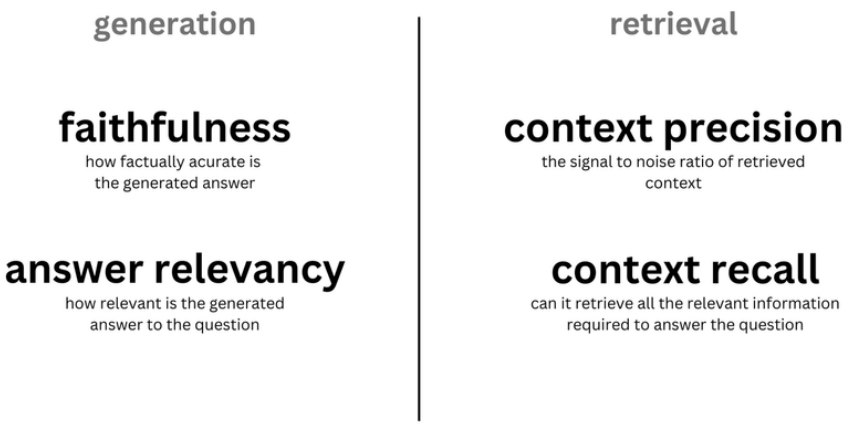

Figura 3.3: Métricas de RAGAS en inglés

Para entender un poco mejor que valor nos aportara cada métrica, es necesario realizar una breve explicación de cada una de ellas.

- **Fidelidad (*Faithfulness*)**
  Evalúa la exactitud de la información de la respuesta generada en relación con el contexto proporcionado. Se calcula comparando la respuesta con el contexto recuperado, y se expresa en una escala de 0 a 1, donde un valor más alto indica una mejor coherencia. Una respuesta generada se considera de alta calidad si todas sus afirmaciones pueden ser derivadas del contexto dado. Para calcular esta métrica, primero se identifica un conjunto de afirmaciones en la respuesta generada. Luego, cada una de estas afirmaciones se verifica contra el contexto proporcionado para determinar si puede deducirse de él. La puntuación de fidelidad se determina de la siguiente manera:

$$\text{faithfulness} = \frac{|\text{Número de afirmaciones deducibles del contexto}|}{|\text{Número total de reclamaciones en la respuesta generada}|}$$

- **Relevancia de la respuesta (*Answer relevancy*)**
  Esta métrica se enfoca en evaluar la adecuación de la respuesta generada en relación con la pregunta planteada. Se otorgan puntuaciones más bajas a respuestas que son incompletas o que contienen información redundante, mientras que puntuaciones más altas indican una mayor relevancia. La métrica se calcula tomando en cuenta la **pregunta**, el **contexto** y la **respuesta**. La relevancia de la respuesta se determina como la similitud promedio del coseno entre la pregunta **original** y un conjunto de preguntas artificiales generadas (mediante ingeniería inversa) a partir de la **respuesta**:

$$\text{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N}\cos(E_{g_i}, E_o)$$

$$\text{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N}\frac{E_{g_i}\cdot E_o}{||E_{g_i}||\,||E_o||}$$

  Donde:

  - E_{g_i} es el *embedding* de la pregunta generada n.º i.
  - E_o es el *embedding* de la pregunta original.
  - N es el número de preguntas generadas, que por defecto es $N = 3$.

  Es importante destacar que, aunque en la práctica la puntuación suele estar entre 0 y 1, esto no está garantizado matemáticamente debido a que la similitud del coseno puede variar entre -1 y 1.

- **Precisión del contexto (*Context precision*)**
  La precisión del contexto es una métrica que evalúa si los elementos relevantes de la verdad fundamental presentes en los contextos están correctamente ordenados. El objetivo es que todos los fragmentos relevantes aparezcan en las primeras posiciones. Esta métrica se calcula utilizando la pregunta, la verdad fundamental

y los contextos, con valores que van de 0 a 1, donde las puntuaciones más altas indican una mayor precisión.

$$\text{Contex Precision@K} = \frac{\sum_{k=1}^{K}(Precision@k \times v_k)}{\text{Número total de artículos relevantes en los primeros K resultados}}$$

$$\text{Precision@k} = \frac{\text{verdaderos positivos@k}}{\text{verdaderos positivos@k + falsos psoitivos@k}}$$

Donde K es el número total de *chunks* en el contexto y $v_k \in \{0,1\}$ es el indicador de relevancia en el rango k.

- **Recuperación del contexto (*Context recall*)**
  La recuperación del contexto mide el grado en que el contexto recuperado coincide con la respuesta anotada, considerada como la verdad fundamental. Los valores varían entre 0 y 1, con valores más altos indicando un mejor rendimiento. Para calcular la recuperación del contexto en relación con la respuesta verdadera, se examina cada afirmación de la respuesta para determinar si puede atribuirse al contexto recuperado. En un escenario ideal, todas las afirmaciones de la respuesta verdadera deberían estar respaldadas por el contexto recuperado.

  La fórmula para calcular la recuperación del contexto es la siguiente:

$$\text{context recall} = \frac{|\text{Reclamaciones GT que pueden atribuirse al contexto}|}{|\text{Número de reclamaciones en GT}|}$$

CAPÍTULO 4

EXPERIMENTACIÓN

En este capítulo se plantean tres casos de estudio distintos, los cuales permitirán analizar y validar el funcionamiento de nuestra implementación de RAG. Estos casos de estudio se deben principalmente al tipo de contenido que pueden albergar los documentos en formato PDF.

El primer caso de estudio es el de documentos científico-académicos. En este esperamos observar la capacidad de extracción de datos del modelo, así como fórmulas o descripciones técnicas. También la capacidad de comprensión de dichos argumentos técnicos y por último su capacidad de síntesis. El segundo caso de estudio es el de documentos o informes financieros. En este esperamos analizar la capacidad de reconocimiento de cifras y los resultados económicos relacionados. Además de la capacidad de comprensión de dichos resultados para incluso determinar ciertas tendencias o relaciones en los mismos. El tercer y último caso de estudio es el de documentos legales. En este caso tenemos como objetivo evaluar la capacidad de reconocimiento de elementos clave o cláusulas contractuales. Asimismo, analizar la capacidad de explicación por parte del modelo a un lenguaje más coloquial de los aspectos legales técnicos.

Para llevar a cabo los distintos gráficos de este capítulo, se han utilizado diferentes scripts [33]. Por un lado, un programa que se encarga de generar los CSV utilizando las herramientas de evaluación de RAGAS y, por otro lado, los que, a partir de estos archivos CSV, crean los gráficos a analizar. De esta forma se tiene un mayor control de los datos generados y la visualización de los mismos.

4.1. Datos utilizados

El programa utiliza las herramientas de RAGAS, a partir de los documentos y las preguntas/respuestas/contextos, calcula y extrae las métricas. En esta herramienta, las preguntas y respuestas (previamente diseñadas) se almacenan en un archivo JSON, con la siguiente estructura, para una mayor manejabilidad de las mismas:

```
{
    "testX": {
        "cat": [
            "Pregunta 1",
            "Pregunta 2",
            "Pregunta n"
        ],

        "en": [
            "Pregunta 1",
            "Pregunta 2",
            "Pregunta n"
        ],

        "es": [
            "Pregunta 1",
            "Pregunta 2",
            "Pregunta n"
        ]
    },
}
```

Cada caso de estudio consta de seis documentos, dos para cada idioma, lo que nos da un total de dieciocho documentos. De estos documentos se realizan tres preguntas a cada uno de ellos (con sus respectivas respuestas). Esto hace que en cada test, se hagan seis preguntas en cada idioma; que supone un total de dieciocho preguntas por test. Dichas preguntas se han generado mediante el uso de la herramienta ChatGPT de OpenAI [7], gracias a que se le pueden pasar documentos PDF y la misma herramienta genera diversas preguntas. Después de que genere diversas preguntas, se han seleccionado las más interesantes para analizar los modelos. Por ejemplo, en el caso de documentos jurídicos, preguntas que tuvieran relevancia en cuanto a comprensión de leyes o derechos. Por otro lado, para los documentos financieros, preguntas que reflejen el objetivo de los informes, análisis de resultados, incluso evaluación del impacto de ciertas actividades. Por último, en los documentos científico-técnicos, las preguntas están enfocadas a extraer los datos clave de los casos de estudio y comprensión de ciertas fórmulas.

A continuación se muestran los documentos utilizados (que se pueden encontrar en el repositorio del proyecto [33]) con una breve explicación de su contenido y algunas preguntas, con sus respectivas respuestas, realizadas:

- **Caso de estudio 1 - Preguntas y respuestas para documentos jurídicos**

  - **Documento 5:** *Constitución Española*
    Este documento establece la estructura política y legal de España, describe los derechos fundamentales de los ciudadanos, y define el funcionamiento de las instituciones del Estado. Se detallan las competencias de las comunidades autónomas y las leyes sobre temas como la propiedad, la educación y la libertad de expresión.

    - **¿Qué derechos lingüísticos reconoce la Constitución Española?**
      La Constitución Española reconoce el castellano como lengua oficial del Estado y establece que todos los españoles tienen el deber de conocerla y el derecho a usarla. También reconoce que las demás lenguas españolas serán oficiales en las respectivas Comunidades Autónomas de acuerdo con sus Estatutos, y protege la riqueza de las distintas modalidades lingüísticas de España como un patrimonio cultural que debe ser objeto de especial respeto y protección.
    - **¿Qué establece la Constitución Española sobre la detención preventiva y los derechos de las personas detenidas?**
      La Constitución Española establece que la detención preventiva no puede durar más del tiempo estrictamente necesario para la realización de las averiguaciones, y en todo caso, el detenido debe ser puesto en libertad o a disposición de la autoridad judicial en un plazo máximo de 72 horas. Además, garantiza el derecho a ser informado de forma inmediata y comprensible sobre los derechos y las razones de la detención, así como el derecho a la asistencia de un abogado.

  - **Documento 4:** *United Kingdom Constitution*
    Este documento aborda la constitución no escrita del Reino Unido, comparando alternativas para su codificación. Expone los principios democráticos, derechos fundamentales, y la estructura de las instituciones como el parlamento, el primer ministro, y la monarquía.

    - **What are the conditions under which an employment contract's probationary period may be extended or terminated?**
      The probationary period may be extended or terminated if the employee's work performance is not up to the required standard or if the working relationship is not agreeable. Both parties are required to give one week's written notice during this period.
    - **What are the stipulations for holiday entitlement and the conditions under which holiday can be carried over to the next year?**
      The holiday entitlement is 5.6 weeks per year, pro-rata for part-time employees. Holidays must be taken within the year, and carrying over untaken holidays requires prior agreement, except in cases of sickness preventing the leave.

  - **Documento 2:** *Contracte indefinit - Govern d'Espanya*
    Este es un contrato de trabajo indefinido que define los términos entre empleador y empleado, incluyendo salario, horas de trabajo, y beneficios.

También regula temas de seguridad social, vacaciones, y bonificaciones específicas para ciertos colectivos.

- **Com es defineix i s'estableix la durada del període de prova en un contracte indefinit segons el document?**
  La durada del període de prova en un contracte indefinit es defineix d'acord amb el que estableix l'Estatut dels Treballadors, respectant les normes legals vigents. Pot variar segons el grup professional o el nivell del treballador, i en alguns casos pot ser de fins a un any si s'acull a certes disposicions legals.
  Quins drets de bonificació a la Seguretat Social es mencionen per a contractes indefinits celebrats amb treballadors pertanyents al Sistema Nacional de Garantia Juvenil?
  Per als contractes indefinits celebrats amb treballadors pertanyents al Sistema Nacional de Garantia Juvenil, es mencionen bonificacions de 300 euros mensuals en la cotització empresarial a la Seguretat Social durant un període de 6 mesos. En el cas de contractes a temps parcial, la bonificació s'ajusta proporcionalment al percentatge de la jornada.

- **Caso de estudio 2 - Preguntas y respuestas para documentos financieros**

  - **Documento 11:** *Informe financiero Banco de España*
    Este documento presenta la evolución financiera del Banco de España, haciendo énfasis en la política monetaria y la supervisión bancaria. Detalla los ingresos por intereses, inversiones y reservas. Además, se explora el impacto de las políticas del BCE en la estabilidad financiera nacional.

    - **¿Cómo se ha comportado la inflación en España durante 2023 y cuáles son las perspectivas para los próximos años?**
      Durante 2023, la inflación en España continuó una moderación más intensa de lo esperado, impulsada principalmente por una evolución favorable de los precios de la energía. Se prevé que la inflación siga desacelerándose gradualmente en los próximos trimestres.
    - **¿Cómo se describe el proceso de envejecimiento poblacional en España y cuáles son sus implicaciones económicas?**
      El envejecimiento poblacional en España se caracteriza por una baja tasa de fecundidad y una alta esperanza de vida, lo que supone un desafío para el mercado laboral y las finanzas públicas debido al incremento del gasto en pensiones y sanidad. Se requieren políticas para promover la prolongación de la vida laboral y mejorar la formación continua.

  - **Documento 10:** *Financial report of NTT DATA*
    El informe trimestral de NTT Data para el periodo que finalizó en junio de 2024, que muestra los resultados de los diferentes ingresos operativos, en comparación al año anterior. Además, se revisa el desempeño en los mercados globales y las estrategias para aumentar la eficiencia.

    - **How did NTT Data's consolidated financial results for the fiscal year ending March 31, 2025, compare to the previous fiscal year?**

NTT Data's consolidated financial results forecast for the fiscal year ending March 31, 2025, includes expected operating revenues of ¥13,460,000 million, a 0.6 % increase, and a profit attributable to NTT of ¥1,100,000 million, a 14.0 % decrease compared to the previous fiscal year.

- **What were the trends in NTT Data's cash flows from operating, investing, and financing activities during the three months ended June 30, 2024?**
  During the three months ended June 30, 2024, NTT Data's cash flows from operating activities decreased by ¥145,742 million compared to the previous year, while cash used in investing activities increased by ¥135,046 million, and cash provided by financing activities increased by ¥193,981 million.

- **Documento 8:** *Informe financer Generalitat de Catalunya*
  Describe los ingresos y gastos del gobierno catalán, destacando los sectores de educación, salud y servicios sociales como principales áreas de inversión. Se incluyen datos sobre endeudamiento público y financiación recibida por parte de fondos estatales y europeos.

  - **Com afecta la capacitat fiscal a la distribució de recursos entre comunitats autònomes?**
    La capacitat fiscal determina que les comunitats autònomes amb menor capacitat rebin transferències positives per equilibrar els recursos necessaris per finançar els serveis públics fonamentals, mentre que aquelles amb major capacitat han de fer aportacions a les altres comunitats.",
  - **Quin impacte han tingut les transferències estatals en els ingressos no financers de la Generalitat de Catalunya el 2022?**
    Les transferències estatals van tenir un gran impacte el 2021 a causa de la despesa Covid, però el 2022, la seva reducció ha fet que els ingressos no financers tornin a dependre més del model de finançament autonòmic tradicional.

- **Caso de estudio 3 - Preguntas y respuestas para documentos científico-técnicos**

  - **Documento 18:** *Artículo sobre el uso del algoritmo k-means*
    El artículo describe el uso del algoritmo K-means para clasificar perfiles de consumo de energía en clientes residenciales usando datos de medidores inteligentes. El estudio muestra cómo esta clasificación puede ayudar a mejorar las políticas públicas y la eficiencia en la distribución de energía.

    - **¿Qué es el algoritmo *K-means* y cómo se utiliza en el contexto del análisis de consumo energético?**
      El algoritmo *K-means* es un método de agrupamiento no jerárquico que particiona un conjunto de datos en un número específico de grupos (clústeres) basándose en las características compartidas. En el contexto del análisis de consumo energético, *K-means* se utiliza para identificar patrones de consumo similares entre diferentes clientes, permitiendo una mejor gestión y optimización de recursos energéticos.

- **¿Cómo se determina el número óptimo de clústeres en un análisis con *K-means*?**
  El número óptimo de clústeres en un análisis con *K-means* se determina mediante la evaluación de la precisión y la calidad de los clústeres utilizando medidas proporcionadas por la teoría de los conjuntos aproximados (RST), así como mediante la ejecución repetida del algoritmo con diferentes particiones iniciales.

- **Documento 15:** *Einstein Relativity Paper*
  Un artículo académico que presenta algunas de las contribuciones más relevantes de Albert Einstein en el campo de la física, como su teoría de la relatividad y su trabajo sobre la naturaleza de la luz y la energía. Se analiza el impacto de sus teorías en el desarrollo de la ciencia moderna y cómo sus descubrimientos siguen influyendo en diversas áreas de la investigación científica actual.

  - **Explain the mass-energy equivalence formula and its significance in special relativity.**
    The mass-energy equivalence formula is given by: $E = mc^2$, where E is the energy, m is the mass, and c is the speed of light. This formula signifies that mass and energy are interchangeable, and a small amount of mass can be converted into a large amount of energy.
  - **How does special relativity reconcile with the principle of the constancy of the speed of light in a vacuum?**
    Special relativity reconciles with the principle of the constancy of the speed of light by postulating that the speed of light in a vacuum is the same for all observers, regardless of their motion relative to the light source. This leads to the need for a new understanding of space and time as being interwoven into a four-dimensional spacetime.

- **Documento 13:** *Article sobre els riscos dels medicaments en la insuficiència cardíaca*
  Detalla los tratamientos farmacológicos para la insuficiencia cardíaca, incluyendo inhibidores de la enzima convertidora de angiotensina (IECA), betabloqueadores, diuréticos y antagonistas de la aldosterona. Explica cómo funcionan los medicamentos, sus efectos secundarios y la importancia del cumplimiento del tratamiento para evitar el empeoramiento de la enfermedad.

  - **Quins són els principals medicaments que poden desencadenar o exacerbar la insuficiència cardíaca segons l'article?**
    Els principals medicaments que poden desencadenar o exacerbar la insuficiència cardíaca inclouen antiinflamatoris no esteroïdals (AINE), certs anestèsics, antiarrítmics, i alguns antidiabètics, entre d'altres, segons els mecanismes de toxicitat miocardíaca o disfunció cardíaca.
  - **Quins mecanismes estan implicats en la toxicitat miocardíaca induïda per certs fàrmacs?**
    Els mecanismes implicats en la toxicitat miocardíaca inclouen l'estrès oxidatiu, la inhibició de prostaglandines, la retenció de sodi i aigua, i la

depressió de la funció miocàrdica, entre d'altres. Aquests mecanismes poden variar segons el fàrmac i la seva acció específica sobre el cor.

La totalidad de preguntas y respuestas se encuentran disponibles en el repositorio del proyecto [33] en formato Markdown.

A continuación, el programa busca tantos contextos como se le indique (en nuestro caso tres) que, por similitud, tengan sentido con la pregunta que se hace, y se crea un conjunto de datos (más conocido como *Dataset*) que contiene las preguntas, las respuestas correspondientes y los distintos contextos (que sería, en este caso, la verdad). Con todo esto, se ejecutan una serie de pruebas que realiza el propio RAGAS con cada modelo de LLM y obtenemos los resultados pertinentes; que se almacenaran en documentos CSV para su posterior procesado y visualización de los datos. Esto, a nivel de código, se vería aproximadamente, de la siguiente forma:

```
# Nuestro dataset con la información a evaluar
data_samples = {
    "question": questions,
    "answer": answers,
    "contexts": contexts,
    "ground_truth": answers
}

# Evaluamos con el llm, embedding y metricas pertinentes
score = evaluate(
    dataset,
    metrics=[faithfulness, answer_relevancy, context_precision,
     ↪  context_recall],
    llm=llm,
    embeddings=embeddings
)

# Guardamos los datos en formato CSV
df_score.to_csv("/ruta")
```

4.2. Estudio del modelo de *embeddings* óptimo

Con el fin de enfocar el análisis en las respuestas proporcionadas por los modelos de LLM, es interesante realizar una selección previa de los modelos de *embeddings* más óptimos para cada caso de estudio. Este enfoque se justifica porque, al generar una respuesta con un modelo LLM, es fundamental disponer de los contextos más relevantes y precisos. De este modo, los casos de estudio se llevan a cabo con una óptima selección de contextos, permitiendo así que el análisis se centre principalmente en la calidad de la generación de contenido, que es el aspecto que tiene un impacto más directo en el usuario final.

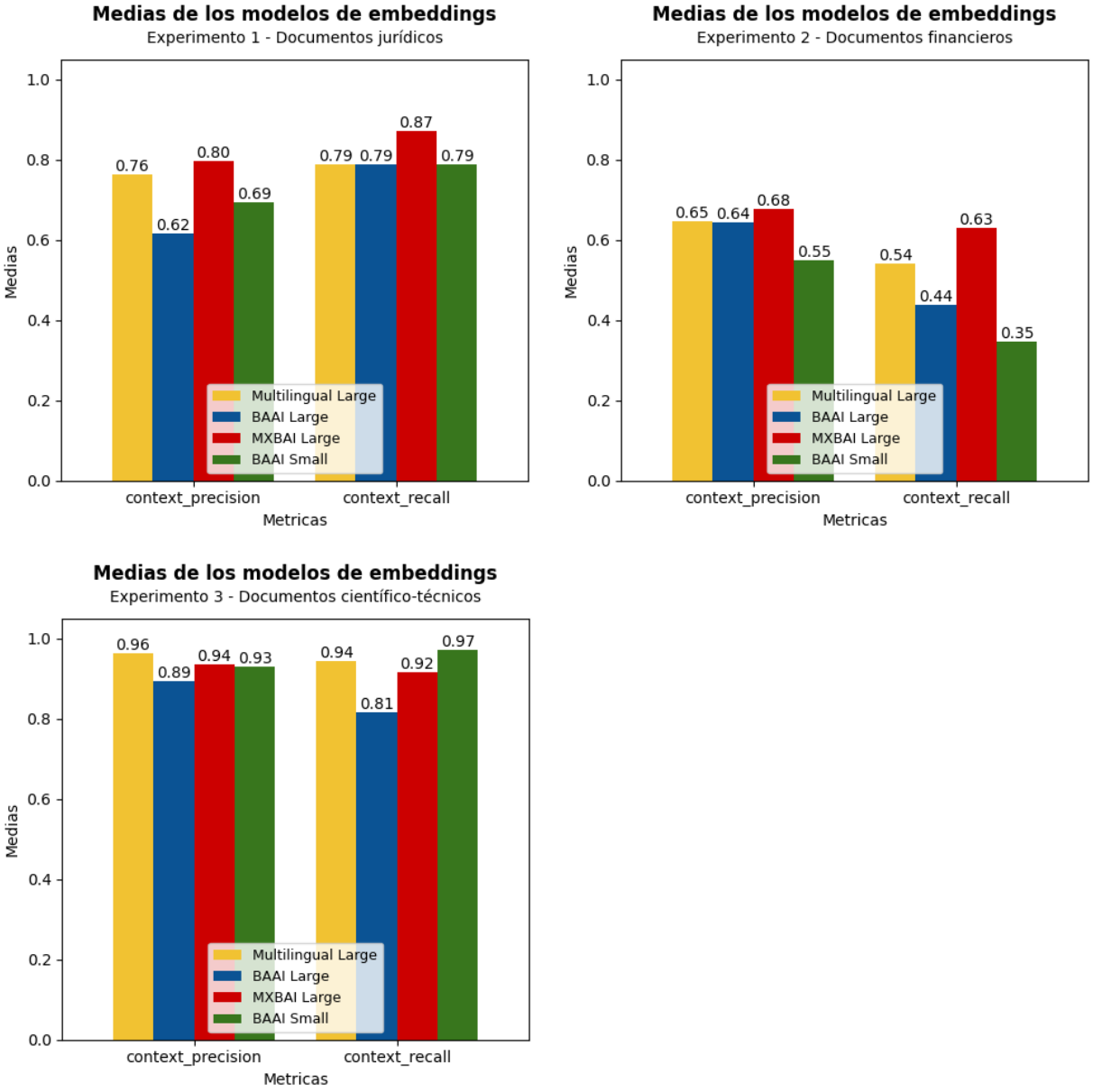

Figura 4.1: Medias modelos de *embeddings*

Al analizar las figuras 4.1 observamos los resultados de los tres casos de estudio por separado. En cada una de ellas, se mide la capacidad de precisión y recuperación de los contextos de los distintos modelos. Representados por distintos colores, a mayor valor de cada uno de estos resultados, mejor resultado y viceversa. Es por ello que, a grandes rasgos, observamos que este tipo de modelos ofrecen mejor rendimiento en los casos de estudio uno y tres, y peor para el caso de estudio con documentos financieros.

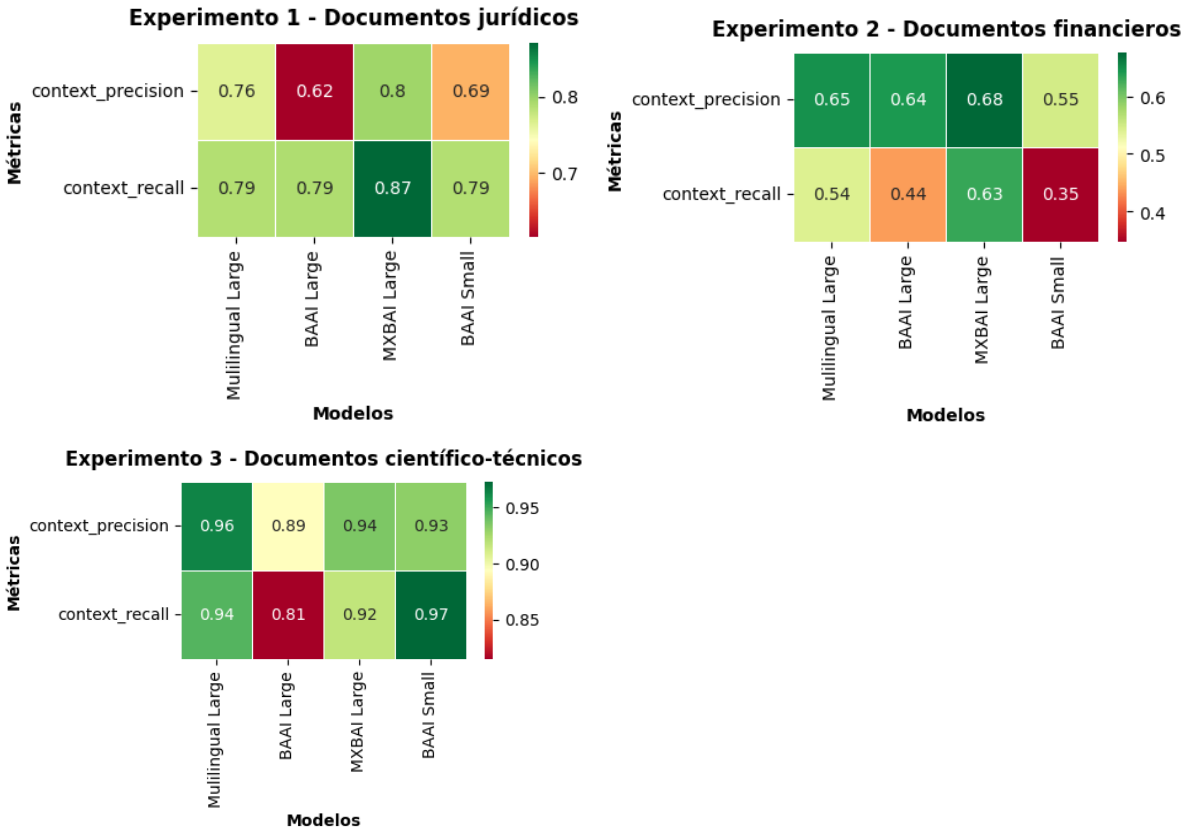

Figura 4.2: Mapas de calor de los modelos de *embeddings*

Si observamos con más detalle el mapa de calor de la figura 4.2 podemos ver cuál es el modelo que mejor resultados ofrece en cada caso de estudio. En los dos primeros es claro ganador el modelo `MXBAI Large` y en el tercer estudio, podría debatirse entre el `Multilingual Large` y otra vez el `MXBAI Large`. Sin embargo, la métrica de precisión de los contextos, es mayor en la del modelo `Multilingual Large` y al tratarse de documentos científico-técnicos, podría ser una mejor opción frente a los otros. Por tanto, nos quedaremos con los siguientes modelos, de cada caso de estudio, para realizar la evaluación de nuestro RAG:

- **Caso de estudio 1:** MXBAI Large
- **Caso de estudio 2:** MXBAI Large
- **Caso de estudio 3:** Multilingual Large

4.3. Primer caso de estudio: Documentos jurídicos

En esta sección se presentan los resultados obtenidos del primer estudio, centrado en documentos jurídicos. El objetivo principal es evaluar la capacidad del sistema RAG para identificar y extraer información relevante en contextos legales, así como su habilidad para generar respuestas precisas y coherentes a partir de dicho contenido.

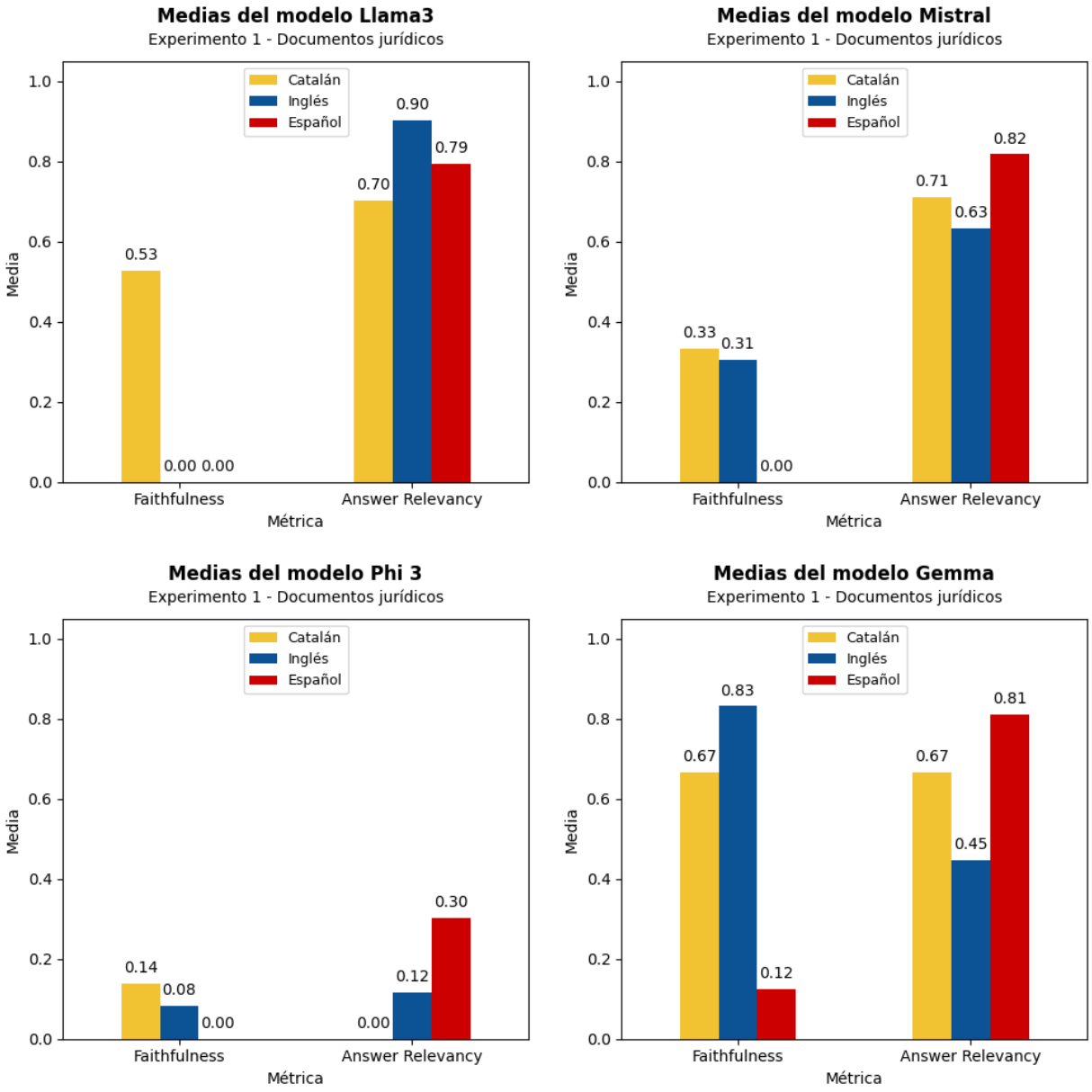

Figura 4.3: Medias modelos LLM del primer caso de estudio

Los gráficos presentados proporcionan una visión general del comportamiento del sistema al trabajar con documentos jurídicos, utilizando siempre el modelo de *embedding* `MXBAI Large` con los distintos modelos de LLM. En cuanto a las métricas

que nos ofrece la librería RAGAS, vemos que en se repite en todas las casuísticas que los LLM rinden bastante bien en cuanto a la relevancia de la respuesta se trata; pero no tanto con la fidelidad de la misma. El modelo que nos da, en general, mejores resultados es el modelo `Gemma` de Google.

Para tener otra herramienta que nos ayude a elegir que modelo es el más adecuado, se ha implementado una tabla con el promedio de los tres idiomas por cada variable. Además, se ha añadido una columna que muestra un nuevo índice, diseñado para evaluar qué modelo ofrece mejores resultados considerando ambas variables: *faithfulness* y *answer relevancy*.

Este índice se calcula realizando ahora la media armónica, ya que en este caso, las variables *faithfullnes* y *answer relevancy* son ratios. La fórmula, por tanto, será la siguiente:

$$\text{Índice General} = \frac{n}{\sum_{i=1}^{i} \frac{1}{x_i}}$$

Donde n es el número de elementos y x el valor de los mismos. Por ejemplo, para calcular la media armónica del modelo `Llama 3`, se realizaría de la siguiente forma:

$$\text{Índice General} = \frac{n}{\sum_{i=1}^{i} \frac{1}{x_i}} = \frac{2}{\frac{1}{0,18} + \frac{1}{0,80}} = 0,29$$

A continuación la tabla con todos los valores:

Modelo LLM	Faithfullnes	Answer Relevancy	Índice General
Llama 3	0,18	0,80	0,29
Mistral	0,21	0,72	0,33
Phi3	0,00	0,14	0,09
Gemma	**0,54**	**0,59**	**0,59**

Cuadro 4.1: Tabla de resultados del primer caso de estudio

Por tanto, en este estudio, el modelo que nos ofrece mejores resultados a nivel general es el modelo de Google `Gemma`.

4.4. Segundo caso de estudio: Documentos financieros

El segundo estudio se centra en documentos financieros, utilizando el modelo de *embedding* `MXBAI Large`; al igual que en el primer caso de estudio. El objetivo de este análisis es evaluar la precisión del sistema RAG en la identificación de datos numéricos, así como su capacidad para interpretar y relacionar estos datos en un contexto financiero. Los gráficos siguientes reflejan las métricas fundamentales obtenidas durante este estudio.

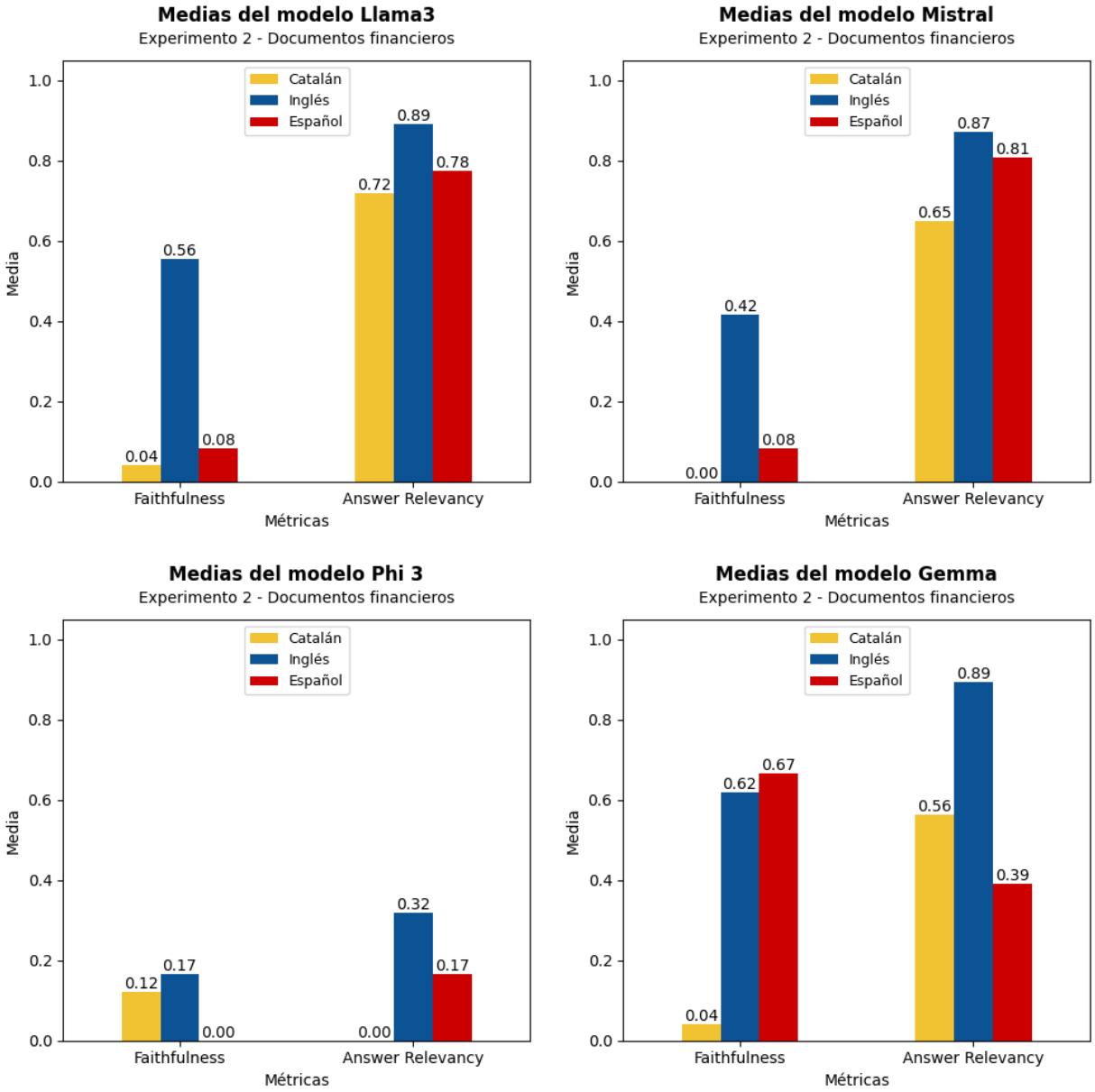

Figura 4.4: Medias modelos LLM del segundo caso de estudio

Los gráficos nos proporcionan una primera aproximación a la capacidad del sistema para manejar información financiera. Encontramos métricas con cierta similitud al caso de estudio anterior, pero debemos fijarnos detenidamente para escoger bien que modelo está funcionando mejor.

Modelo LLM	Faithfullnes	Answer Relevancy	Índice General
Llama 3	0,23	0,80	0,36
Mistral	0,17	0,78	0,28
Phi3	0,10	0,16	0,12
Gemma	**0,44**	**0,51**	**0,53**

Cuadro 4.2: Tabla de resultados del segundo caso de estudio

En este caso particular, a simple vista mirando las gráficas de la figura 4.4 podríamos decir que el modelo de Google podría ser otro buen candidato, ya que sobre todo en el idioma inglés nos proporciona los mejores resultados. Sin embargo, si observamos las medias del modelo `Llama 3` y `Mistral`, observamos que ofrecen incluso mejor resultado que el modelo `Gemma` en cuanto a *answer relevancy* se trata. No obstante, la media armónica favorece más a los modelos que presentan resultados menos dispares entre variables. Por tanto, llegamos a la conclusión que, en términos generales, el modelo `Gemma` ofrece mejores resultados.

4.5. Tercer caso de estudio: Documentos científico-técnicos

El tercer estudio aborda la aplicación del sistema RAG en documentos científico-técnicos, evaluando su capacidad para procesar y generar respuestas a partir de información técnica y compleja. En este caso de estudio, en concreto, hemos utilizado el modelo de *embeddings* `Multilingual Large` para hacer las pruebas con cada uno de los LLM. Asimismo, los gráficos incluidos a continuación resumen el rendimiento del sistema en este escenario:

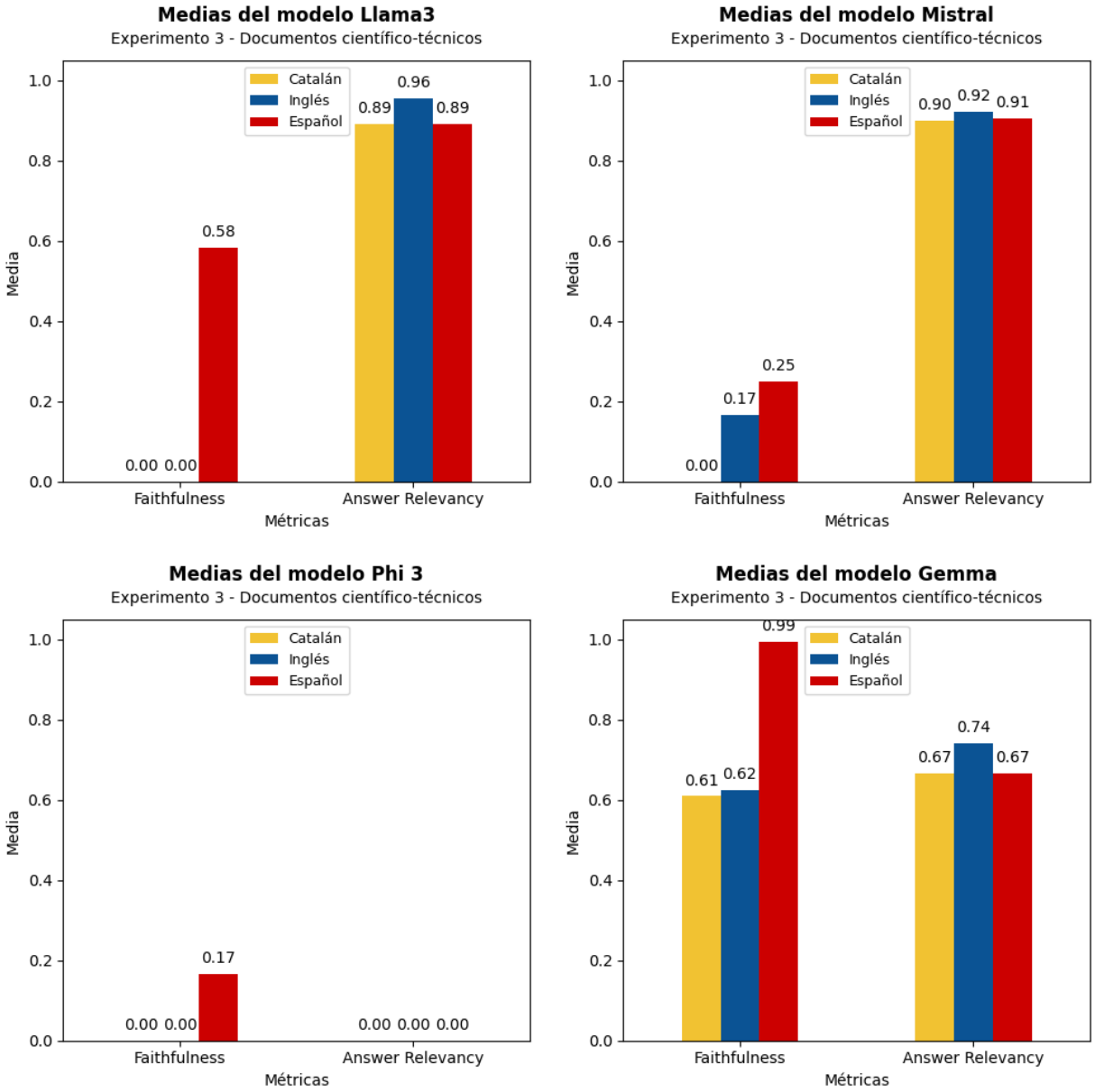

Figura 4.5: Medias modelos LLM del tercer caso de estudio

Estos resultados permiten observar cómo se comporta el sistema ante la complejidad de los documentos científico-técnicos. Este último estudio sorprende por ser el que mejor rendimiento ha proporcionado, en cuanto a medias se refiere; y el modelo `Gemma` ha despuntado notablemente.

Modelo LLM	Faithfullnes	Answer Relevancy	Índice General
Llama 3	0,19	0,91	0,31
Mistral	0,14	0,91	0,24
Phi3	0,06	0,00	0,08
Gemma	**0,74**	**0,69**	**0,71**

Cuadro 4.3: Tabla de resultados del tercer caso de estudio

Por otro lado, es relevante como la métrica de fidelidad de la respuesta, es bastante baja en general en todos los modelos y como el modelo `Phi3`, pese a ser el peor en todos los casos de estudio, en este muestra unos resultados pésimos.

CAPÍTULO 5

ANÁLISIS DE RESULTADOS

Este capítulo se centra en el análisis y discusión de los resultados obtenidos a partir de los casos de estudio realizados con el RAG. A lo largo de los estudios descritos en el capítulo anterior, se ha evaluado el rendimiento del sistema en contextos variados, como documentos jurídicos, financieros y científico-técnicos. El objetivo de este capítulo es interpretar resultados, identificar patrones y analizar tendencias de los datos recopilados. Este análisis es fundamental para evaluar la efectividad del sistema, así como para identificar posibles áreas de mejora.

Sin embargo, hay que tener en cuenta que estas pruebas se han hecho con la librería de Python RAGAS. Dicha librería no se trata de ningún estándar reconocido en el campo de la IA Generativa y los resultados tampoco se deben interpretar como inmutables. Además, los modelos *open source* no están especializados en ningún ámbito. De hecho, se entrenan con millones de datos de distintas temáticas. Es por eso que siempre que hablamos de los modelos LLM, estamos hablando de modelos preentrenados; como se comenta en el capítulo 1.

5.1. Primer caso de estudio

En este caso de estudio, el modelo que ha dado mejor rendimiento es el `Gemma`. Esto se debe a que este modelo consta de una capacidad de entendimiento del lenguaje bastante avanzada y precisión a la hora de generar texto especializado, como puede ser el jurídico. En el idioma que da mejores resultados es en inglés, lo cual no es una sorpresa. La mayoría de LLM del mercado se entrenan con cantidades de datos masivas, y gran parte de estos son en inglés.

Sorprende el buen resultado en el idioma catalán. Se debe a que el LLM no distingue muy bien entre los contextos escritos en español o en catalán, debido a la similitud de ambos idiomas. Durante la experimentación, al realizar preguntas en catalán, el

modelo de *embeddings* ha recuperado información en castellano y con poco sentido frente a la pregunta formulada. Por ejemplo

- **Pregunta:** Quins drets de bonificació a la Seguretat Social es mencionen per a contractes indefinits celebrats amb treballadors pertanyents al Sistema Nacional de Garantia Juvenil?
- **Respuesta:** Per als contractes indefinits celebrats amb treballadors pertanyents al Sistema Nacional de Garantia Juvenil, es mencionen bonificacions de 300 euros mensuals en la cotització empresarial a la Seguretat Social durant un període de 6 mesos. En el cas de contractes a temps parcial, la bonificació s'ajusta proporcionalment al percentatge de la jornada.
- **Contextos recuperados:**
  - indemnización en el supuesto de extinción de su relación con la Sociedad, siempre que no se produzca como \n consecuencia de un incumplimiento de las obligaciones del directivo, por jubilación, invalidez o por su propia \n voluntad no fundamentada en alguno de los supuestos indemnizables recogidos en los citados contratos.\n La Sociedad tiene formalizado un contrato de seguro colectivo con objeto de garantizar dichas prestaciones a los
  - principalmente en la reorganización llevada a cabo en los ejercicios 2020 y 2021 y que se encuentra detallada en \n la memoria de la Sociedad de dichos ejercicios. Todos estos activos no registrados derivan de ajustes, diferencias \n de valor y deducciones generadas por la Sociedad en ejercicios anteriores y que se encuentran reflejados en los \n modelos de declaración del Impuesto sobre Sociedades correspondientes.\n 43
  - 5.- La prestación a sus sociedades participadas de servicios de planificación, gestión comercial y asistencia \n técnica o financiera, con exclusión de las actividades que se hallen legalmente reservadas a entidades \n financieras o de crédito.\n El objeto social contempla también el desarrollo de las actividades anteriores de modo indirecto, en cualquiera \n de las formas admitidas en Derecho y, en particular, a través de la titularidad de acciones o participaciones en \n sociedades con objeto idéntico o análogo.

Por otro lado, los resultados de `Phi3` son bastante bajos. Veremos que en los siguientes casos de estudio tampoco ofrece unos resultados "normales" o "esperados", sino que muestra métricas decepcionantes. Esto se debe a que Microsoft ha desarrollado este LLM con expectativas de que ejecute en dispositivos con menos capacidad de hardware; son conocidos por Small Language Models (SLM) [38]. Este concepto es bastante nuevo dentro del mundo de la IA Generativa y apuesta por modelos livianos, pensados para ejecutarse en dispositivos como teléfonos móviles.

Finalmente, los modelos `Llama 3` y `Mistral`, ofrecen resultados bastante similares. Por lo general muestran buenos resultados en cuanto a la relevancia de la respuesta, es decir, saben generar una respuesta bien estructurada y con sentido frente a una

pregunta, pero la fidelidad de estos datos generados es bastante baja. Sin embargo, el modelo `Gemma` ofrece unos resultados entre un 20 % y un 25 % mejores respectivamente.

5.2. Segundo caso de estudio

En este caso de estudio, el modelo que mejor resultados ha dado es el `Gemma` de Google también. Aunque en cuestión de relevancia de la respuesta, el `Llama 3` y `Mistral` hayan dado mejores resultados en los tres idiomas, si hacemos un analisis global como el que nos ofrece la tabla, la media armónica nos indica que el modelo `Gemma` presenta mejor resultado. Además, los resultados en español son correctos, algo que no nos brinda el otro modelo. `Mistral` también da buenos resultados, pero mayoritariamente en inglés. Y `Phi 3`, como se ha comentado en el anterior estudio, ofrece muy malos resultados al tratarse de un SLM.

Es interesante analizar como la relevancia de la respuesta en los dos primeros casos de estudio, es parecida. Esto seguramente se deba a que comparten modelo de *embedding*, ya que esta métrica se calcula a partir de los *embeddings* sacados para generar la respuesta; tal y como vemos en la sección 3.3.1.

Sin embargo, al tratarse de documentos financieros, deberíamos centrarnos más en la fidelidad de la respuesta. Esto es porque dicha métrica evalúa que tan exacta, a nivel de precisión de los datos, es la respuesta. Esto, por tanto, ofrece una mayor exactitud y precisión, algo importante cuando hablamos de cuentas, saldos, informes anuales de rendimiento, etc. Es por eso que el modelo `Gemma` sería más interesante que el `Llama 3`.

5.3. Tercer caso de estudio

Este caso de estudio se ha realizado con otro modelo de *embedding* y por ese motivo, puede que el modelo `Llama 3` no se comporte de igual forma. De hecho, sorprende bastante las métricas obtenidas de fidelidad; siendo bastante bajas comparadas con la de las figuras 4.3 y 4.4. Sin embargo, en este caso estamos hablando de documentos académicos, científicos, técnicos, etc. Estos documentos contienen información detallada, técnica e incluso formulas matemáticas y es interesante que nuestro sistema pueda identificar y priorizar esa información más relevante para el usuario.

Por tanto, a pesar de a los muy buenos resultados que dan los modelos `Llama 3` y `Mistral` en cuanto a coherencia de la generación de la respuesta en comparación con el modelo `Gemma`, este último es, de nuevo, el más indicado. Esto se debe a que nos ofrece un mejor resultado a nivel general; como podemos ver en la tabla 4.3.

También es interesante analizar el pico de fidelidad en español. Esto se debe a que, en este caso, en lo referente a las preguntas en español sobre documentos científico-técnicos, se ha recopilado muy bien la información e incluso con distintas fórmulas. Por ejemplo, del artículo científico sobre el algoritmo *k-means*:

- **Pregunta:** ¿Qué metodología se utiliza para tratar los datos atípicos antes de aplicar el algoritmo *K-means*?
- **Respuesta:** La metodología utilizada para tratar los datos atípicos antes de aplicar el algoritmo K-means incluye la detección de atípicos utilizando un rango de tres desviaciones estándar y la imputación de valores ausentes o atípicos mediante el valor medio del resto de las lecturas válidas, asegurando que los datos sean representativos y minimizando la distorsión en los clústeres resultantes.
- **Contextos recuperados:**
  - Marrero, Carrizo, García-Santander y Ulloa-Vásquez: Uso de algoritmo K-means para clasificar perfiles de clientes...781 Detección e imputación de atípicos\n En la búsqueda de su estructura, el análisis clúster \n desarrollado es muy sensible a la inclusión de \n variables irrelevantes. Los datos atípicos pueden \n representar tanto observaciones verdaderas, que no son \n representativas en general, como una muestra reducida \n del grupo que provoca una mala representación. En
  - identificación de relaciones.En el presente estudio, se aplica el método no \n jerárquico K-means que particiona los individuos, \n en este caso clientes, en un número específico de \n grupos. Este algoritmo ha mantenido su popularidad \n en aplicaciones de agrupamiento debido a su buen \n rendimiento y competitividad con enfoques sugeridos \n más recientemente [20].\n Partiendo de un conjunto de N observaciones de \n una variable aleatoria x d-dimensional { $x1$, $x2$, ...
  - Marrero, Carrizo, García-Santander y Ulloa-Vásquez: Uso de algoritmo K-means para clasificar perfiles de clientes...783 $A = RI'k()k = 1K\sum$ \n $RS'k()k = 1K\sum(8)$ \n Datos e implementación\n Los datos iniciales corresponden a las lecturas \n de consumo registradas por los 1179 medidores \n inteligentes durante el período comprendido entre el \n lunes 4 de marzo y el jueves 4 de abril de 2019, con \n intervalos de registros de energía cada 15 minutos.\n Previo al empleo del algoritmo de agrupamiento

Por otro lado, el `Phi 3` como hemos visto en los demás casos de estudio, no ofrece ningún tipo de ventaja. De hecho, al tener tantos valores a cero, es posible que este cambio de modelo de *embedding* no sea lo más adecuado para este LLM.

5.4. Análisis general de los LLM

Para concluir este análisis de las métricas obtenidas por cada modelo y, por intentar resumir los resultados, podemos observar la figura 5.1. Dicha figura muestra las medias de cada modelo para todos los documentos evaluados, es decir, una media de todos los idiomas por cada caso de estudio. El objetivo es mostrar una visión general del rendimiento de cada modelo de LLM en cada caso de estudio; y, por tanto, presentar una mejor perspectiva de cada uno de ellos.

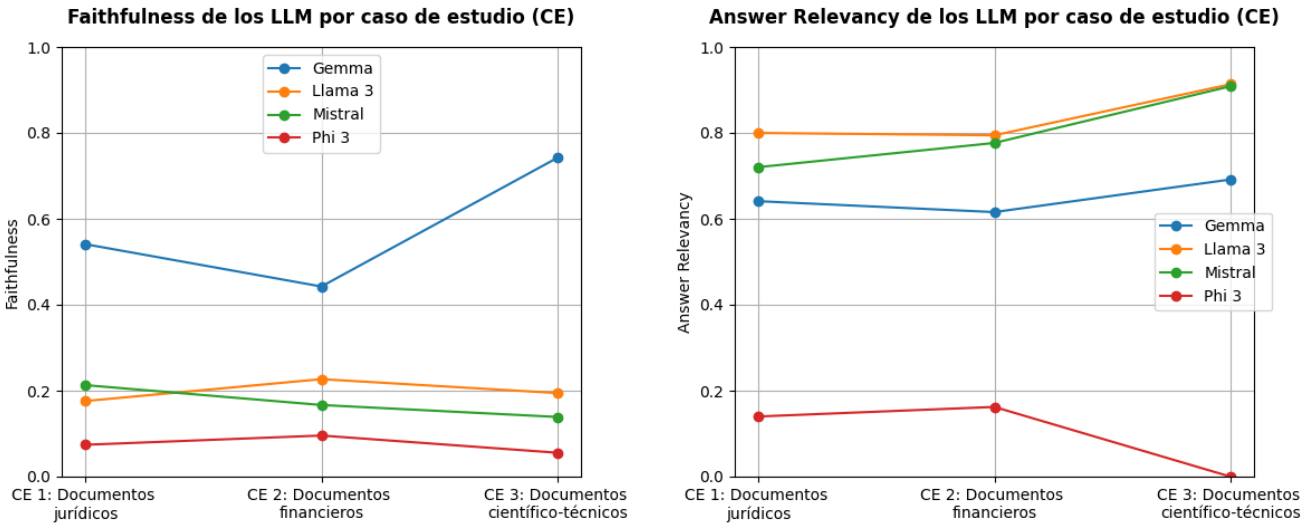

Figura 5.1: Media de las métricas obtenidas de cada LLM en cada caso de estudio

Podemos apreciar que, en términos de fidelidad, el modelo `Gemma` ofrece el mejor resultado en todos los casos de estudio. Por otro lado, cuando nos referimos a la relevancia de las respuestas, el modelo `Gemma` se ve superado por los modelos `Llama 3` y `Mistral`; que ofrecen un mejor resultado en cuanto a documentos jurídicos se trata. Sin embargo, para los otros dos casos de estudio, presentan resultados muy similares y podrían considerarse igual de válidos para acometer dichas tareas.

Cabe decir que estos gráficos no se pueden comparar directamente con los valores obtenidos en las tablas de las secciones anteriores 4.1, 4.2, 4.3. Esto se debe a que en estas tablas se saca el mejor modelo para cada caso de estudio haciendo una media de las dos métricas. Por el contrario, en las gráficas de la figura 5.1 estamos observando el rendimiento de las métricas de los modelos LLM por separado. Por tanto, la elección de un modelo u otro, dependerá de principalmente de si queremos un RAG más balanceado o de uno que priorice más ciertas métricas.

CAPÍTULO 6

LIMITACIONES

En este capítulo se analizarán las principales limitaciones encontradas durante el desarrollo e implementación del RAG, abarcando aspectos técnicos, operativos, éticos y legales que han influido en el rendimiento y la aplicabilidad del sistema en diversos contextos. El reconocimiento de estas limitaciones es esencial para comprender el alcance real del proyecto y para identificar las áreas que requieren mejoras en futuras versiones. Aunque el sistema ha demostrado ser efectivo, existen barreras y desafíos que deben ser abordados para optimizar su rendimiento y garantizar su escalabilidad y sostenibilidad a largo plazo.

6.1. Técnicas

6.1.1. Extracción de datos

Una de las principales limitaciones técnicas encontradas es la extracción de datos desde los documentos PDF. A pesar de que el sistema está diseñado para procesar este formato de manera eficiente, ciertos documentos presentaron desafíos, particularmente aquellos con estructuras complejas como tablas o gráficos integrados. Los algoritmos de extracción de texto, como los utilizados por el módulo `PyPDFDirectoryLoader` de la librería `LangChain`, han mostrado dificultades para manejar correctamente estas estructuras; lo que ha resultado en una pérdida de información o en una interpretación incorrecta del contenido.

Esto se debe a que este tipo de herramientas recoge principalmente los metadatos de los documentos y las imágenes o las tablas, no tienen un formato estándar. Este problema limita la precisión del sistema en contextos donde la integridad de la estructura del documento es crítica para la comprensión del contenido.

6.1.2. Capacidad multilingüe

Otro aspecto técnico que ha presentado limitaciones es la capacidad multilingüe del sistema. A pesar de estar diseñado para procesar documentos en inglés, español y catalán, el rendimiento no ha sido uniforme en todos los idiomas. Se ha observado que los modelos de lenguaje y *embeddings* utilizados presentan variaciones en la precisión y relevancia de las respuestas generadas en diferentes idiomas. Este fenómeno se debe, en parte, a la disponibilidad desigual de datos de entrenamiento y modelos optimizados para cada idioma; lo que impacta negativamente en la consistencia del sistema en contextos multilingües. Esto lo hemos visto claramente reflejado en los resultados obtenidos en los apartados 4.3, 4.4 y 4.5; por ejemplo, en las preguntas donde se mezclaban contextos en español y catalán.

6.1.3. Uso de memoria

El uso intensivo de memoria es otra limitación técnica significativa del sistema. Dado que el proceso de recuperación de información y generación de respuestas involucra el manejo de grandes volúmenes de datos y operaciones complejas. El sistema requiere una cantidad considerable de memoria RAM para funcionar de manera eficiente. De hecho, durante la ejecución de los casos de estudio ha sido necesario un servidor de la Universidad. El motivo es porque mi ordenador personal, a pesar de tener un buen hardware, no ha sido capaz de ejecutarlo. Sin embargo, dicho servidor contaba con la misma RAM que el primer ordenador, pero el número de CPUs era diez veces mayor. Por tanto, es posible que el problema no fuera tanto de la memoria RAM, sino de la memoria caché.

Esto puede ser un obstáculo en entornos con recursos limitados, como dispositivos con capacidades de hardware restringidas u ordenadores personales. Además, el almacenamiento en memoria de bases de datos vectoriales de gran tamaño, junto con la necesidad de mantener múltiples modelos de lenguaje en ejecución, puede llevar a un uso subóptimo de los recursos disponibles, afectando la velocidad y eficiencia del sistema.

6.2. Operativas

6.2.1. Coste del mantenimiento

El mantenimiento de un sistema RAG puede ser considerablemente costoso, especialmente cuando se implementa a gran escala. Los costes operativos incluyen no solo la infraestructura necesaria para ejecutar el sistema, como servidores y almacenamiento, sino también el costo de actualizar y mantener la base de datos vectorial y los modelos de lenguaje [39].

Uno de los principales desafíos es la necesidad de mantener actualizados los datos externos, lo que puede requerir procesos automáticos de actualización de fuentes de datos, así como la revisión periódica del rendimiento del sistema. Además, a medida que se expande la base de datos y aumenta el número de documentos y consultas pro-

cesadas, el sistema puede requerir más recursos computacionales, lo que incrementa los costos de operación.

Asimismo, la actualización y reentrenamiento de los modelos de lenguaje, aunque menos frecuente que la actualización de los datos, representa otro costo significativo. Esto es necesario para asegurar que el sistema siga siendo competitivo y capaz de manejar nuevas clases de consultas o adaptar su desempeño a cambios en el dominio del conocimiento.

Finalmente, el personal especializado requerido para la supervisión, mantenimiento y mejora continua del sistema, también representan un componente importante del coste operativo. Estos factores deben considerarse al evaluar la viabilidad a largo plazo del sistema.

6.3. Éticas y Legales

6.3.1. Privacidad y seguridad de los datos

Dado que el sistema RAG está diseñado para operar de manera local, se minimizan muchos de los riesgos asociados con la transmisión de datos sensibles a través de redes externas. Sin embargo, esto no elimina la necesidad de establecer protocolos robustos de seguridad para proteger la información almacenada y procesada localmente.

El tratamiento de documentos que contienen información personal o confidencial requiere la implementación de medidas de seguridad adecuadas, como el cifrado de datos en reposo y en tránsito, así como controles de acceso estrictos para garantizar que solo el personal autorizado pueda interactuar con el sistema. Además, es esencial asegurar que cualquier procesamiento de datos cumpla con las normativas de protección de datos aplicables, como el Reglamento General de Protección de Datos (RGPD) en Europa.

Otra consideración importante es la gestión de los `logs` [1] y otros datos generados por el sistema durante su operación. Estos deben manejarse con el mismo nivel de cuidado que los documentos originales, ya que pueden contener información sensible derivada de las consultas realizadas.

6.3.2. Sesgo de los datos

Los sistemas de IA, incluido el RAG, pueden heredar o amplificar sesgos presentes en los datos con los que han sido entrenados [16]. Este es un riesgo significativo, especialmente cuando se utilizan para tareas que requieren imparcialidad, como la toma de decisiones legales o la evaluación financiera.

Es fundamental evaluar continuamente el sistema para detectar y mitigar cualquier sesgo en las respuestas generadas. Esto puede implicar la revisión y ajuste de los mo-

[1] Registro de toda la actividad que ha sucedido en un periodo de tiempo para llevar a cabo cierta actividad o tarea.

delos de lenguaje y de las bases de datos utilizadas para asegurar que las respuestas proporcionadas sean equitativas y no perpetúen estereotipos o discriminaciones.

Además, es importante que los desarrolladores y usuarios del sistema sean conscientes de estos riesgos y tomen medidas proactivas para minimizar el impacto del sesgo. Esto incluye la selección cuidadosa de las fuentes de datos, la implementación de técnicas de ajuste de modelos y la realización de auditorías regulares del sistema para detectar y corregir posibles problemas.

CAPÍTULO 7

CONCLUSIONES

Este trabajo ha abordado la creación, implementación y evaluación de un sistema RAG en un entorno local, utilizando modelos de código abierto para garantizar la privacidad y seguridad de los datos. A lo largo de este proceso, se han planteado y cumplido diversos objetivos que han permitido evaluar la capacidad del sistema en diferentes contextos. En estas conclusiones, se examinan los resultados obtenidos en función de los objetivos propuestos, así como las fortalezas, limitaciones y futuras direcciones del sistema desarrollado.

7.1. Cumplimiento de Objetivos

Los objetivos planteados para este TFG han sido alcanzados de manera satisfactoria. A continuación, se detallan los logros asociados a cada uno de ellos:

O1 Diseño y creación de un RAG en local.
El sistema RAG implementado es capaz de operar de manera local utilizando modelos abiertos de la plataforma HuggingFace. La capacidad del sistema de procesar archivos PDF y permitir la selección de modelos, asegurando su adaptabilidad a diferentes tipos de datos y configuraciones.

O2 Selección e ingesta de documentos por parte del RAG.
El sistema ha demostrado su capacidad para procesar eficientemente documentos de diferentes tipos, como documentos financieros y académicos en formato PDF.

O3 Diseño y evaluación de preguntas para la evaluación sobre el contenido de los documentos.
La inserción de datos en el sistema y la generación de preguntas sobre los documentos han sido llevadas a cabo con éxito. Se ha creado un conjunto de cincuenta y cuatro preguntas, de diferente dificultad, que han sido utilizadas para evaluar la capacidad del RAG de generar respuestas precisas.

O4 Analizar la capacidad, tanto de comprensión como de expresión, del sistema en distintas lenguas (inglés, castellano y catalán).
A través de los casos de estudio realizados, se ha logrado evaluar el rendimiento del sistema en diferentes lenguas. El sistema es capaz de procesar y generar respuestas en inglés, castellano y catalán con resultados muy dispares. Por tanto, actualmente es difícil asegurar una buena versatilidad del sistema en entornos multilingües; lo cual es una característica de gran valor, y pendiente de mejora, en aplicaciones prácticas.

7.2. Análisis de los Resultados

El análisis de los resultados obtenidos refleja varios aspectos importantes sobre el rendimiento del sistema. En primer lugar, se ha demostrado que el uso de diferentes modelos de *embeddings* tiene un impacto significativo en la calidad de los contextos recuperados, lo que, a su vez, influye en la precisión de las respuestas generadas por los modelos LLM. La evaluación comparativa de los modelos de *embeddings*, presentada en las figuras 4.1 y 4.2, pone de manifiesto que algunos modelos, como `Multilingual Large`, tienden a ser más precisos en la recuperación de datos técnicos, mientras que otros modelos pueden ser más adecuados para textos generales.

Por otro lado, se ha observado que el sistema RAG es capaz de gestionar un volumen considerable de datos. Con más de dieciocho documentos insertados, que equivalen a un total de quinientas ochenta páginas procesadas y miles de *embeddings* generados. Esto demuestra la robustez del sistema en términos de capacidad de procesamiento y escalabilidad. Sin embargo, también se han identificado áreas de mejora, como la optimización del tiempo de recuperación de información relevante y generación de respuestas, especialmente en documentos extensos o altamente técnicos.

En cuanto a la generación de respuestas, el sistema ha mostrado una buena precisión, lo que confirma su potencial en aplicaciones; donde la consulta de grandes volúmenes de documentos es esencial. No obstante, los resultados varían ligeramente dependiendo del modelo de LLM utilizado, lo que sugiere que la combinación de modelos LLM y modelos de *embeddings* debe ser cuidadosamente seleccionada para optimizar el rendimiento en contextos específicos.

7.3. Implicaciones Futuras y Desarrollo

Los resultados obtenidos en este Trabajo Final de Grado abren diversas posibilidades para el desarrollo futuro del sistema RAG. En primer lugar, sería interesante explorar la integración de modelos de mayor capacidad como el reciente Llama 3.1 en la versión de 405B de parámetros [40]. Estos modelos LLM podrían mejorar la precisión y la coherencia de las respuestas generadas; aunque sería necesario un hardware extremadamente potente a día de hoy. Concretamente, sería necesario un computador con 256 GB de RAM y 8 gráficas de NVIDIA [41] con 80 GB de VRAM[1] cada una. Claramente, otras opciones podrían ser más rentables con y aun así garantizar una buena

[1] La VRAM es la memoria RAM dedicada que tiene una tarjeta gráfica

optimización del sistema, especialmente en términos de tiempo de respuesta; lo que permitiría su aplicación en entornos donde la rapidez es crítica.

Otro aspecto a considerar es la incorporación de una evaluación más exhaustiva en términos de comprensión multilingüe. Aunque el sistema ha demostrado ser competente, la expansión a otros idiomas podría ser valiosa, especialmente en aplicaciones globales. Por último, la implementación de mecanismos de retroalimentación y aprendizaje continuo permitiría al sistema mejorar sus capacidades de recuperación y generación de manera autónoma. Esto es esencial, ya que uno de los principales inconvenientes de esta tecnología es mantener la información actualizada para que el sistema sea capaz de ofrecer respuesta a lo largo del tiempo.

BIBLIOGRAFÍA

[1] K. M. G. López, "Inteligencia artificial generativa: Irrupción y desafíos," *Enfoques*, vol. 4, no. 2, pp. 57–82, 2023. 1

[2] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. .ºReilly Media, Inc.", 2016. 1

[3] Z. Ghahramani, "Unsupervised learning," in *Summer school on machine learning*. Springer, 2003, pp. 72–112. 1

[4] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A gentle introduction to reinforcement learning and its application in different fields," *IEEE access*, vol. 8, pp. 209 320–209 344, 2020. 1

[5] IBM, "What is semi-supervised learning?" 2024, accedido: 16 de octubre de 2024. [Online]. Available: https://www.ibm.com/topics/semi-supervised-learning 1

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 1

[7] OpenAI, "Openai official web," 2024, accedido: 12 de julio de 2024. [Online]. Available: https://openai.com/ 1, 4.1

[8] Leonie Monigatti and Zain Hasan, "Which is better, retrieval augmentation (rag) or fine-tuning? both." 2024, accedido: 30 de octubre de 2024. [Online]. Available: https://snorkel.ai/blog/which-is-better-retrieval-augmentation-rag-or-fine-tuning-both/ 1.1

[9] M. E. Zarlenga, P. Barbiero, G. Ciravegna, G. Marra, F. Giannini, M. Diligenti, F. Precioso, S. Melacci, A. Weller, P. Lio *et al.*, "Concept embedding models," in *NeurIPS 2022-36th Conference on Neural Information Processing Systems*, 2022. 1

[10] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023. 2

[11] C.-W. Sung, Y.-K. Lee, and Y.-T. Tsai, "A new pipeline for generating instruction dataset via rag and self fine-tuning," in *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2024, pp. 2308–2312. 2.1

[12] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020. 2.1, 2.2.1

[13] Hoang Tran, "A gentle introduction to vector databases," 2023, accedido: 30 de octubre de 2024. [Online]. Available: https://weaviate.io/blog/what-is-a-vector-database 2.1

[14] E. B. Thomas, "Decoding the enigma: A deep dive into transformer model architecture," https://medium.com/@ebinbabuthomas_21082/decoding-the-enigma-a-deep-dive-into-transformer-model-architecture-749b49883628, 2023, accedido: 7 de junio de 2024. 2.1

[15] Catav, Amnon and Miara, Roy and Giloh, Ilai and Cordeiro, Nathan and Ingber, Amir, "Rag makes llms better and equal," 2024, accedido: 12 de julio de 2024. [Online]. Available: https://www.pinecone.io/blog/rag-study/ 2.2

[16] Z. Wang, Z. Wang, L. Le, H. S. Zheng, S. Mishra, V. Perot, Y. Zhang, A. Mattapalli, A. Taly, J. Shang *et al.*, "Speculative rag: Enhancing retrieval augmented generation through drafting," *arXiv preprint arXiv:2407.08223*, 2024. 2.2.1, 6.3.2

[17] M. Allahyari and A. Yang, "A practical approach to retrieval-augmented generation systems," 2023, accedido: 16 de septiembre de 2024. [Online]. Available: https://mallahyari.github.io/rag-ebook/02_rag.html 2.2.2

[18] V. Singh, "Building llm applications: Advanced rag (part 10)," 2024, accedido: 5 de julio de 2024. [Online]. Available: https://medium.com/@vipra_singh/building-llm-applications-advanced-rag-part-10-ec0fe735aeb1 2.3

[19] Prompting Guide, "Retrieval-augmented generation (rag)," 2024, accedido: 5 de julio de 2024. [Online]. Available: https://www.promptingguide.ai/research/rag 2.3

[20] Ollama, "Llama 3," 2024. [Online]. Available: https://ollama.com/library/llama3 2.4

[21] ——, "Mistral," 2024. [Online]. Available: https://ollama.com/library/mistral 2.4

[22] ——, "Phi 3," 2024. [Online]. Available: https://ollama.com/library/phi3 2.4

[23] ——, "Gemma," 2024. [Online]. Available: https://ollama.com/library/gemma 2.4

[24] Intfloat, "Multilingual e5 large," 2024. [Online]. Available: https://huggingface.co/intfloat/multilingual-e5-large 2.4

[25] BAAI, "Bge large zh v1.5," 2024. [Online]. Available: https://huggingface.co/BAAI/bge-large-zh-v1.5 2.4

[26] Beijing Academy of Artificial Intelligence, "Beijing academy of artificial intelligence official web," 2024, accedido: 15 de julio de 2024. [Online]. Available: https://www.baai.ac.cn/ 2.4

[27] Mixedbread AI, "Mxbai embed large v1," 2024. [Online]. Available: https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1 2.4

[28] BAAI, "Bge small en v1.5," 2024. [Online]. Available: https://huggingface.co/BAAI/bge-small-en-v1.5 2.4

[29] A. PG, "Future trends in retrieval-augmented generation (rag): Innovations and applications," 2024, accedido: 16 de octubre de 2024. [Online]. Available: https://www.cloud2data.com/future-trends-in-retrieval-augmented-generation-rag/ 2.5

[30] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, "Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely," 2024. [Online]. Available: https://arxiv.org/abs/2409.14924 2.5

[31] Y. Gao, Y. Xiong, M. Wang, and H. Wang, "Modular rag: Transforming rag systems into lego-like reconfigurable frameworks," *arXiv preprint arXiv:2407.21059*, 2024. 2.5

[32] Git SCM, "Git - distributed version control system," 2024. [Online]. Available: https://git-scm.com/ 1

[33] L. Barca Pons, "Document question answering," https://github.com/Luisbp27/documentQuestionAnswering, 2024. 3, 3.1.2, 4, 4.1

[34] HuggingFace, "Hugging face official website," https://huggingface.co/, 2024, accedido: 10 de mayo de 2024. 2

[35] Ollama, "Ollama official website," https://ollama.com/, 2024, accedido: 10 de junio de 2024. 3.1.2

[36] Snowflake, "Streamlit official website," https://streamlit.io/, 2024, accedido: 25 de mayo de 2024. 3.1.2

[37] ExplodingGradients, "Ragas official documentation," 2024, accedido: 2 de junio de 2024. [Online]. Available: https://docs.ragas.io/en/stable/ 3.3

[38] Bilenko, Misha, "Introducing phi-3: Redefining what's possible with slms," 2024, accedido: 19 de septiembre de 2024. [Online]. Available: https://azure.microsoft.com/en-us/blog/introducing-phi-3-redefining-whats-possible-with-slms/ 5.1

[39] T. Selvaraj, "Calculate the total cost of a retrieval-augmented generation (rag) solution," 2023, accedido: 15 de agosto de 2024. [Online]. Available: https://medium.com/searchblox/calculate-the-total-cost-of-a-retrieval-augmented-generation-rag-solution-935aa5b397cf 6.2.1

[40] L. Model, "Llama 3.1 requirements," 2024, accedido: 22 de octubre de 2024. [Online]. Available: https://llamaimodel.com/requirements/ 7.3

[41] NVIDIA, "Gpu h100 tensor core," 2024, accedido: 22 de octubre de 2024. [Online]. Available: https://www.nvidia.com/es-es/data-center/h100/ 7.3