

Recuperación de información aumentada para grandes modelos de lenguaje de código abierto

Un caso de estudio en ayudas públicas del
Gobierno de España.

UOC

Desiderio Martí Alcaraz

Máster en Ciencia de Datos
Área 2 - Aula 1

Tutor/a de TF

José Luis Iglesias Allones

**Profesor/a responsable de
la asignatura**

Josep-Anton Mir Tutusaus, Esther
Ibáñez Marcelo

09/01/2025

Universitat Oberta
de Catalunya



Esta obra está sujeta a una licencia de Reconocimiento-
NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Ficha del Trabajo Final

Título del trabajo:	Recuperación de información aumentada para grandes modelos de lenguaje de código abierto: Un caso de estudio en ayudas públicas del Gobierno de España.
Nombre del autor/a:	Desiderio Martí Alcaraz
Nombre del Tutor/a de TF:	Jose Luis Iglesias Allones
Nombre del/de la PRA:	Josep-Anton Mir Tutusaus, Esther Ibáñez Marcelo
Fecha de entrega:	09/01/2025
Titulación o programa:	Máster en Ciencia de Datos
Área del Trabajo Final:	Área 2 - Aula 1
Idioma del trabajo:	Castellano
Palabras clave	Web scraping, RAG, Open LLMs
Resumen del Trabajo	
<p>Este Trabajo de Fin de Máster (TFM) tiene como objetivo desarrollar un sistema basado en Retrieval-Augmented Generation (RAG) y modelos de lenguaje abiertos (Open LLMs) para la extracción automatizada de información relevante de la web del Sistema Nacional de Ayudas y Subvenciones Públicas.</p> <p>La finalidad del proyecto es mejorar la accesibilidad y eficiencia en la búsqueda de subvenciones, permitiendo a los usuarios obtener respuestas concisas y contextualmente correctas basadas en datos públicos disponibles en la web.</p> <p>En el contexto actual, el acceso a la información sobre ayudas y subvenciones puede ser complicado debido a la gran cantidad de datos y la forma de consumo de esta información. Para abordar este desafío, el sistema combina el web scraping con técnicas RAG proporcionando respuestas basadas tanto en el contenido recuperado como en las capacidades de los LLMs.</p> <p>Queremos plantear la realización de una especie de "asistente" virtual que</p>	

pueda responder a preguntas concretas sobre el tema comentado

Se pretende implementar una aplicación que mediante estas tecnologías podamos (1) extraer los datos mediante web scraping de las páginas de subvenciones, (2) almacenar esta información, (3) usar un modelo LLM abierto para generar respuestas a partir de esta información, y (4) evaluar del rendimiento del sistema en términos de precisión, relevancia y utilidad de las respuestas.

Los resultados del proyecto deben demostrar que la combinación de RAG y Open LLMs mejora significativamente la experiencia de usuario al ofrecer respuestas precisas y rápidas sobre las ayudas disponibles.

Las conclusiones deben destacar el potencial de esta tecnología para aplicaciones en áreas con grandes volúmenes de datos públicos y su aplicabilidad a datos privados.

Abstract

This Master's Thesis (TFM) aims to develop a system based on Retrieval-Augmented Generation (RAG) and open language models (Open LLMs) for the automated extraction of relevant information from the website of the National Aid and Subsidies System Public.

The purpose of the project is to improve accessibility and efficiency in grant searching, allowing users to obtain concise and contextually correct answers based on public data available on the web.

In the current context, access to information on aid and subsidies can be complicated due to the large amount of data and the way this information is consumed. To address this challenge, the system combines web scraping with RAG techniques providing answers based on both the retrieved content and the capabilities of the LLMs.

We want to propose the creation of a kind of virtual "assistant" that can answer specific questions about the topic discussed.

The aim is to implement an application that, using these technologies, can (1) extract data through web scraping from grant pages, (2) store this information, (3) use an open LLM model to generate responses from this information, and (4) evaluate the performance of the system in terms of accuracy, relevance and usefulness of the responses.

The results of the project must demonstrate that the combination of RAG and Open LLMs significantly improves the user experience by offering accurate and fast responses on the available aids.

Conclusions should highlight the potential of this technology for applications in areas with large volumes of public data and its applicability to private data.

Índice

1.	Introducción	6
1.1.	Descripción de la propuesta, justificación del interés y la relevancia de la propuesta	6
1.2.	Explicación de la motivación personal	8
1.3.	Definición de los objetivos (principales y secundarios)	8
1.4.	Impacto en sostenibilidad, ético-social y de diversidad	9
1.5.	Descripción de la metodología empleada en el desarrollo del proyecto	10
1.6.	Planificación y plan de investigación del proyecto	12
2.	Estado del Arte	15
2.1.	Contexto	15
2.2.	Recuperación de Información y Web Scraping	15
2.2.1.	Contexto y evolución histórica	15
2.2.2.	Web scraping	17
2.2.3.	Consideraciones éticas y legales	18
2.3.	Procesamiento de Lenguaje Natural y Grandes Modelos del Lenguaje	19
2.3.1.	Introducción al NLP y evolución hasta los LLMs	19
2.3.1.1.	Contexto general	19
2.3.1.2.	Evolución hasta los LLMs	20
2.3.2.	Representaciones vectoriales (Embeddings)	22
2.3.2.1.	Contexto	22
2.3.2.2.	Definición	23
2.3.2.3.	Técnicas de Generación de Embeddings	24
2.3.3.	Almacenamiento de la información y Bases de datos vectoriales	26
2.3.3.1.	Introducción	26
2.3.3.2.	Bases de datos Vectoriales	26
2.3.4.	Grandes Modelos de Lenguaje (LLMs)	27
2.3.4.1.	Contexto y transformers	27
2.3.4.2.	Comparativa de modelos históricos	28
2.3.4.3.	Avances recientes: capacidades multimodales	30
2.3.4.4.	Importancia y aplicaciones	31
2.3.5.	Técnicas avanzadas para optimizar LLMs	32
2.3.5.1.	Fine-tuning: personalización de modelos para tareas específicas.	32

2.3.5.2.	Recuperación de Información Aumentada (RAG)	33
2.3.5.2.1.	Introducción	33
2.3.5.2.2.	Ventajas	34
2.3.5.3.	Prompt Engineering	35
2.3.6.	Evaluación de modelos	35
2.3.6.1.	Contexto	35
2.3.6.2.	Evaluación de RAG	36
2.4.	Desafíos y Consideraciones Éticas	38
2.4.1.	Centralizar los aspectos técnicos, éticos y sostenibles.	38
2.5.	Resumen y aplicabilidad al trabajo actual	39
3.	Materiales y métodos	40
3.1.	Descripción general y arquitectura	40
3.2.	Fuentes de datos y Web Scrapping	43
3.2.1.	Obtención de los datos mediante el API REST	45
3.2.2.	Estructura y características de los datos	45
3.2.3.	Consideraciones técnicas	46
3.3.	Preprocesamiento de los datos y generación de Embeddings	46
3.3.1.	Limpieza y estructuración de los datos	46
3.4.	Almacenamiento y búsqueda de datos	48
3.4.1.	Base de datos vectorial: ChromaDB y FAISS	48
3.5.	Tecnologías utilizadas	49
3.5.1.	Embeddings: paraphrase-multilingual-mpnet-base-v2	49
3.5.2.	Base de Datos Vectorial: ChromaDB y FAISS	49
3.5.3.	LLMs: Ollama y OPENAI	50
3.5.4.	Despliegue en HuggingFace Spaces	50
3.6.	Interfaz de usuario	51
4.	Evaluación y resultados	55
4.1.	Evaluación con RAGAS	55
4.1.1.	Planteamiento general y objetivos	55
4.1.2.	Métricas utilizadas	56
4.1.3.	Generación de datasets de preguntas	57
4.1.3.1.	Introducción	57
4.1.3.2.	Datasets de preguntas sintéticos	57

4.1.3.3.	Datasets de preguntas con RAGAS	58
4.1.4.	Generación de Datasets para la evaluación	58
4.1.4.1.	Generación de Datasets mediante IU	58
4.1.4.2.	Generación de Datasets mediante búsquedas desatendidas	59
4.1.5.	Evaluación de la solución	59
4.1.6.	Resultados Cuantitativos – Dataset sintético	60
4.1.7.	Resultados Cuantitativos – Dataset generado por RAGAS	61
4.1.8.	Resultados finales de evaluación.	63
4.2.	Resultados	64
5.	Conclusiones y trabajos futuros	65
6.	Glosario	67
7.	Bibliografía	68
8.	Anexos	72
8.1.	Planificación detallada	72
8.2.	Embeddings	76
8.3.	Comparativa de Bases de Datos Vectoriales	81
8.4.	Retrieval-Augmented Generation Analytics and Scoring (RAGAS)	83
8.5.	Consideraciones legales al obtener datos de API	87
8.6.	Datasets para evaluación con RAGAS	89
8.7.	Ejemplos de resultados de la evaluación	94

Lista de Figuras

Figura 1: Diagrama Gantt Resumido	14
Figura 2: Etapas Web Scraping	17
Figura 3: Evolución técnicas PLN	20
Figura 4: Diagrama Embeddings	23
Figura 5: Diagrama Generación Embeddings	24
Figura 6: Diagrama Embeddings WORD2VEC	25
Figura 7: Diagrama Embeddings BERT	26
Figura 8: Capas Transformers	28
Figura 9: Diagrama LLMs desde distintas variables	29
Figura 10: Diagrama RAG	34
Figura 11: Diagrama de arquitectura a alto nivel	42
Figura 12: Sistema Nacional de Publicidad de Subvenciones	43
Figura 13: Ficha detalle de ayuda pública	44
Figura 14: Ficha en formato PDF	44
Figura 15: Imagen web accesible en HuggingFace Spaces	51
Figura 16: Imagen de página de carga de ayudas en pdf	53
Figura 17: Imagen de página de listado de documentos	53
Figura 18: Imagen de página Ragas - Datasets	54
Figura 19: Imagen de página Ragas - Resultados	54
Figura 20: Imagen de página Aviso legal	55
Figura 21: Histograma de las métricas dataset sintético	60
Figura 22: Histograma de las métricas dataset RAGAS	62
Figura 23: Diagrama de Gantt (parte 1)	75
Figura 24: Diagrama de Gantt (parte 2)	76
Figura 25: Comparativa BD Vectoriales desde 2 perspectivas	83
Figura 26: Árbol Métricas RAGAS	84
Figura 27: Página de Aviso Legal	88

Lista de Tablas

Tabla 1: Recursos Hardware	12
Tabla 2: Recursos Software	12
Tabla 3: Métricas RAGAS evaluadas	57
Tabla 4: Estadísticas descriptivas del dataset sintético.	60
Tabla 5: Estadísticas descriptivas del dataset ragas.	61
Tabla 6: Comparativa de bases de datos vectoriales	81
Tabla 7: Clasificación Métricas RAGAS	84

1. Introducción

1.1. Descripción de la propuesta, justificación del interés y la relevancia de la propuesta

El punto de partida de este proyecto es la creciente complejidad y volumen de información relacionada con las ayudas y subvenciones públicas en España. En la actualidad, tanto ciudadanos como empresas que buscan acceder a estas ayudas enfrentan dificultades para encontrar información relevante de manera eficiente. Esta situación genera una barrera de entrada para muchos potenciales beneficiarios, quienes a menudo carecen de tiempo, conocimientos o recursos para efectuar búsquedas exhaustivas.

El problema se agrava por la forma en que se presenta la información: aunque está disponible, los datos no siempre son fácilmente comprensibles o accesibles para todos los usuarios. Además, el proceso de búsqueda manual no es solo ineficiente, sino que también puede llevar a la pérdida de oportunidades debido a la falta de visibilidad de algunas subvenciones o de información contenida en documentos que no está estructurada ni permite su búsqueda. Este es un tema fundamental porque las ayudas y subvenciones públicas son herramientas clave para impulsar el desarrollo económico y social, y su accesibilidad es fundamental para garantizar una distribución equitativa de los recursos públicos.

Al inicio de este trabajo, las soluciones disponibles se limitaban a sistemas de búsqueda tradicionales o portales con filtrado básico. No existe un sistema capaz de integrar grandes volúmenes de información, procesarla y proporcionar respuestas precisas y personalizadas a las consultas de los usuarios de una manera simple.

La aportación principal de este proyecto es la implementación de un sistema basado en RAG (Retrieval-Augmented Generation) y Open LLM (Open Large Language Model) que permita la extracción automatizada, almacenamiento y búsqueda de información relevante sobre ayudas y subvenciones.

Existen muchas definiciones de lo que es un modelo de lenguaje grande (LLM, por sus siglas en inglés). Una primera aproximación la encontramos en Wikipedia (Modelo extenso de lenguaje, 2024), que describe un LLM como un modelo de lenguaje basado en una red neuronal con un número muy alto de parámetros (generalmente miles de millones o más), entrenado con enormes volúmenes de texto sin etiquetar mediante aprendizaje auto supervisado o semi-supervisado. De forma más sencilla, podríamos decir que un LLM es una inteligencia artificial entrenada para comprender y generar texto de manera natural. Este modelo aprende de un conjunto finito de enormes cantidades de texto, como libros,

artículos y páginas web, y es utilizado para responder preguntas, crear resúmenes o mantener una conversación coherente.

Estos modelos de lenguaje presentan diversas limitaciones, siendo una de las más importantes la cantidad limitada de información que pueden almacenar. Esto los hace útiles para responder preguntas relacionadas con el conocimiento incluido en el modelo; sin embargo, ¿qué ocurre si deseamos obtener información que no ha sido incorporada en su entrenamiento? En estos casos, existen diferentes enfoques, entre ellos los siguientes:

- Fine-tuning: consiste en reentrenar el modelo original con la información específica que necesitamos consultar, aunque este enfoque implica un alto costo en términos de tiempo y recursos computacionales.
- Bases de datos de conocimiento: se basa en el uso de bases de datos semánticas, grafos de conocimiento o sistemas de preguntas y respuestas para organizar y acceder a la información.
- Prompts personalizados: afinando las consultas mediante prompts estructurados, lo cual facilita la recuperación de información específica a través de un modelo de lenguaje.
- Retrieval-Augmented Generation (RAG): una técnica en la que se recupera información externa, no almacenada en el modelo, para proporcionarle al modelo un contexto adicional que le permita generar respuestas más precisas y relevantes.

En este trabajo nos centraremos en la técnica de RAG. Es importante entender qué es RAG; en Wikipedia (Retrieval-augmented generation, 2024), se define como un tipo de inteligencia artificial generativa que incorpora capacidades de recuperación de información. Esta técnica combina dos conceptos clave: en primer lugar, la obtención de información actualizada a partir de fuentes como documentos, sitios web, bases de datos, entre otros, que se almacenan en bases de datos vectoriales o grafos de conocimiento; en segundo lugar, el uso de un modelo de lenguaje grande (LLM) que se alimenta de esta información específica, permitiendo así generar respuestas más precisas y actualizadas.

La propuesta de este trabajo se basa en el uso de técnicas de RAG mediante LLMs de código abierto, los cuales pueden implementarse en infraestructuras on-premise. Esta configuración permite garantizar la privacidad de los datos, una característica esencial para muchas empresas que prefieren no utilizar LLMs como servicio, evitando así que la información del modelo se almacene en sistemas de terceros.

1.2. Explicación de la motivación personal

Mi motivación para desarrollar este Trabajo de Fin de Máster surge de mi experiencia profesional como ingeniero de software y mánager, especializado en la creación y gestión de intranets, portales documentales y plataformas colaborativas. A lo largo de mi carrera, he trabajado con grandes volúmenes de información interna en entornos empresariales, enfrentándome a uno de los principales desafíos: asegurar que los usuarios accedan a la información correcta de manera rápida y eficiente.

A lo largo de los años, he utilizado soluciones de terceros Out of the Box (OOB), como Google Mini, para implementar búsquedas en intranets corporativas, en su momento ofrecidas por Google. También he empleado productos de terceros en bases de datos de conocimiento y asistentes virtuales para optimizar la recuperación de información y mejorar la precisión en la resolución de consultas.

Este proyecto me ofrece la oportunidad de explorar y probar la técnica de Retrieval-Augmented Generation (RAG) combinada con modelos de lenguaje abiertos (LLMs) en el contexto de acceso a información pública, como son las ayudas y subvenciones del Gobierno de España. En el futuro, el objetivo es que estas técnicas se conviertan en una solución altamente efectiva para mejorar las búsquedas en intranets corporativas y sistemas documentales internos de las empresas.

Uno de los principales impulsores de este trabajo es la posibilidad de aplicar, en el futuro, los conocimientos adquiridos a entornos empresariales internos. Mi objetivo a largo plazo es implementar esta tecnología en portales documentales internos, donde la dispersión de información y la falta de herramientas avanzadas de búsqueda son problemáticas comunes. Considero que un sistema basado en Retrieval-Augmented Generation (RAG) y modelos de lenguaje (LLMs) tiene el potencial de transformar la forma en que se accede y gestiona la información dentro de las organizaciones, proporcionando una ventaja competitiva en términos de eficiencia operativa y en la toma de decisiones.

1.3. Definición de los objetivos (principales y secundarios)

El presente Trabajo de Fin de Máster (TFM) tiene como objetivo general desarrollar un sistema que, mediante el uso de técnicas de RAG y Open LLMs, facilite la extracción automatizada de información relevante del Sistema Nacional de Ayudas y Subvenciones Públicas del Gobierno de España. Este objetivo se desglosa en los siguientes objetivos principales y secundarios:

- Extracción de datos estructurados y no estructurados de webs

- Identificar y seleccionar fuentes de información relevantes en el Sistema Nacional de Ayudas y Subvenciones Públicas para extraer datos clave mediante técnicas de web scraping.
 - Desarrollar un sistema de extracción automática de información, utilizando web scraping para capturar datos de ayudas y subvenciones públicas de forma estructurada y no estructurada.
- Indexación de información extraída
- Implementar un modelo RAG que integre la recuperación de documentos con la generación de texto, optimizando las respuestas generadas basadas en los datos recuperados.
 - Integrar la solución de Open LLM con el fin de que sea capaz de procesar y generar respuestas personalizadas y contextualizadas a las consultas de los usuarios sobre ayudas y subvenciones.
 - Optimizar el sistema para diferentes tipos de consultas, garantizando que las respuestas proporcionadas sean claras, comprensibles y accesibles para un amplio espectro de usuarios, desde ciudadanos hasta empresas.
 - Evaluar la precisión, relevancia y utilidad del sistema a través de pruebas con usuarios y métricas de rendimiento, comparando los resultados con los métodos tradicionales de búsqueda de información. Aquí utilizaremos el framework RAGAS (Ragas, 2024) que nos ayuda a esta evaluación.
 - Implementación de una interfaz web que, a través de un chatbot, proporcione una experiencia de uso accesible para la consulta y consumo de esta información.
 - Documentación de las metodologías, procesos y resultados obtenidos a lo largo del desarrollo del sistema, incluyendo un análisis crítico sobre las ventajas y limitaciones de la solución propuesta.

1.4. Impacto en sostenibilidad, ético-social y de diversidad

Este trabajo, centrado en el contexto de ayudas públicas, tiene como objetivo facilitar un acceso más equitativo a la información para todos los ciudadanos. Buscamos lograr un equilibrio entre eficiencia, sostenibilidad y ética, incorporando principios de transparencia, optimización energética y diversidad en el diseño.

En lo que respecta a la sostenibilidad, es fundamental optimizar el uso de datos y controlar el consumo computacional para mitigar el impacto ambiental. Se propone la implementación de modelos ligeros ejecutados localmente, lo que nos permitirá optimizar los costes energéticos y evitar la dependencia de grandes infraestructuras en la nube.

Por otro lado, el impacto ético-social del uso de información pública es sumamente relevante, ya que afecta directamente a los ciudadanos y a su bienestar económico. En este sentido, existen varios aspectos clave que deben considerarse en el análisis ético-social de estos sistemas:

- Privacidad y uso responsable de datos: Es fundamental respetar la privacidad de los usuarios y cumplir con normativas de protección de datos, como la Ley Orgánica de Protección de Datos (LOPD).
- Prevención de sesgos: Es crucial que los LLMs se desarrollen y ajusten para minimizar los sesgos en las respuestas generadas.
- Inclusión de mecanismos de retroalimentación: Es necesario ofrecer a los usuarios la capacidad de proporcionar retroalimentación, lo que permitirá que el sistema mejore con el tiempo y se adapte mejor a las necesidades de diferentes grupos poblacionales. Esto fomentará una relación más ética y dinámica entre los ciudadanos y la tecnología

Otro aspecto importante de este trabajo es la diversidad, que debe garantizar la satisfacción de las necesidades de todos los grupos de la sociedad. Para ello, es fundamental considerar los siguientes principios:

- Datos representativos: Para que sean útiles, los datos deben reflejar la totalidad de los sectores de la sociedad. Esto implica integrar información sobre ayudas para todos los colectivos, por muy minoritarios que sean, así como para diferentes niveles socioeconómicos y personas con capacidades diversas.
- Pluralidad cultural y lingüística: El sistema debe ser capaz de comprender y adaptarse a las diversas formas en que las personas expresan sus necesidades. En este contexto, utilizaremos un lenguaje común que abarque a toda la sociedad española, dado que los modelos de lenguaje actuales no logran representar la diversidad lingüística existente.
- Accesibilidad: La aplicación debe ser accesible para personas con discapacidad o limitaciones tecnológicas, especialmente en el caso de personas de la tercera edad.
- Lenguaje inclusivo y adaptado a la audiencia: Las respuestas generadas deben ser inclusivas y adecuarse a personas con diferentes niveles de educación, de modo que sean comprensibles para todos.

1.5. Descripción de la metodología empleada en el desarrollo del proyecto

Para abordar el desarrollo del proyecto se opta por utilizar la metodología Waterfall separando el desarrollo de 5 módulos diferentes:

- Desarrollo del módulo web scraping.

- Desarrollo almacenamiento en bases de datos vectoriales: Identificación de una base de datos adecuada.
- Desarrollo e integración con un Open LLM. Identificación de un Open LLM que sirva para el propósito de este proyecto.
- Desarrollo de aplicación web.
- Evaluación del modelo mediante RAGAS.

Estos cinco módulos se implementarán inicialmente con los aspectos básicos, con el objetivo de desarrollar un MVP (Minimum Viable Product) que cumpla con los requisitos mínimos. En iteraciones sucesivas durante la fase de desarrollo, se mejorará y perfeccionará cada uno de los módulos con el fin de alcanzar el producto final.

Se puede afirmar que se sigue una metodología Waterfall con refinamiento, ya que se establecen fases que tienen como objetivo principal obtener un MVP, y en las fases posteriores se busca afinar cada vez más la entrega. Los objetivos secundarios que no puedan ser abordados en este trabajo se dejarán como tareas futuras.

1.6. Planificación y plan de investigación del proyecto

Recursos necesarios

Para la realización y materialización del proyecto se han empleado los siguientes elementos:

Hardware	
PC sobremesa	Intel(R) Core(TM) i5-9600K CPU @ 3.70GHz
Tarjeta gráfica	NVIDIA GeForce RTX 4070 SUPER
Memoria	16GB RAM

Tabla 1: Recursos Hardware

Software	
Sistema Operativo	Windows 11 Pro 23H2
Gestor extensiones Python	Anaconda
Editor de desarrollo	Visual Studio Code
Extensiones VS Code	Python Python Debugger Jupyter
Python	Python 3.9.20
Principales Librerías Python	langchain, ollama, openai, faiss-cpu, chromadb, Sentence-Transformer, Request, torch con soporte de CUDA, ragas y Streamlit. Nota: La lista completa de referencias se puede ver en el repositorio de código fuente en el fichero requirements.txt

Tabla 2: Recursos Software

Tareas y Planificación

A continuación, se presenta un resumen detallado de las fases, las principales tareas y los hitos del proyecto, junto con un diagrama de Gantt también resumido. Las entregas de documentación en el campus están identificadas como hitos clave.

FASE DE PLANIFICACIÓN

Definición del tema, objetivos y análisis de herramientas

- HITO: Definición del TFM: enunciado y entrega (M1)
- HITO: Entrega comité ética y convenios (M1)
- HITO: Estado del arte: enunciado y entrega de la actividad (M2)

ANÁLISIS DE REQUISITOS

Revisión de requisitos y especificación de casos de uso

DISEÑO DEL SISTEMA

Diseño detallado de cada uno de los módulos del sistema

DESARROLLO DEL PROYECTO

Desarrollo y pruebas unitarias

FASE DE PRUEBAS

Pruebas de integración y UAT.

DOCUMENTACIÓN

Documentación del proyecto

REVISIÓN FINAL DEL TFM

Redacción del documento final del TFM, revisión y corrección

- HITO: Redacción de la memoria: entrega preliminar (M4)
- HITO: Redacción de la memoria: entrega final (M4)
- Preparación del video TFM
- HITO: Presentación audiovisual del trabajo (M4)

DEFENSA DEL TFM

Preparación de la presentación y ensayos de defensa.

- HITO: Presentación ante el tribunal

Planificación resumida - Diagrama de Gantt

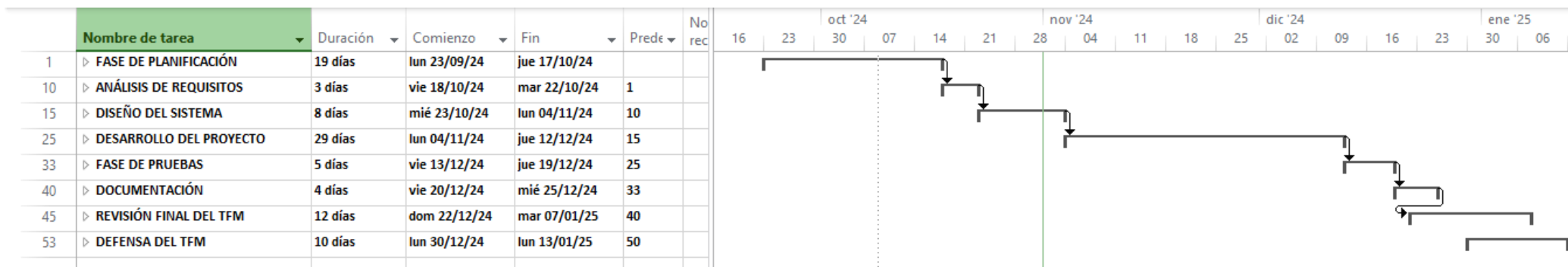


Figura 1: Diagrama Gantt Resumido

En el Anexo 12.1 se presenta un listado detallado de las fases y tareas, así como el diagrama de Gantt completo.

2. Estado del Arte

2.1. Contexto

La adopción de modelos de lenguaje de gran tamaño (Large Language Models, LLMs) ha transformado notablemente la manera en que las tecnologías de procesamiento de lenguaje natural (PLN) interactúan con el usuario y extraen información de fuentes diversas. Estos modelos han demostrado una habilidad sin precedentes para comprender, generar y sintetizar texto humano, permitiendo el desarrollo de aplicaciones avanzadas como asistentes virtuales, sistemas de recomendación o chatbots, entre otros. Sin embargo, hay un largo recorrido en lo que respecta al uso de LLMs en ámbitos de información privada y en dominios específicos, como es el caso de uso que se presenta aplicar en el presente trabajo centrado en la extracción y consulta de información de organismos públicos.

En este contexto, este trabajo se centra en la aplicación de técnicas RAG y el uso de modelos de lenguaje de código abierto para crear un sistema de asistencia conversacional que facilite la búsqueda de información relevante sobre subvenciones y ayudas públicas del Gobierno de España.

La metodología para la revisión bibliográfica ha consistido en hacer una búsqueda exhaustiva de publicaciones científicas que nos permitan conocer la evolución de estas tecnologías de inteligencia artificial y procesamiento del lenguaje natural a lo largo del tiempo, así como conocer su estado de la actualidad. Se han seleccionado temáticas y tecnologías clave sobre la recuperación de información (mediante técnicas de web scraping), codificación y procesamiento de la información (mediante técnicas de embedding), almacenamiento y consulta de esta información (mediante bases de datos vectoriales) y generación de respuestas de esta (mediante grandes modelos del lenguaje (LLMs)).

2.2. Recuperación de Información y Web Scraping

2.2.1. Contexto y evolución histórica

La recuperación de información (Information retrieval, IR) constituye un área de investigación centrada en el diseño y desarrollo de sistemas destinados a buscar y extraer información desde diversos orígenes de datos, abarcando tanto información estructurada como no estructurada.

Se entiende por información estructurada aquella que permite la identificación de datos específicos, al estar almacenada en bases de datos, webs o en documentos organizados de manera clara y precisa, facilitando así su recuperación y análisis. En contraste, la información no estructurada procede de fuentes sin una estructura definida, tales como registros de eventos (logs) o documentos sin un formato organizado o etiquetas explícitas. La gestión y análisis de este tipo de datos presentan desafíos adicionales en comparación con la información estructurada, dada la ausencia de un orden preestablecido que permita la identificación directa de datos específicos. Los sistemas de recuperación de información intentan satisfacer las necesidades informativas del usuario, procesando esta información y almacenándola de una forma persistente para su posterior explotación.

El campo de la recuperación de la información ha evolucionado desde los primeros modelos basados en vectores (Salton et al., 1975) donde los autores nos presentan una representación matemática de documentos y consultas como vectores en un espacio multidimensional, donde cada dimensión corresponde a un término. La relevancia entre un documento y una consulta se mide mediante la similitud del coseno entre sus vectores, lo que permite la recuperación basada en términos ponderados. Este modelo sentó las bases para métodos de búsqueda eficientes y sigue siendo un pilar en los sistemas modernos de recuperación.

Posteriormente en (Baeza-Yates y Ribeiro-Neto, 1999) se abordan tanto los fundamentos como los avances modernos en esta disciplina. En este libro los autores exploran técnicas de búsqueda y clasificación, así como cuestiones sobre escalabilidad y rendimiento en grandes volúmenes de datos. También se abordan otros temas importantes como el modelado de documentos, la interfaz de usuario y la evaluación de sistemas de recuperación, dándonos una visión integral como base para investigadores y profesionales del campo.

En este otro artículo (Singhal, 2001) se ofrece una visión general de la evolución y los desafíos de la recuperación de información en la era digital. Plantea una transición hacia sistemas basados en el aprendizaje automático, al igual que en (Manning, 2009). En ambos artículos se realiza una introducción exhaustiva de los conceptos y métodos en la recuperación de información, desde los modelos clásicos hasta las técnicas avanzadas basadas en aprendizaje automático. En este segundo (Manning, 2009) se abordan temas como el procesamiento de lenguaje natural, la clasificación y el clustering de documentos, la indexación y los motores de búsqueda. Esta obra tiene un enfoque más pedagógico y es usada como recurso formativo en el campo de la recuperación de información.

Todos estos trabajos contribuyeron a la consolidación de la recuperación de información, abordando sus fundamentos y desafíos, y estableciendo metodologías que hoy en día aún siguen siendo relevantes.

2.2.2. Web scraping

El web scraping es una técnica automatizada para extraer información de sitios web, ampliamente utilizada en aplicaciones como la recolección de precios en e-commerce, la monitorización de tendencias en redes sociales y el análisis financiero. Su relevancia ha crecido debido a la gran cantidad de información en la web que no está disponible a través de APIs, las cuales ofrecen acceso seguro y controlado a los datos que los sitios deciden exponer. Cuando no existen APIs o estas no proporcionan la información requerida, el web scraping se convierte en una alternativa eficaz para obtener datos estructurados y no estructurados desde diversas fuentes, siguiendo una arquitectura que incluye varias etapas iterativas que se muestran en la Figura 2.

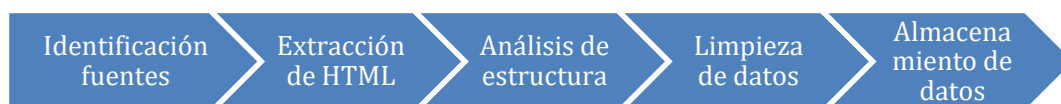


Figura 2: Etapas Web Scraping

Haciendo un repaso de los trabajos previos relacionados con el web scraping vemos que uno de los primeros trabajos en explorar y clasificar las herramientas de extracción de datos web es (Laender et al., 2002). En este artículo se destaca las dificultades técnicas de extraer datos de la web debido a la heterogeneidad de las estructuras HTML y al hecho de que la información en la web no está diseñada para ser extraída en masa. Los autores clasifican los métodos en dos enfoques: los basados en reglas, conocidos como wrappers, y los métodos basados en aprendizaje automático. La principal limitación identificada para los wrappers es su alta dependencia de la estructura de las páginas, lo que requiere constantes actualizaciones cuando los sitios web cambian. Los métodos de aprendizaje automático ofrecen más adaptabilidad, aunque son limitados por la necesidad de grandes cantidades de datos etiquetados y la dificultad para generalizar entre diferentes sitios web. Además, subrayan la necesidad de métodos que equilibren la precisión con la capacidad de adaptación y mantenimiento, una cuestión que sigue siendo relevante a medida que la web se expande y se vuelve más compleja.

En (Arasu y Garcia-Molina, 2003) proponen el sistema DEPTA (Data Extraction and Processing of Tree-like Attributes) para extraer datos estructurados de páginas web no estructuradas, especialmente aquellas que presentan información en forma de listas o tablas. DEPTA utiliza el modelo de objetos de documento (Document Object Model, DOM), que es una interfaz de programación que nos permite crear, cambiar, o remover elementos de un documento, para representar la estructura de la página como un árbol y localizar patrones repetitivos en las webs basadas en lenguajes de marcas de hipertexto (HyperText Markup Language, HTML) identificando subsecciones que contienen datos relevantes y organizándolas en formato tabular. Este enfoque permite una extracción de datos

más precisa y automatizada, adaptándose a variaciones en el diseño de las páginas sin requerir reglas manuales específicas para cada sitio, lo cual representa un avance significativo en la extracción de datos web.

Un análisis más amplio es presentado en (Chang et al., 2006) donde clasifican las técnicas de extracción en tres categorías: sistemas basados en wrappers, métodos de aprendizaje automático, y enfoques híbridos. Los sistemas basados en wrappers siguen siendo útiles para sitios con estructuras estables, mientras que el aprendizaje automático permite la generalización en múltiples sitios, aunque a costa de una mayor complejidad y requerimientos de datos. Los autores reconocen la importancia de los sistemas híbridos, que integran reglas específicas con aprendizaje automático para lograr mayor precisión y flexibilidad. Al explorar los desafíos, este estudio destaca las limitaciones de escalabilidad y el alto coste de mantenimiento de los extractores, aspectos que motivan la búsqueda de sistemas más autónomos y adaptativos.

Por su parte, (Ferrara et al., 2014) amplía la perspectiva al discutir tanto las técnicas de extracción como sus aplicaciones. Los autores identifican tres tipos de técnicas: wrappers manuales y semiautomáticos, métodos basados en aprendizaje automático y NLP, y métodos híbridos avanzados. Este artículo profundiza en el uso de técnicas avanzadas como las redes neuronales para mejorar la precisión y adaptabilidad de los sistemas de extracción, anticipando la creciente influencia de la inteligencia artificial en este campo. Los autores también analizan aplicaciones prácticas, como la minería de opiniones y la vigilancia de mercados, y subrayan los desafíos éticos y legales, como el respeto por la privacidad y los derechos de propiedad de los datos. Este enfoque centrado en aplicaciones es relevante para el desarrollo de sistemas de extracción que puedan integrar y analizar grandes volúmenes de datos de múltiples fuentes, algo esencial en sectores como el comercio, la investigación y la analítica empresarial.

Concluyendo, ya sea mediante una técnica de web scraping o mediante un API el objetivo es que la información extraída pueda almacenarse e integrarse en sistemas de análisis de datos y explotación de información, permitiendo procesarla de forma centralizada. Esto resulta especialmente valioso en el desarrollo de aplicaciones que requieren grandes volúmenes de datos, como los modelos de inteligencia artificial, entre ellos los LLMs y otros algoritmos de aprendizaje automático.

2.2.3. Consideraciones éticas y legales

El web scraping plantea desafíos legales y éticos, ya que algunos sitios web prohíben explícitamente la extracción de datos sin consentimiento, y en otros, los datos pueden estar protegidos por derechos de autor. Como fase previa al proceso de web scraping se debe realizar una revisión exhaustiva de los términos de uso

de cada sitio web y, en caso de estar disponible, la utilización de interfaces de programación de aplicaciones (Abstract Programming Interfaces, APIs) oficiales que nos garanticen un acceso seguro, fiable y autorizado a los datos.

2.3. Procesamiento de Lenguaje Natural y Grandes Modelos del Lenguaje

2.3.1. Introducción al NLP y evolución hasta los LLMs

2.3.1.1. Contexto general

El Procesamiento del Lenguaje Natural (NLP) es una rama de la inteligencia artificial (IA) que busca capacitar a las computadoras para comprender, interpretar, generar y responder al lenguaje humano. Este campo combina conocimientos de lingüística, ciencia de datos, aprendizaje automático y redes neuronales para resolver problemas relacionados con el texto y el habla.

El lenguaje es una de las herramientas más complejas y fundamentales de la comunicación humana. Procesar y analizar grandes cantidades de texto o habla de manera eficiente es crucial en una era donde los datos no estructurados, como correos electrónicos, publicaciones en redes sociales y documentos, representan una gran parte de la información generada. El NLP permite a las máquinas extraer valor de esta información, automatizar tareas repetitivas y facilitar la interacción humano-computadora.

Entre las áreas más importantes de aplicación se encuentran:

- Análisis de texto para la identificación de sentimientos, clasificación de documentos o análisis de opiniones.
- Traducción automática entre idiomas.
- Chatbots y asistentes virtuales que posibilitan la interacción conversacional.
- Reconocimiento de voz para la conversión a texto.
- Extracción de información para identificar entidades y relaciones relevantes en documentos.
- Sistemas de preguntas y respuestas.

Es ampliamente reconocido que el lenguaje natural es inherentemente ambiguo, diverso y dependiente del contexto, lo que plantea una serie de desafíos significativos, entre los cuales destacan los siguientes:

- Ambigüedad léxica y sintáctica: una misma palabra puede poseer múltiples significados dependiendo del contexto, y una oración puede ser interpretada de diversas maneras según su estructura.
- Lenguaje figurado y coloquial: el uso de metáforas, sarcasmo o expresiones informales requiere de un nivel avanzado de comprensión contextual y semántica para su adecuada interpretación.

2.3.1.2. Evolución hasta los LLMs

Desde sus comienzos el NLP ha experimentado una notable evolución, transitando desde enfoques iniciales basados en reglas hasta los sofisticados modelos neuronales que predominan en la actualidad.

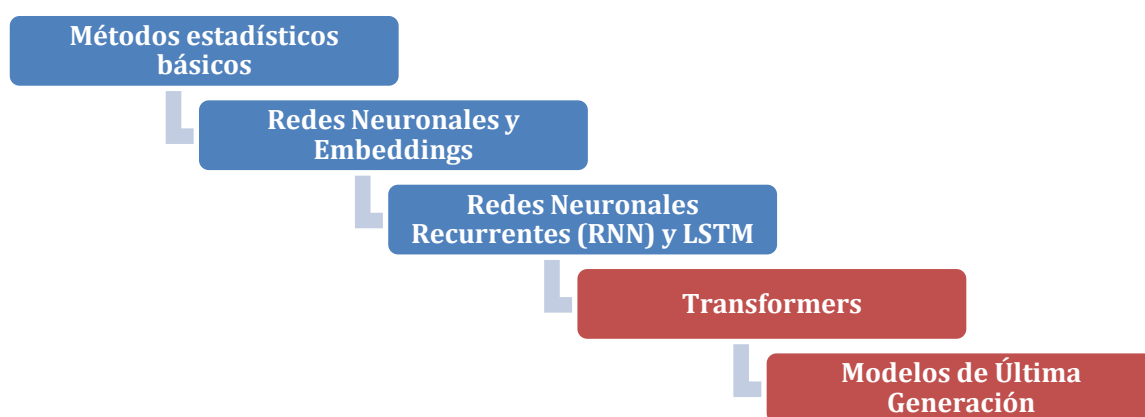


Figura 3: Evolución técnicas PLN

Décadas de 1950 a 1980: Predominio de enfoques basados en reglas manuales y métodos estadísticos rudimentarios. Durante este período, las primeras aproximaciones al Procesamiento del Lenguaje Natural se fundamentaron en reglas gramaticales elaboradas manualmente. Entre los desarrollos más representativos se encuentran las gramáticas generativas propuestas por (Chomsky, 1956) y el chatbot ELIZA que se presenta en (Weizenbaum, 1966) y considerado un hito temprano en la interacción humano-computadora.

Décadas de 1990 a 2000: Avance hacia modelos estadísticos sofisticados y la incorporación del aprendizaje automático. Con el tiempo, el crecimiento exponencial de los datos digitales durante las décadas de 1980 y 1990 marcó un hito significativo en el NLP. En esta etapa surgieron modelos estadísticos que aprovechaban grandes corpus textuales para modelar relaciones lingüísticas complejas. Surgieron enfoques basados en métodos de aprendizaje supervisado como en (Church, 1989) donde se introduce el uso de enfoques estadísticos para

el etiquetado POS (Part-of-Speech), basado en modelos de probabilidades. También se popularizaron algoritmos como los modelos de Markov. Uno de los artículos más conocidos basado en los modelos de Markov es (Rabiner, 1989) donde se utiliza en tareas de etiquetado POS.

Posteriormente, la incorporación de algoritmos de aprendizaje automático (Machine Learning, ML) marcó un avance significativo en el campo del NLP. Estos algoritmos introdujeron modelos capaces de generalizar patrones lingüísticos complejos a partir de datos anotados, superando las limitaciones de los enfoques basados únicamente en reglas o estadísticas simples. Esta transición permitió abordar tareas lingüísticas con mayor precisión y flexibilidad, impulsando el desarrollo de aplicaciones más robustas y adaptativas. En este sentido aparecen referencias bibliográficas fundamentales en el campo del NLP como en (Manning y Schütze, 1999) donde utilizan métodos estadísticos, centrando su atención en cómo aplicar modelos probabilísticos (como los modelos de Markov y N-grama) para resolver tareas de NLP. En este trabajo se introducen conceptos fundamentales como la estimación de parámetros, la modelización de texto y los Modelos Ocultos de Markov (HMM), que son esenciales para tareas como el etiquetado POS, análisis sintáctico y desambiguación de palabras. Posteriormente en (Jurafsky, 2000) se abordan temas relacionados con la comprensión y producción del lenguaje por parte de las máquinas introduciendo conceptos básicos de lingüística computacional, morfología, sintaxis y la semántica. También se examinan otros temas como el reconocimiento de voz y la traducción automática.

Década de 2010: Las primeras redes neuronales y embeddings fueron usadas por los primeros modelos de procesamiento de lenguaje, como Word2Vec en (Mikolov, 2013) y GloVe en (Pennington et al., 2014), donde se representaban palabras en espacios vectoriales permitiendo relaciones de similitud entre términos limitando a un solo contexto por palabra. Si bien fueron innovadores en su momento, estos modelos no podían capturar contextos complejos o polivalentes.

Posteriormente, las redes neuronales recurrentes (Recurrent Neural Network, RNN) en (Rumelhart et al., 1975) y los modelos de memoria a largo plazo en (Long Short-Term Memory, LSTM) en (Hochreiter, 1997) introdujeron mejoras en la generación de secuencias y en tareas de lenguaje natural al incorporar memoria y contexto a lo largo de las oraciones. Sin embargo, su capacidad de procesamiento secuencial limitaba la longitud de texto que podían manejar, y tendían a perder el contexto en oraciones largas o textos complejos.

2017 al presente: La revolución en los modelos de lenguaje comenzó con la introducción del enfoque de transformers en (Vaswani, 2017) que supuso un cambio de paradigma del NLP. Los transformers eliminaron la dependencia

secuencial de los modelos recurrentes, utilizando una arquitectura basada en mecanismos de atención que permite al modelo identificar qué partes de una oración son relevantes para una palabra en particular sin necesidad de procesar el texto en orden. Este cambio estructural permitió el entrenamiento de modelos con mayor capacidad para capturar contextos complejos y largas dependencias, acelerando la capacidad de procesamiento y mejorando la precisión en tareas de NLP. En la sección 2.3.5.2 del presente documento se detalla el funcionamiento de estos transformers.

Esta evolución dio paso a nuevas arquitecturas y modelos revolucionarios, como BERT (Devlin et al., 2018) y GPT, que introdujeron un aprendizaje contextualizado más profundo. Estas innovaciones mejoraron significativamente el rendimiento en tareas complejas como la respuesta a preguntas, la generación de texto y el análisis semántico.

Posteriormente, modelos de gran escala como GPT-3 (Brown et al., 2020) y LLaMA (Large Language Model Meta AI) (Touvron et al., 2023) establecieron el estado del arte en el campo. Estos modelos, entrenados con volúmenes masivos de datos y con miles de millones de parámetros, han demostrado una capacidad notable para abordar tareas avanzadas de NLP con adaptaciones mínimas, destacándose por su flexibilidad y precisión en una amplia gama de aplicaciones.

En la sección 2.3.5.3 del presente documento se hace una descripción más pormenorizada de estos y otros modelos.

2.3.2. Representaciones vectoriales (Embeddings)

2.3.2.1. Contexto

En el ámbito del NLP, el entendimiento y la generación de texto han experimentado avances significativos impulsados por dos conceptos clave: los *embeddings* y los LLMs. Estos enfoques han permitido a los sistemas de inteligencia artificial capturar de manera más precisa y contextual tanto el significado como las relaciones inherentes al lenguaje humano. Como resultado, han revolucionado una amplia variedad de aplicaciones, que incluyen motores de búsqueda, asistentes conversacionales, traducción automática y análisis de sentimientos, entre otras.

Los embeddings son técnicas que permiten transformar palabras, frases o documentos en vectores numéricos que capturan sus relaciones semánticas en un espacio matemático. Estas representaciones capturan relaciones semánticas y pueden ser utilizadas en tareas de clasificación, análisis de sentimientos o

búsqueda de información, entre otros. Por otro lado, los LLMs representan un avance significativo, ya que no solo generan embeddings, sino que también están diseñados para comprender, generar y manipular texto de manera contextual y coherente, permitiéndoles realizar tareas avanzadas de lenguaje, como generación de texto, traducción y respuesta a preguntas. Estos modelos, basados principalmente en la arquitectura de Transformers, cuentan con millones de parámetros que les permiten adaptarse a diversas tareas sin necesidad de ajustes complejos. Además, utilizan grandes volúmenes de datos para entrenarse, lo cual les permite aprender patrones lingüísticos y operar en una amplia variedad de aplicaciones sin requerir ajustes específicos para cada tarea.

En los siguientes capítulos, nos enfocaremos en explicar de manera detallada cada uno de estos 2 conceptos (embedding y LLMs), con el fin de justificar su aplicación en el presente trabajo. Además, en el Anexo 8.2 de este documento, se puede consultar un análisis exhaustivo de las diversas librerías y modelos de embeddings.

2.3.2.2. Definición

Como se ha adelantado en capítulos anteriores, los embeddings son representaciones vectoriales de datos en un espacio continuo de múltiples dimensiones. Estas representaciones permiten que datos como texto, imágenes o audio puedan ser representados en vectores, capturando relaciones semánticas y contextuales.

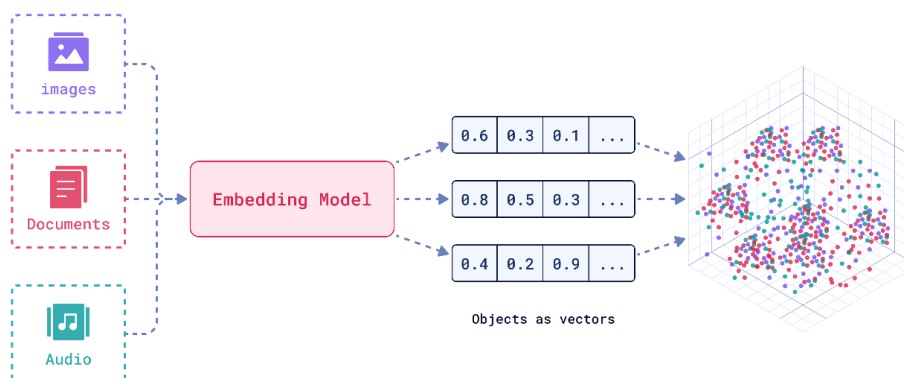


Figura 4: Diagrama Embeddings

En el contexto de los modelos de lenguaje, los embeddings permiten que los modelos comprendan y procesen texto al mapear palabras o frases en espacios vectoriales donde la proximidad refleja la similitud semántica. En la Figura 5 podemos ver un ejemplo de generación de embeddings a partir de cadenas de texto.

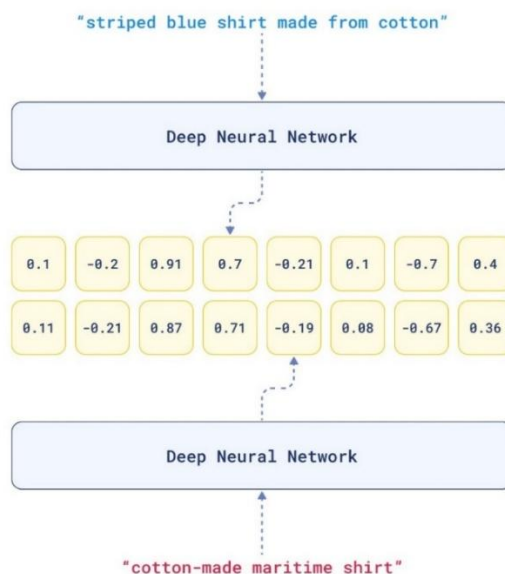


Figura 5: Diagrama Generación Embeddings

2.3.2.3. Técnicas de Generación de Embeddings

Existen diversas técnicas para la generación de embeddings, cada una con enfoques únicos para capturar las relaciones contextuales en el lenguaje. Algunas de las más destacadas son las siguientes:

- **Word2Vec** introducido en (Mikolov, 2013) es un modelo de redes neuronales de baja complejidad que utiliza dos enfoques principales: skip-gram y CBOW (Continuous Bag of Words). El modelo aprende a predecir palabras en función de su contexto (skip-gram) o a predecir el contexto basado en la palabra (CBOW), generando vectores que capturan relaciones semánticas entre términos.
- **GloVe** (Global Vectors for Word Representation) GloVe, es creado en (Pennington et al., 2014) y se basa en la matriz de co-ocurrencia de palabras, utilizando estadísticas globales de co-ocurrencia de palabras en un corpus de texto. Esta técnica busca maximizar la relación de palabras que co-ocurren frecuentemente.
- **FastText** En (Bojanowski et al., 2017) mejora Word2Vec al considerar subpalabras (n-gramas), permitiendo una mejor representación de palabras raras o desconocidas. Este enfoque es útil para lenguajes morfológicamente ricos y mejora la generalización de los embeddings.

Tanto Word2Vec como GloVe aprenden a comprender el contexto en el que se presentan las palabras, generando vectores de alta dimensionalidad que capturan propiedades semánticas y complejidades contextuales avanzadas de cada término, como se puede ver en la Figura 6.

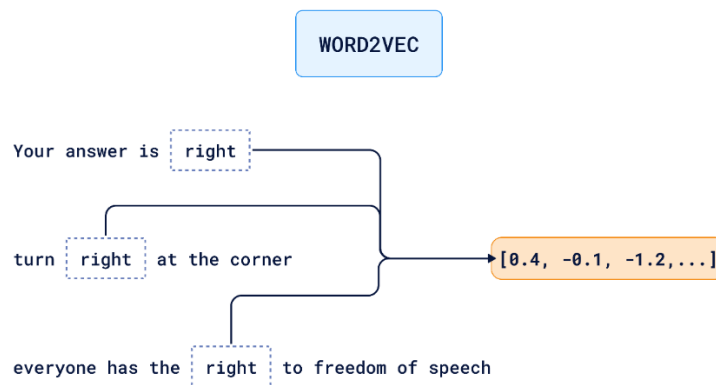


Figura 6: Diagrama Embeddings WORD2VEC

Sin embargo, estos modelos siguen teniendo limitaciones ya que representan cada palabra con un único vector. En caso concreto indicado en la Figura 6 significa que todos los matices de la palabra “right” se transforman en una única representación vectorial. Esa información no es suficiente para que los ordenadores comprendan completamente el contexto.

Posteriormente, la arquitectura basada en transformers propuesta en (Vaswani, 2017) revolucionó el procesamiento del lenguaje al posibilitar el análisis paralelo del texto mediante mecanismos de atención. Este avance dio lugar al surgimiento de modelos más sofisticados, como BERT (Bidirectional Encoder Representations from Transformers), presentado por Devlin et al. (2018), y las distintas versiones de GPT, incluyendo GPT (Radford, 2018), GPT-2 (Radford et al., 2019) y GPT-3 (Brown et al., 2020). Estos modelos marcaron un hito en la capacidad de los sistemas de inteligencia artificial para comprender y generar texto de manera más eficiente y precisa. El modelo comprende el uso específico de una palabra en su entorno y luego crea diferentes incrustaciones para cada una. Un ejemplo lo tenemos en la Figura 7 que ejemplifica cómo el modelo BERT representa la palabra “right” según su contexto codificado con diferentes embeddings.

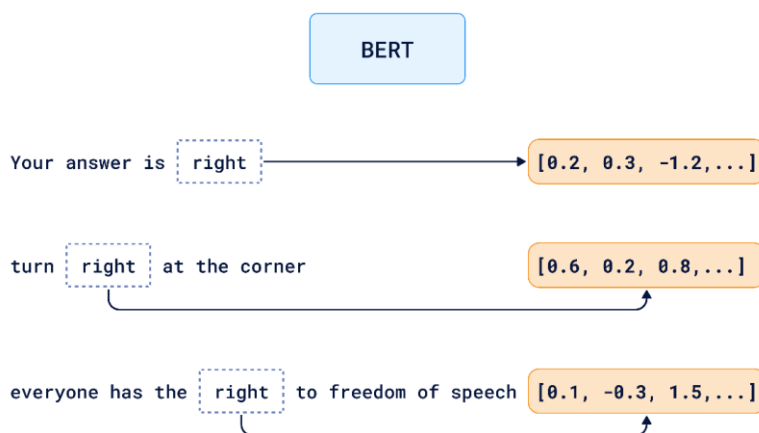


Figura 7: Diagrama Embeddings BERT

2.3.3. Almacenamiento de la información y Bases de datos vectoriales

2.3.3.1. Introducción

Una vez que los datos se indexan en embeddings, el siguiente paso consiste en persistirlos en una base de datos eficiente para su posterior recuperación. La selección adecuada de la base de datos es crucial para garantizar la escalabilidad y el rendimiento del sistema. Algunas de las opciones disponibles incluyen:

- Bases de datos vectoriales: Diseñadas específicamente para la búsqueda de similitudes, permiten una recuperación rápida de embeddings. Ejemplos incluyen Chroma, FAISS, Weaviate y Pinecone.
- Bases de datos basadas en grafos: Son ideales para tareas que requieren un análisis profundo del contexto y relaciones semánticas complejas. Un ejemplo prominente es Neo4j.
- Bases de datos clave-valor: Adecuadas para casos de uso de menor escala que implican un mapeo sencillo de clave a embedding vectorial.
- Bases de datos generales: Expansiones de bases de datos SQL/NoSQL tradicionales, como PostgreSQL, que incluyen extensiones vectoriales para gestionar embeddings.

2.3.3.2. Bases de datos Vectoriales

Las bases de datos vectoriales han ganado un papel fundamental en el campo de los LLMs y otras aplicaciones de inteligencia artificial, especialmente en técnicas RAG. Estas bases de datos permiten almacenar y consultar grandes volúmenes

de datos en formato vectorial, optimizando la rapidez y precisión de las consultas en tareas de búsqueda semántica.

Una base de datos vectorial es un sistema de almacenamiento diseñado específicamente para gestionar datos en formato vectorial, lo cual permite efectuar búsquedas semánticas eficaces mediante el uso de distancias y similitudes entre vectores. En el contexto de los LLMs, estas bases son especialmente útiles para almacenar representaciones de texto en el espacio vectorial, permitiendo que el modelo recupere información contextual relevante en tiempo real, aumentando así la precisión en tareas como respuesta a preguntas y generación de lenguaje.

Las bases de datos vectoriales constituyen un componente fundamental en los sistemas RAG y LLMs, facilitando búsquedas semánticas precisas y contextualizadas en amplios volúmenes de información. Este enfoque es esencial en aplicaciones de IA generativa que requieren una integración efectiva entre el modelo de lenguaje y los datos actualizados, como en chatbots, asistentes virtuales, motores de recomendación y búsqueda semántica avanzada.

A diferencia de las bases de datos relacionales o documentales tradicionales, las bases de datos vectoriales están diseñadas para almacenar, indexar y consultar datos en forma de vectores, que son representaciones numéricas de características extraídas de contenido textual, imágenes o audio. Estos vectores se almacenan en la base de datos capturando similitudes semánticas entre diferentes objetos al calcular la proximidad entre ellos en el espacio vectorial, generalmente mediante métricas de distancia, como la distancia euclidiana o el coseno de similitud.

En el Anexo 8.3 del presente documento se puede ver una comparativa de diferentes bases de datos vectoriales.

2.3.4. Grandes Modelos de Lenguaje (LLMs)

2.3.4.1. Contexto y transformers

Los LLMs representan un desarrollo avanzado en el PLN con la capacidad de comprender, generar y manipular texto de manera coherente y contextualizada. A diferencia de las librerías de embeddings, que se centran en representar palabras y frases como vectores para capturar relaciones semánticas, los LLMs no solo producen representaciones contextuales, sino que también poseen una capacidad significativamente superior para generar y entender el lenguaje de forma integral.

Los LLMs se fundamentan en arquitecturas basadas en transformers y están estructurados en numerosas capas. Cada capa procesa una secuencia de

vectores como entradas —un vector por cada palabra del texto proporcionado—, y va incorporando información adicional para desambiguar el significado de las palabras. Este proceso permite que el modelo prediga de manera más precisa lo que podría seguir en el texto, como se ilustra en la Figura 9.

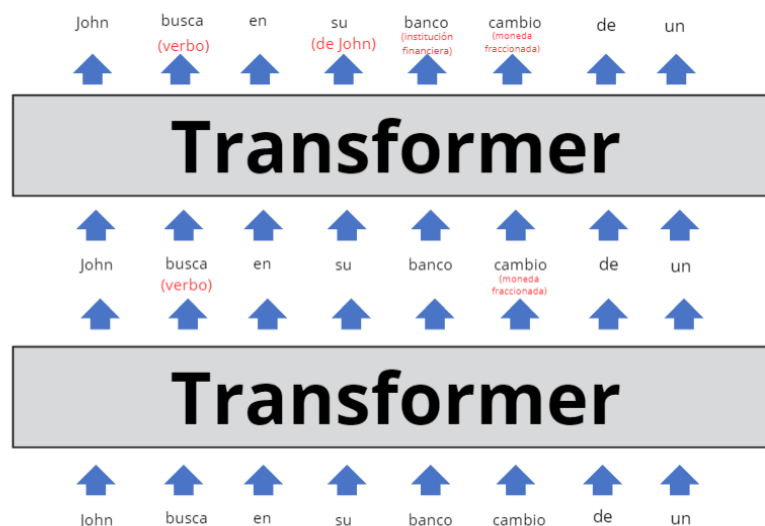


Figura 8: Capas Transformers

Los transformers representan una arquitectura revolucionaria en el campo del Procesamiento del Lenguaje Natural (NLP), presentada en el artículo "Attention is All You Need" (Vaswani et al., 2017). Este modelo se basa en un mecanismo de atención autorregresiva, que permite a las redes neuronales procesar secuencias completas de texto de manera simultánea, sin necesidad de iterar palabra por palabra como ocurría en los modelos recurrentes.

Los diversos usos y aplicaciones de los modelos de lenguaje de gran tamaño (LLMs) abarcan tareas tanto de comprensión del lenguaje, como el resumen y la traducción, como de generación de texto, que es el objetivo principal de este trabajo.

2.3.4.2. Comparativa de modelos históricos

A partir de la arquitectura de los transformers, han emergido grandes modelos de lenguaje (LLMs) que emplean miles de millones de parámetros y han sido entrenados con vastos volúmenes de datos textuales. En los primeros trabajos técnicos, el término "LLM" aún no era comúnmente utilizado; en su lugar, se hacía referencia al concepto de "GPT", desarrollado por OpenAI. El primer trabajo destacado de esta serie fue GPT-1 (Radford, 2018), seguido por GPT-2 (Radford et al., 2019) y GPT-3 (Brown et al., 2020). Estos estudios sentaron las bases para el desarrollo de otros modelos de lenguaje a gran escala por diversas empresas,

como BERT (Devlin et al., 2018) de Google, Gemini de Google, LLaMA (Touvron et al., 2023) de Meta, Claude de Anthropic y modelos de código abierto como Mistral 7B (Jiang et al., 2023) de Mistral. Estos avances han permitido la creación de sistemas cada vez más sofisticados en el campo del NLP.

En (Evolución de los modelos de lenguaje de gran escala, 2024) hay una comparativa donde se muestra de forma visual la Figura 9 que representa desde siete variables distintas:

- Fecha de lanzamiento: indicada por la posición vertical (2018 hasta 2023)
- Tipología del modelo: representada por las ramas de colores. Los colores indican si son modelos de solo codificación (encoder-only), solo decodificación (decoder-only), o modelos de codificación-decodificación (encoder-decoder).
- Código abierto (cuadrados sólidos) vs. Código cerrado (cuadrados huecos)
- Relaciones entre modelos: indicadas por las líneas.
- Cantidad de modelos por empresa (gráfico de barras apiladas).
- Empresa o institución desarrolladora: Indicado por los logotipos.
- Transformers y otros modelos: La distinción entre modelos basados en transformers y otros tipos de modelos.

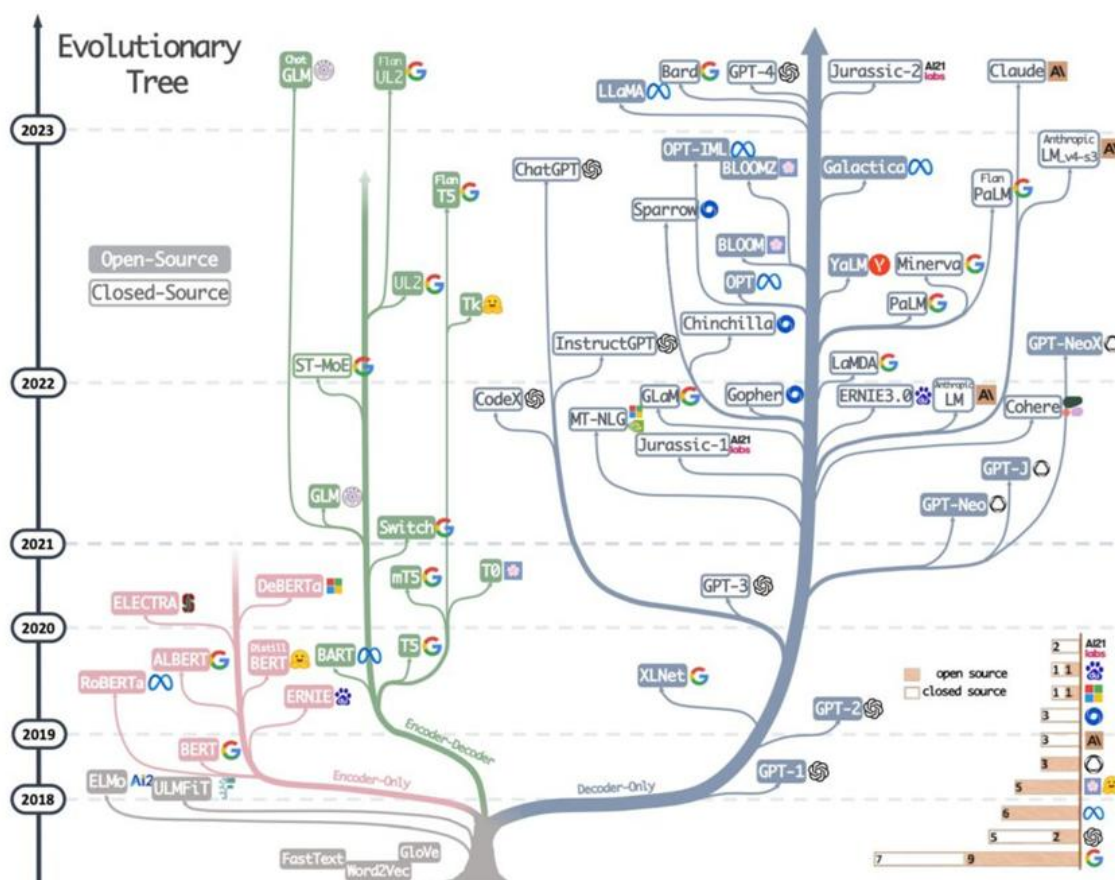


Figura 9: Diagrama LLMs desde distintas variables

2.3.4.3. Avances recientes: capacidades multimodales

En los últimos años, los modelos multimodales han emergido como una de las fronteras más avanzadas en inteligencia artificial, combinando capacidades de lenguaje, visión y, en algunos casos, audio para resolver problemas complejos que requieren una comprensión integrada de diferentes tipos de datos. Estas tecnologías están impulsadas por la convergencia de LLMs junto con arquitecturas especializadas para procesar imágenes. En los siguientes párrafos se muestran algunas de estas investigaciones lideradas por grandes compañías como OPENAI, Microsoft, Facebook o Salesforce.

Por una parte, OPENAI presenta CLIP en (Radford et al., 2021). Este modelo entrena representaciones conjuntas de imágenes y texto mediante aprendizaje contrastivo, alineándolos en un espacio latente compartido y logrando capacidades avanzadas de zero-shot learning para tareas de búsqueda y clasificación sin entrenamiento adicional.

Por otro lado, en (Huang et al., 2023) Microsoft presenta Kosmos-1 como un marco para alinear modalidades perceptuales como visión y audio con modelos de lenguaje grandes (LLMs), usando adaptadores ligeros que mejoran el razonamiento en tareas multimodales complejas al integrar visión y lenguaje en un espacio semántico unificado.

Flamingo es presentado en (Alayrac et al., 2022) por DeepMind como un modelo multimodal diseñado para tareas de few-shot learning que combina un modelo de lenguaje preentrenado con módulos de atención cruzada para procesar entradas visuales y textuales de manera contextual, superando modelos especializados en subtítulo, preguntas visuales y razonamiento multimodal.

En (Wang et al., 2023) Microsoft introduce BEiT, un modelo que aplica una técnica tipo BERT a imágenes tratándolas como un "lenguaje distinto", permitiéndole aprender representaciones visuales y lingüísticas ricas que mejoran la generalización en tareas multimodales.

Salesforce, en (Li et al., 2023) propone un marco eficiente que conecta codificadores visuales congelados como Vision Transformers con modelos de lenguaje grandes mediante un puente de adaptación, reduciendo costos computacionales y mejorando el aprendizaje multimodal en generación de subtítulos y preguntas visuales (*Salesforce Research, 2023*).

Meta AI en (Xu et al., 2023) amplía las capacidades de los modelos multimodales al introducir razonamiento explícito paso a paso, logrando mejoras en tareas

complejas como razonamiento visual y generación textual mediante el enfoque chain-of-thought.

Finalmente, en (Team et al., 2023) Google presenta DeepMind; una familia de modelos que combinan lenguaje, visión y razonamiento avanzado en una arquitectura modular y escalable, ideal para tareas como generación multimodal y planificación contextual en dominios complejos.

2.3.4.4. Importancia y aplicaciones

Los Modelos de Lenguaje de Gran Tamaño (LLMs) han supuesto un cambio paradigmático en el campo de la inteligencia artificial, ya que su capacidad para procesar, comprender y generar texto de manera coherente ha abierto una amplia gama de aplicaciones en casi todos los sectores. Estas aplicaciones están transformando industrias, facilitando procesos complejos y mejorando la eficiencia de diversas áreas. A continuación, se detallan algunas de las áreas clave en las que los LLMs están teniendo un impacto significativo:

- **Automatización del servicio al cliente:** Los LLMs están impulsando el desarrollo de chatbots avanzados y asistentes virtuales, que no solo pueden comprender las consultas de los usuarios, sino también generar respuestas contextualizadas y detalladas en tiempo real. Estos sistemas están mejorando la experiencia del cliente y reduciendo la carga de trabajo en las áreas de soporte, permitiendo una atención más eficiente y personalizada.
- **Educación:** En el ámbito educativo, los LLMs se utilizan para crear tutores virtuales capaces de ofrecer enseñanza personalizada a estudiantes de diferentes niveles y disciplinas. Además, estos modelos pueden generar materiales educativos adaptados a las necesidades específicas de los estudiantes, como resúmenes, ejercicios y explicaciones adicionales. Esto contribuye a un aprendizaje más dinámico y accesible.
- **Ciencias de la salud:** Los LLMs están siendo aplicados para analizar registros médicos, generar informes automatizados y asistir en el diagnóstico. Estos modelos pueden extraer información relevante de grandes volúmenes de datos clínicos y proporcionar recomendaciones que ayuden a los profesionales de la salud en su toma de decisiones. Además, los LLMs pueden ser utilizados para mejorar la eficiencia en la investigación biomédica, identificando patrones en los datos y acelerando el desarrollo de nuevos tratamientos.
- **Análisis financiero:** En el sector financiero, los LLMs están facilitando la generación de resúmenes automáticos de los mercados financieros, la detección de anomalías en los patrones de datos y la predicción de tendencias. Esto permite a las instituciones financieras tomar decisiones informadas con mayor rapidez y precisión. Los modelos también son útiles para el análisis de riesgos, la automatización de informes y la optimización de estrategias de inversión.

2.3.5. Técnicas avanzadas para optimizar LLMs

2.3.5.1. Fine-tuning: personalización de modelos para tareas específicas.

El fine-tuning es un proceso crítico en el ajuste de Modelos de Lenguaje de Gran Escala (LLMs) para adaptar sus capacidades generales de comprensión y generación de texto a tareas específicas. A diferencia del entrenamiento inicial, donde los LLMs se exponen a una amplia variedad de datos textuales con el fin de aprender patrones lingüísticos generales, el fine-tuning permite optimizar el modelo para una tarea concreta, como la clasificación de texto, el análisis de sentimientos o el resumen de documentos. Este proceso se ha convertido en una técnica fundamental para mejorar el rendimiento de los modelos de lenguaje en entornos prácticos, facilitando la creación de aplicaciones de IA de alto valor agregado.

La necesidad de ajustar los modelos para tareas específicas fue observada en trabajos pioneros como el de BERT en (Devlin et al., 2018) y sucesivas versiones de GPT, como GPT-2 en (Radford et al., 2019) y GPT-3 en (Brown et al., 2020). Estos modelos, aunque versátiles en tareas de lenguaje, mostraban un rendimiento significativamente superior en tareas específicas después de un proceso de fine tuning. En el caso de BERT se centró en análisis de sentimientos y clasificación de texto mostrando resultados sobresalientes frente a modelos tradicionales. Aplicar técnicas de fine-tuning a GPT-3 permitió mejorar su capacidad para tareas de generación de contenido, como resúmenes de documentos, redacción de artículos y respuestas a preguntas específicas. En otros casos como el modelo T5 presentado en (Raffel et al., 2020) se centró en el entrenamiento en idiomas específicos alcanzando un rendimiento competitivo respecto a otros modelos de traducción.

El proceso de fine-tuning de Modelos de Lenguaje de Gran Tamaño (LLMs) es crucial para adaptar modelos pre entrenados a tareas específicas o dominios particulares. Sin embargo, este proceso presenta diversos desafíos tanto técnicos como prácticos. A continuación, se desarrollan en mayor detalle los dos principales desafíos mencionados:

- **Sobrecarga Computacional:** El fine-tuning de LLMs implica ajustar un modelo pre entrenado que ya cuenta con miles de millones (o incluso billones) de parámetros. Esto conlleva una gran demanda de recursos computacionales.
- **Sobrefitting (Sobreajuste):** El sobreajuste, o overfitting, es otro desafío importante durante el fine-tuning de LLMs. Este fenómeno ocurre cuando un

modelo se ajusta excesivamente a los datos de entrenamiento específicos, perdiendo su capacidad de generalización a datos nuevos o no vistos.

2.3.5.2. Recuperación de Información Aumentada (RAG)

2.3.5.2.1. Introducción

Los LLMs tradicionales, como GPT-3 y Llama, están muy orientados en la generación de texto, pero tienen limitaciones en términos de precisión y actualización de la información. Al carecer de un mecanismo que les permita acceder a información externa, estos modelos generan respuestas basadas únicamente en los datos de entrenamiento, los cuales pueden estar desactualizados y carecer de detalles contextuales específicos. Como consecuencia, aunque las respuestas generadas son gramaticalmente correctas y coherentes, pueden resultar imprecisas, irrelevantes o incluso incorrectas, especialmente cuando se abordan temas altamente específicos o recientes.

Con el objetivo de mejorar esta problemática en (Lewis et al., 2020) se presenta la idea de integrar un modelo de recuperación de información con un modelo generativo de lenguaje para llevar a cabo tareas de procesamiento del lenguaje natural que requieren acceso a conocimientos actualizados o a fuentes externas. Este enfoque de recuperación de información aumentada (Retrieval-Augmented Generation, RAG) implica la utilización de un componente de búsqueda que recupera información relevante, complementado por un modelo generativo LLM que emplea dicha información para producir respuestas coherentes y precisas. Este proceso intermedio permite a los modelos de lenguaje acceder a bases de datos o conjuntos de documentos específicos, obteniendo información relevante para una consulta antes de proceder a generar la respuesta. En la Figura 10 podemos ver un diagrama de la arquitectura RAG.

RAG Architecture Model

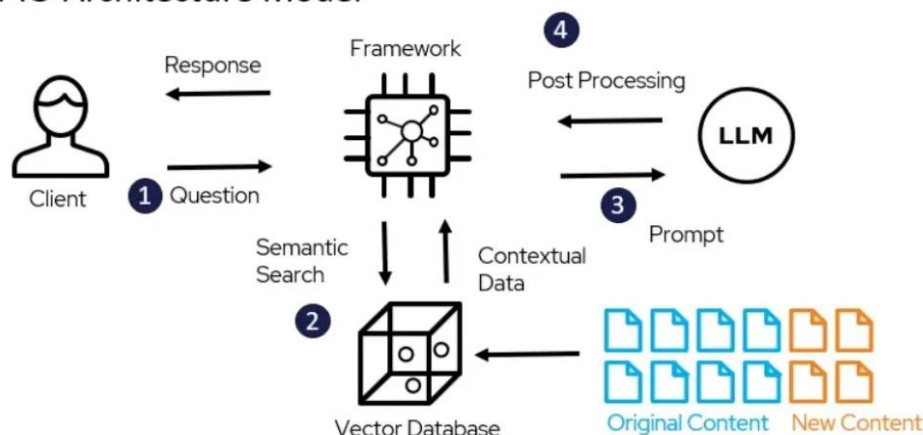


Figura 10: Diagrama RAG

En términos técnicos, RAG utiliza una arquitectura de doble etapa: un componente de recuperación que selecciona documentos relevantes y un modelo generativo que sintetiza una respuesta a partir de estos documentos, garantizando así mayor precisión y relevancia.

Las aplicaciones iniciales de RAG se encuentran en contextos de atención al cliente, generación de resúmenes de contenido extenso y sistemas de asesoramiento en áreas especializadas como el ámbito legal o médico, donde la precisión de la información y la actualidad son críticas. También ha sido implementado en sistemas de búsqueda inteligente, mejorando significativamente la precisión de las respuestas en comparación con los métodos tradicionales.

2.3.5.2.2. Ventajas

Esta técnica combina lo mejor de los modelos de lenguaje generativos con métodos de recuperación de información, no solo optimiza el proceso de búsqueda, sino que también mejora la calidad de la información presentada.

A continuación, se detallan las ventajas que ofrece:

- Precisión y actualización de la información: Utiliza un paso de recuperación de información en tiempo real, lo que permite que las respuestas generadas se basen en datos actuales y relevantes.
- Relevancia en las respuestas: Realiza una búsqueda previa en una base de datos o en un conjunto de documentos, las respuestas son más precisas en relación con la consulta específica del usuario.
- Reducción de alucinaciones: Al depender en parte de la información recuperada de la base de datos, RAG reduce la probabilidad de que el modelo genere información falsa o "alucinaciones".

- Flexibilidad para trabajar con dominios concretos: Permite adaptar el modelo a múltiples áreas de conocimiento mediante la incorporación de fuentes de información específicas.

2.3.5.3. Prompt Engineering

El Prompt Engineering es una técnica para diseñar y estructurar de manera cuidadosa los prompts (solicitudes o entradas) que se proporcionan a un LLM, con el objetivo de obtener respuestas más precisas, relevantes y útiles. En otras palabras, se trata de la formulación de las preguntas o instrucciones dadas al modelo para que este pueda interpretar y responder de la mejor manera posible. El diseño de prompts se ha convertido en un área crítica, ya que pequeños cambios en su redacción pueden afectar en gran medida las respuestas de los LLMs. Las técnicas de prompt tuning y prompt engineering están en auge para mejorar la efectividad sin necesidad de ajustar directamente el modelo LLM.

Investigaciones como (Brown et al., 2020) y (Liu et al., 2023) han demostrado la capacidad de los LLMs para realizar tareas complejas con pocos o ningún ejemplo de entrenamiento, gracias al diseño adecuado de prompts. Técnicas como el prompt tuning propuesto por Lester et al. (2021) permiten ajustar de manera eficiente los parámetros del modelo sin necesidad de un ajuste completo del mismo, haciendo el proceso más rápido y eficiente.

Asimismo, el uso de prompts que facilitan razonamientos estructurados, como el propuesto en (Wei et al., 2022), y el enfoque de (Kojima et al., 2022) para resolver problemas mediante razonamientos implícitos, amplían las capacidades de los modelos para abordar tareas complejas de forma más precisa. Además, el trabajo (Ouyang et al., 2022) que muestra cómo la retroalimentación humana puede ser incorporada a los prompts para mejorar la alineación de las respuestas con las intenciones del usuario, optimizando así la interacción con los modelos.

Este enfoque de personalización de prompts es fundamental para maximizar la efectividad de los LLMs, ya que permite adaptar estos modelos a tareas específicas con mayor precisión y eficiencia, mejorando la generación de respuestas y el razonamiento en tareas complejas.

2.3.6. Evaluación de modelos

2.3.6.1. Contexto

La evaluación de modelos es el proceso de medir el rendimiento de un modelo de machine learning o inteligencia artificial para asegurar que cumpla con los objetivos específicos para los cuales fue entrenado. Esta evaluación se realiza

mediante el uso de métricas cuantitativas y cualitativas que permiten comparar diferentes modelos y elegir el más adecuado para la tarea en cuestión.

Existen varios artículos que abordan la evaluación de diferentes modelos como en (Lewis et al., 2020) se destaca el beneficio de recuperar documentos relevantes antes de generar respuestas, aumentando significativamente la exactitud en tareas de generación de texto informativo en las respuestas. En este método se evalúan en benchmarks preguntas y respuestas y demuestra que los modelos RAG son especialmente útiles en dominios donde la información evoluciona rápidamente.

Posteriormente en (Thakur et al., 2021) presentan BEIR, un conjunto de datos exhaustivo y heterogéneo diseñado para evaluar modelos de recuperación de información en escenarios de zero-shot (modelos sin entrenamiento en tareas o dominios específicos). Este benchmark permite una evaluación comparativa precisa para modelos que buscan realizar recuperación de información en diferentes contextos y tipos de datos, incluso cuando no han sido ajustados previamente. Se cubren 18 conjuntos de datos diversos que abarcan tareas y dominios como medicina, tecnología, redes sociales y finanzas. Su enfoque permite evaluar modelos sin entrenamiento específico en cada tarea, midiendo su capacidad de generalización en distintos contextos. Con métricas como MRR o NDCG se facilita una evaluación estandarizada, promoviendo el desarrollo de modelos robustos y adaptables que pueden responder eficazmente a nuevos dominios sin ajustes previos. Este benchmark ha sido fundamental para avanzar en la creación de modelos más generalizables y útiles en aplicaciones prácticas de recuperación de información.

Otra aproximación es la que realizan en (Srivastava et al., 2022) donde examina las capacidades de LLMs mediante métricas detalladas de rendimiento y extrapola su potencial para tareas complejas, incluyendo las que requieren razonamiento avanzado y recuperación de conocimientos. La evaluación sugiere un enfoque de "stress testing" o pruebas intensivas en LLMs, lo cual también es aplicable en la evaluación de modelos RAG.

2.3.6.2. Evaluación de RAG

La evaluación de modelos RAG (Retrieve and Generate) se centra en medir la capacidad del sistema para integrar adecuadamente la recuperación de información y la generación de respuestas precisas y relevantes. La evaluación de un modelo RAG debe abordar varios aspectos clave, entre los que se incluyen:

- Precisión de la Recuperación: Evalúa qué tan relevante es la información recuperada para la pregunta planteada. Se mide a través de métricas como la

precisión o el recall de los documentos o fragmentos de información recuperados.

- **Calidad de la Generación:** Mide la capacidad del modelo para generar respuestas coherentes, precisas y bien formuladas basadas en la información recuperada. Esto se evalúa utilizando métricas como la coherencia, fluidez y exactitud.
- **Pertinencia de la Respuesta:** Se refiere a la relevancia de la respuesta generada en relación con la consulta original, considerando la exactitud y la cobertura de la información.
- **Evaluación Automática vs. Evaluación Humana:** Además de las métricas automáticas, como BLEU o ROUGE para la comparación de n-gramas, es importante la evaluación humana para juzgar la relevancia y la calidad general de las respuestas generadas en contextos más complejos.

La combinación de estas métricas permite una evaluación integral de los sistemas RAG, identificando tanto los puntos fuertes como las áreas de mejora en la integración entre recuperación y generación de texto.

Existen varios frameworks que permiten realizar una evaluación de sistemas RAG como, por ejemplo, los siguientes:

- **LangChain Evaluation:** es una herramienta diseñada para evaluar aplicaciones y pipelines desarrollados con el framework LangChain, enfocándose en tareas de recuperación y generación. Permite evaluar automáticamente aspectos como relevancia, factualidad y coherencia, e incluso incluye opciones para integrar evaluaciones humanas.
- **OpenAI Evals:** Inicialmente creado para modelos como GPT, pero es compatible con modelos open-source. Ofrece métricas estandarizadas y permite realizar evaluaciones automáticas y manuales en diferentes tipos de tareas.
- **Retrieval-Augmented Generation Analytics and Scoring (RAGAS):** Se centra en métricas clave como factualidad, relevancia y completitud, además de detectar alucinaciones generadas por el modelo. Es fácil de integrar con frameworks como LangChain y bases de datos vectoriales, siendo especialmente útil en proyectos avanzados que requieren optimizar tanto la recuperación como la generación de manera iterativa.

Para la evaluación del sistema RAG de este trabajo se va a utilizar RAGAS y en el anexo 8.4 del presente documento se hace una descripción detallada del mismo y de las métricas que se utilizan.

2.4. Desafíos y Consideraciones Éticas

2.4.1. Centralizar los aspectos técnicos, éticos y sostenibles.

Las técnicas RAG ofrecen grandes ventajas en tareas de búsqueda y generación de texto, pero también presentan algunos desafíos y riesgos que debemos abordar desde múltiples perspectivas, algunas de ellas son: técnicas, éticas / sociales, sostenibilidad y diversidad. El análisis y tratamiento de estas perspectivas o factores es esencial para asegurar que estas herramientas no solo sean efectivas, sino también responsables. En los siguientes puntos se abordan cada uno de estos factores.

Factores Técnicos

- Calidad y actualización de los datos. Se hace muy importante mantener la calidad, integridad y actualización de los datos. Si las fuentes de datos están desactualizadas o contienen errores, las respuestas generadas pueden no ser correctas o estar desactualizadas.
- Escalabilidad y rendimiento. La recuperación de información y la generación de respuestas deben adaptarse al volumen de datos. La falta de escalabilidad de estos puede afectar a los resultados y usabilidad de la herramienta.
- Eficiencia computacional y requerimientos de Hardware. Los sistemas basados en RAGs y LLMs requieren de recursos computacionales significativos. Se debe hacer un trabajo exhaustivo para elegir correctamente los recursos necesarios para no elevar los costes.

Factores éticos / sociales

- Privacidad y seguridad. El uso y almacenamiento de datos puede plantear desafíos y riesgos sobre la privacidad de estos, sobre todo si se trabaja con información confidencial o sensible.
- Transparencia y comprensibilidad. Se debe fomentar una total transparencia en cómo los modelos de lenguaje generan respuestas, explicar la procedencia de la información y cómo se generan las respuestas para que los usuarios tengan toda la información necesaria.
- Sesgos y desinformación. Los modelos de lenguaje y las técnicas de recuperación están expuestos a sesgos con base en los datos con los que han sido entrenados. Se hace indispensable identificar y mitigar, dentro de lo posible, estos sesgos en las respuestas generadas con el fin de evitar discriminar o desinformación a ciertos colectivos o grupos y esto se hace más preocupante en áreas más sensibles, como, por ejemplo, sanidad, finanzas o justicia.

Factores de sostenibilidad

- Consumo energético y optimización de los modelos. Estas herramientas de procesamiento de grandes volúmenes de datos requieren de un alto consumo de energía. Siempre que sea posible, se debe optar por la utilización de modelos especializados y optimizados en la necesidad que se requiera con el objetivo de minimizar ese consumo de energía y ayudar a reducir la huella de carbono.
- Ciclo de vida de los datos. Gestionar de forma sostenible los datos implica desde la obtención y almacenamiento hasta la eliminación segura de aquellos que ya no son relevantes. Con este planteamiento se consigue una reducción del coste de almacenamiento y procesamiento de los datos, un menor impacto ambiental y minimizar el riesgo de problemas de seguridad.

Factor de Diversidad

- Inclusión de múltiples perspectivas. Los orígenes de datos deben estar basados en fuentes y conjuntos de datos diversos, con el objetivo de evitar respuestas limitadas a una única perspectiva cultural, regional o de género.
- Accesibilidad y usabilidad. La accesibilidad y usabilidad de estas herramientas se deben diseñar para que todos los usuarios puedan acceder a ellas sin barreras tecnológicas, lingüísticas o económicas.

2.5. Resumen y aplicabilidad al trabajo actual

El uso de sistemas de RAG ha demostrado ser una herramienta eficaz para mejorar la precisión y relevancia en aplicaciones de procesamiento de lenguaje natural en dominios especializados. No obstante, está en constante evolución, y existen diversas áreas de investigación en la actualidad para continuar mejorando la evolución de la tecnología y su aplicación en diferentes ámbitos:

- Integración de datos en tiempo real: Incorporar actualizaciones en tiempo real permitiría que el sistema RAG pueda adaptarse ante la actualización de datos.
- Mejora en la personalización de respuestas: Las técnicas de ajuste fino pueden ser optimizadas para que el sistema no solo genere respuestas precisas, sino que también las adapte al perfil y necesidades específicas del usuario.
- Reducción del costo computacional: El desarrollo de técnicas de optimización de recursos, como el uso de modelos de lenguaje de nicho o entrenados de forma eficiente puede ser más eficiente computacionalmente manteniendo la calidad de respuesta.
- Mejora de los métodos de desambiguación y manejo de polisemia: La ambigüedad semántica es un desafío importante en el procesamiento de lenguaje natural. La investigación de nuevas técnicas de embeddings y

estructuras semánticas que ayuden a los sistemas RAG a desambiguar términos comunes.

- Avances en la ética y la transparencia en la IA: A medida que las técnicas RAG se vuelven más comunes, es fundamental explorar formas de asegurar que estos sistemas funcionen de manera ética y transparente.
- Desarrollo de técnicas de evaluación automática: En la medida en que los LLMs se ajustan para tareas específicas cobra más relevancia contar con métodos para evaluar la precisión, relevancia y actualización de las respuestas generadas.

El presente trabajo se centra en la aplicación de técnicas RAG sobre LLMs OpenSource y en su aplicabilidad sobre información de ayudas públicas del Gobierno de España.

3. Materiales y métodos

3.1. Descripción general y arquitectura

El presente trabajo implementa la técnica RAG (Retrieval-Augmented Generation) para optimizar la búsqueda y consulta de información en la web del Sistema Nacional de Ayudas y Subvenciones Públicas de España, con el objetivo de centralizar y facilitar el acceso a las convocatorias disponibles para ciudadanos y entidades. La propuesta responde a dos necesidades principales:

- Proveer acceso rápido y preciso a información relevante dentro de un vasto conjunto de datos.
- Generar explicaciones comprensibles y contextualizadas que ayuden a interpretar la información obtenida.

Para lograrlo, se integran tres componentes clave:

- Obtención de datos: Extrae datos estructurados y actualizados de las páginas web relevantes mediante un API.
- Base de Datos Vectorial: Utiliza herramientas como FAISS o ChromaDB para almacenar representaciones semánticas de los documentos y realizar búsquedas eficientes basadas en similitud.
- Modelos de Lenguaje (LLMs): Generan respuestas claras y detalladas a partir de los datos recuperados, empleando modelos como Ollama (LLaMA) o OpenAI GPT que se ha utilizado para el despliegue en Hugging Face Spaces.

Este enfoque combina la recuperación de datos relevantes con la generación de texto natural en dos etapas principales:

- Recuperación de Datos Relevantes:
 - Los datos ya estructurados se procesan para crear embeddings mediante modelos de lenguaje.

- Durante una consulta, se generan embeddings del texto ingresado y se comparan con los existentes en la base vectorial, recuperando los documentos más relevantes.
- Generación de Texto:
 - Los documentos recuperados alimentan a un LLM, que interpreta la información y genera respuestas precisas o explicaciones basadas en los datos.

Para garantizar la calidad del sistema, se utiliza el framework RAGAS (Retrieval-Augmented Generation Analytics and Scoring) que evalúa:

- La relevancia y utilidad de los documentos recuperados.
- La coherencia, exactitud y claridad de las respuestas generadas por el LLM.

Además de esto, dentro de la parte de evaluación también se han desarrollado 2 scripts que nos permiten la generación de preguntas sintéticas en base a plantillas y autogeneradas en base al contenido generado con RAGAS mediante un LLM. Esto nos permite evaluar el modelo con una diversidad de preguntas autogeneradas y no solo en base a las interacciones del usuario con la aplicación.

El código fuente está accesible en la plataforma GitHub accediendo mediante esta URL https://github.com/desimartiout/TFM_DesiMarti y la versión web está en la URL <https://huggingface.co/spaces/DesiMarti/TFMCienciaDatos>.

La Figura 11 presenta un diagrama de la arquitectura de la solución final, mostrando las distintas alternativas de parametrización para optimizar el sistema. En capítulos posteriores se explicarán con detalle cada uno de estos componentes de la arquitectura. En esta sección solo se quiere dar una pincelada global para tener el contexto base con el objetivo de que se entiendan mejor los siguientes capítulos.

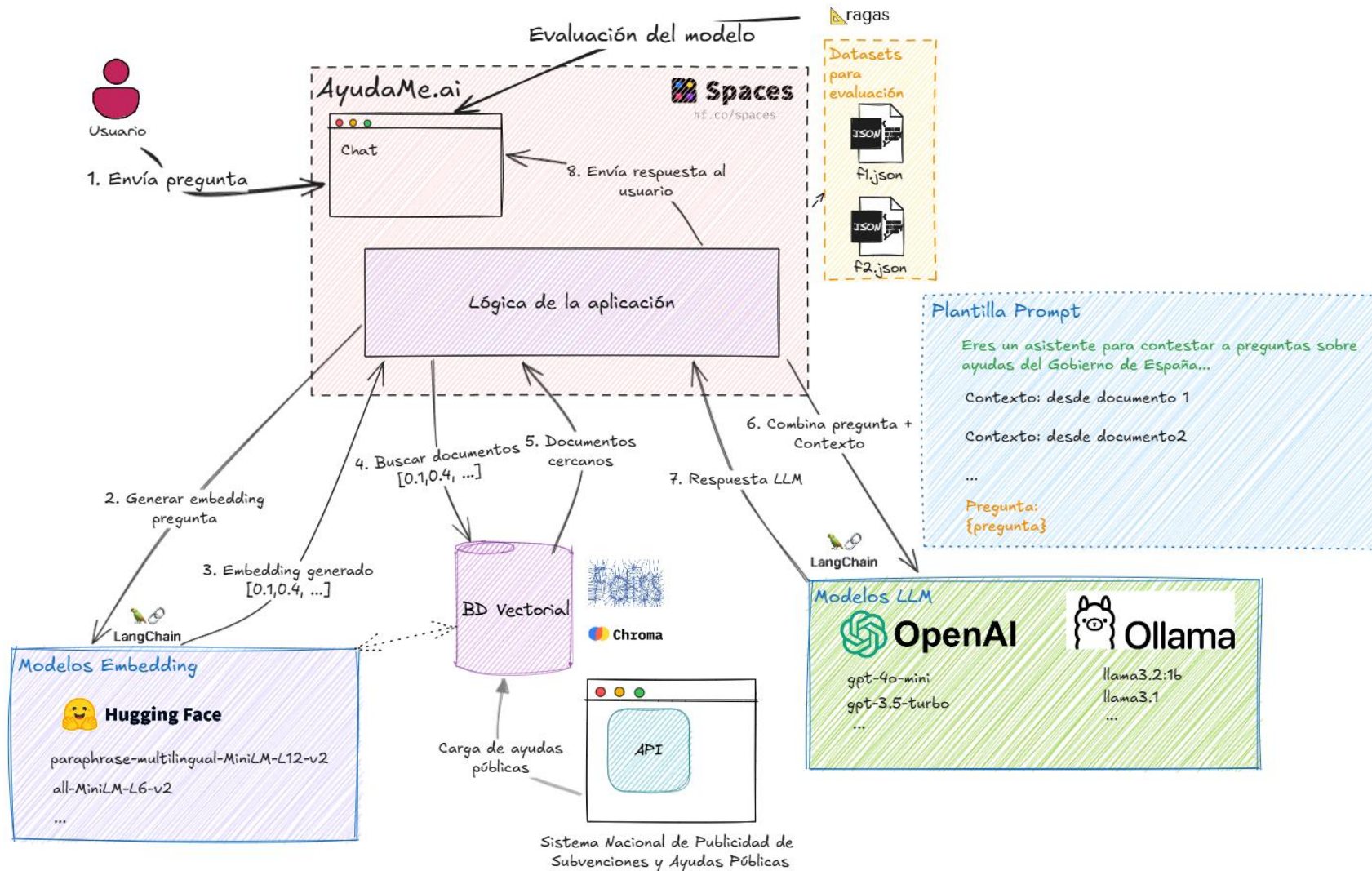


Figura 11: Diagrama de arquitectura a alto nivel

3.2. Fuentes de datos y Web Scrapping

La información utilizada en este trabajo proviene de la web del Sistema Nacional de Publicidad de Subvenciones y Ayudas Públicas, una plataforma que centraliza datos sobre convocatorias de subvenciones ofrecidas por diversas entidades y administraciones públicas del Reino de España.

Página principal del Sistema Nacional de Publicidad de Subvenciones y Ayudas Públicas. <https://www.pap.hacienda.gob.es/bdnstrans/GE/es/inicio>



Código BDNS	Administración	Departamento	Órgano	Fecha de Registro	Título	Título cooficial
801030	HERMIGUA	AYUNTAMIENTO DE HERMIGUA		04/12/2024	SUBVENCIONES DESTINADAS A LA ADQUISICIÓN DE MATERIAL ESCOLAR Y DEPORTIVO PARA EL CURSO 2024-2025	
801029	DIPUTACIÓN PROV. DE BADAJOZ	DIPUTACIÓN PROVINCIAL DE BADAJOZ		04/12/2024	BASES REGULADORAS DE SUBVENCIONES DE LA DIPUTACIÓN DE BADAJOZ DESTINADAS A ASOCIACIONES Y ENTIDADES SIN ÁNIMO DE LUCRO CON SEDE O DELEGACIÓN PERMANENTE EN LA PROVINCIA DE BADAJOZ QUE DESARROLLEN PROYECTOS DE ACCIÓN SOCIAL EN EL EJERCICIO ECONÓMICO 2025	

Figura 12: Sistema Nacional de Publicidad de Subvenciones y Ayudas Públicas

En su página principal aparece un listado con el detalle de todas las convocatorias ordenado por fecha más reciente y permite ir paginando todos los resultados y accediendo al detalle de las estas. Un ejemplo de acceso al detalle de una convocatoria concreta, en este caso 801030, de la ayuda en la web lo podemos ver en la Figura 13 y que podemos acceder al mismo mediante esta URL <https://www.pap.hacienda.gob.es/bdnstrans/GE/es/convocatorias/801030>



Sistema Nacional de Publicidad de Subvenciones y Ayudas Públicas

Bienvenido | Benvingut | Benvido | Ongi etorri | Welcome

Inicio Convocatorias Alertas Concesiones Planes Estratégicos Infracciones y sanciones

Información de la convocatoria

Órgano convocante: HERMIGUA AYUNTAMIENTO DE HERMIGUA

Sede electrónica para la presentación de solicitudes: www.hermigua.sedelectronica.es

Código BDNS: 801030 Mecanismo de Recuperación y Resiliencia: NO Fecha de registro: 04/12/2024 Tipo de convocatoria: Concurrencia competitiva - canónica

Presupuesto total de la convocatoria: 1.000,00 € Instrumento de ayuda: SUBVENCIÓN Y ENTREGA DINERARIA SIN CONTRAPRESTACIÓN

Título de la convocatoria en español: SUBVENCIONES DESTINADAS A LA ADQUISICIÓN DE MATERIAL ESCOLAR Y DEPORTIVO PARA EL CURSO 2024-2025

Título de la convocatoria en otra lengua cooficial: www.hermigua.sedelectronica.es

Tipo de beneficiario elegible: PERSONAS FÍSICAS QUE NO DESARROLLAN ACTIVIDAD ECONÓMICA Sector económico del beneficiario: P - EDUCACIÓN Región de impacto: ES706 - La Gomera Finalidad (política de gasto): EDUCACIÓN

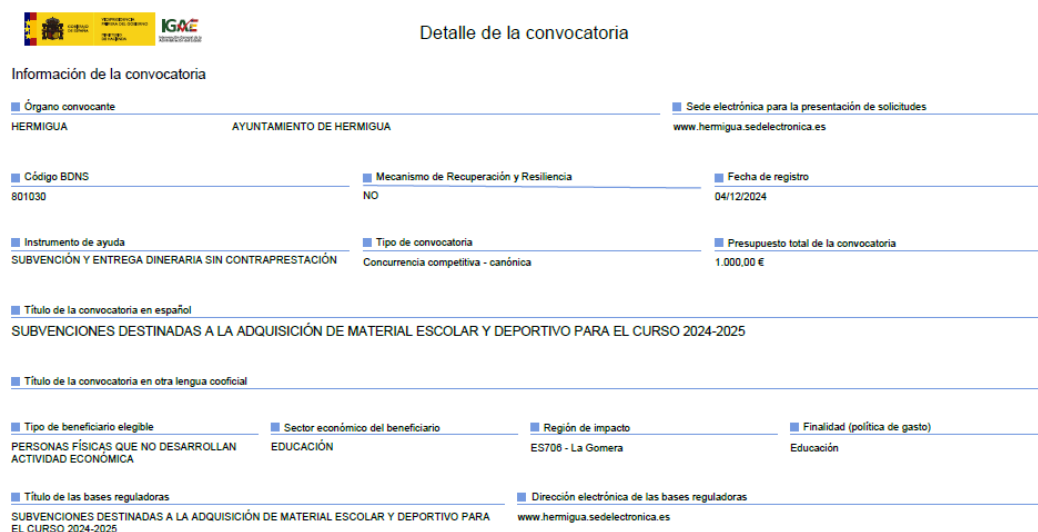
Título de las Bases reguladoras: SUBVENCIONES DESTINADAS A LA ADQUISICIÓN DE MATERIAL ESCOLAR Y DEPORTIVO PARA EL CURSO 2024-2025 Dirección electrónica de las bases reguladoras: www.hermigua.sedelectronica.es

Ver concesiones de esta convocatoria Ver concesiones de Ayudas de Estado de esta convocatoria Ver concesiones de minimis de esta convocatoria Ficha PDF de la convocatoria

Figura 13: Ficha detalle de ayuda pública

Dentro del detalle de la convocatoria también se permite visualizar el detalle en formato PDF, como se puede ver en la Figura 14. Es descargable mediante esta URL (donde el id es el identificador interno de la convocatoria)

<https://www.pap.hacienda.gob.es/bdnstrans/api/convocatorias/pdf?id=1002590&vpd=GE>



Detalle de la convocatoria

Información de la convocatoria

Órgano convocante: HERMIGUA AYUNTAMIENTO DE HERMIGUA

Sede electrónica para la presentación de solicitudes: www.hermigua.sedelectronica.es

Código BDNS: 801030 Mecanismo de Recuperación y Resiliencia: NO Fecha de registro: 04/12/2024

Instrumento de ayuda: SUBVENCIÓN Y ENTREGA DINERARIA SIN CONTRAPRESTACIÓN Tipo de convocatoria: Concurrencia competitiva - canónica Presupuesto total de la convocatoria: 1.000,00 €

Título de la convocatoria en español: SUBVENCIONES DESTINADAS A LA ADQUISICIÓN DE MATERIAL ESCOLAR Y DEPORTIVO PARA EL CURSO 2024-2025

Título de la convocatoria en otra lengua cooficial:

Tipo de beneficiario elegible: PERSONAS FÍSICAS QUE NO DESARROLLAN ACTIVIDAD ECONÓMICA Sector económico del beneficiario: EDUCACIÓN Región de impacto: ES706 - La Gomera Finalidad (política de gasto): Educación

Título de las bases reguladoras: SUBVENCIONES DESTINADAS A LA ADQUISICIÓN DE MATERIAL ESCOLAR Y DEPORTIVO PARA EL CURSO 2024-2025 Dirección electrónica de las bases reguladoras: www.hermigua.sedelectronica.es

Figura 14: Ficha en formato PDF

Estudiando la forma de obtener la información de esta se utilizó un profiler de red para ver las llamadas http y se visualizó que la información se obtenía mediante una API de un servicio REST mediante formato JSON. El uso del API REST ofrece varias ventajas en comparación con el scraping directo de las páginas web:

- Estandarización: Los datos proporcionados por el API están estructurados y normalizados, lo que reduce la necesidad de realizar preprocesamientos complejos.

- Actualización dinámica: Los datos obtenidos reflejan cambios o actualizaciones realizadas en tiempo real por la plataforma.
- Eficiencia: La comunicación RESTful optimiza la transferencia de datos, lo que facilita consultas rápidas y escalables.

Tras evaluar las diferentes opciones posibles nos decantamos por la utilización de este API REST ya que nos proporciona una comunicación eficiente basada en un formato JSON para la consulta y recuperación de datos estructurados.

3.2.1. Obtención de los datos mediante el API REST

El uso del API nos asegura que los datos estén actualizados y representen fielmente las convocatorias vigentes en el momento de la consulta. Una vez decidida la utilización del API REST lo siguiente era identificar las diferentes llamadas a realizar. Necesitaríamos una forma de acceder para obtener:

- Detalle de todo el listado de convocatorias.
- Identificación detallada de cada convocatoria de forma unívoca.

Durante las llamadas al servicio vimos aparece un aviso legal y observamos que se permite la reutilización de la información. Esta utilización es tanto para fines comerciales como no comerciales bajo unas condiciones generales en la reutilización de los datos. Tras revisar detenidamente esta información tendremos en cuenta todos estos detalles en el proyecto que vamos a realizar. En el Anexo 8.5 se puede ver más detalle tanto del aviso legal como de las condiciones generales.

Por otra parte, centrándonos en concreto en las llamadas a realizar al API, hemos utilizado un profiler de llamadas HTTP para identificar 2 tipos de URLs para obtener la información que necesitamos.

- Url de acceso al servicio REST de búsqueda y listado de información de las ayudas.

<https://www.pap.hacienda.gob.es/bdnstrans/api/convocatorias/ultimas?vpd=GE&page=0&pageSize=2&order=fechaRecepcion&direccion=desc>

- Url de acceso al detalle de una ayuda a través del API

<https://www.pap.hacienda.gob.es/bdnstrans/api/convocatorias?numConv=801030&vpd=GE>

3.2.2. Estructura y características de los datos

Los campos principales extraídos incluyen:

- Metadatos generales: Identificador único, fecha de publicación, estado de la convocatoria.

- Información descriptiva: Descripción textual de los objetivos, beneficiarios, y bases reguladoras.
- Aspectos económicos: Presupuesto total disponible y posibles mecanismos de financiación.
- Datos temporales: Fechas de inicio y fin del período de solicitud.

3.2.3. Consideraciones técnicas

Se implementaron herramientas y bibliotecas en Python, como requests y json, para interactuar con el API, enviar peticiones GET y procesar las respuestas JSON. Adicionalmente, se integraron mecanismos de autenticación y manejo de errores para garantizar la estabilidad del sistema en escenarios de alta carga o fallos temporales en el API.

3.3. Preprocesamiento de los datos y generación de Embeddings

El preprocesamiento de los datos es una etapa crucial para garantizar que la información obtenida sea correcta. Entre las tareas importantes dentro de la fase de preprocesamiento de los datos se encuentran las siguientes:

- Limpieza y estructuración:
 - Conversión de datos extraídos a formato JSON u otros formatos procesables.
 - Eliminación de datos redundantes o inconsistentes.
- Enriquecimiento de datos:
 - Agregación de metadatos relevantes para mejorar las búsquedas.
 - Traducción o normalización de términos si es necesario.

3.3.1. Limpieza y estructuración de los datos

Los datos obtenidos en formato JSON fueron revisados y limpiados para garantizar su consistencia y relevancia. Este proceso incluyó:

- Eliminación de datos redundantes: Se descartaron campos no relevantes para el análisis, como identificadores internos no utilizados.
- Normalización de valores: Se unificaron los formatos de fechas, números y textos descriptivos para evitar ambigüedades.

En nuestro caso al utilizar un API con información estructurada este proceso se facilitó considerablemente. Si es cierto que en una primera aproximación de la solución se optó por utilizar directamente el JSON devuelto como información almacenada en la base de datos vectorial y no solo como metadatos asociados. El resultado no fue el esperado por los motivos que se demuestra a continuación:

- Había muchos datos que no aportaban información relevante para las búsquedas de los usuarios.
- Las etiquetas de los elementos JSON no tenían los textos correctos como para que el proceso de búsqueda y procesamiento del LLM fuera correcto.

Esto hizo que en sucesivas iteraciones del proceso de desarrollo se optara por otras opciones de almacenamiento de texto como, por ejemplo:

- Procesar el detalle de la ayuda en PDF y pasar un OCR para el reconocimiento de texto.
- Conversión del JSON a un formato más legible y entendible por el LLM.

El primer caso, si bien funcionaba mejor que el procesamiento del JSON directamente, no tenía mucho sentido ya que perdíamos la estructuración de los datos que da un carácter semántico. Además, que el OCR no siempre funcionaba correctamente. Fue entonces cuando revisamos la posibilidad de convertir el texto en formato JSON a otro formato distinto que fuera mejor para el procesamiento del LLM. Para ello buscamos cumplir las siguientes premisas:

- El texto debía ser entendible basado en texto.
- La información debería estar estructurada correctamente.
- No debería aumentar mucho el almacenamiento de la información mediante la utilización de etiquetas o información irrelevante.

Fue entonces cuando nos planteamos revisar un formato ligero como YAML. Este formato fue elegido por sus características de legibilidad y estructura jerárquica, lo que facilita que el modelo entienda la relación semántica entre los diferentes elementos del texto. Además, aprovechamos este proceso de conversión de JSON a YAML para:

- Eliminar información no importante del JSON.
- Renombrar etiquetas por otras más legibles y entendibles por el modelo.

Resumiendo, el preprocesamiento de la información se centró en realizar esta conversión del formato JSON a YAML con las adaptaciones comentadas. Por tanto, almacenamos los metadatos en formato JSON y utilizamos el texto en formato YAML para optimizar su comprensión por el modelo LLM. Esta estrategia facilitó:

- Una fácil integración con la base de datos vectorial para indexación.
- Una recuperación rápida y estructurada de la información relevante durante la generación de respuestas.
- La comprensión del modelo LLM y la mejora de la respuesta.

3.4. Almacenamiento y búsqueda de datos

3.4.1. Base de datos vectorial: ChromaDB y FAISS

El almacenamiento eficiente de representaciones vectoriales es un componente clave en sistemas de RAG (Retrieval-Augmented Generation), ya que permite realizar búsquedas rápidas y precisas sobre documentos procesados. Inicialmente en una primera aproximación, el sistema fue desarrollado utilizando FAISS (Facebook AI Similarity Search) como biblioteca de almacenamiento en memoria para la indexación de vectores.

FAISS es una biblioteca de código abierto diseñada para la indexación y búsqueda de vectores, optimizada para trabajar en memoria. Durante las primeras etapas del desarrollo, FAISS nos permitió:

- Realizar unas pruebas rápidas experimentales con vectores generados por el modelo.
- Realizar indexaciones en memoria que nos ofrecían tiempos de respuesta muy bajos para conjuntos de datos pequeños o medianos.

Tras esta primera aproximación con FAISS también realizamos una prueba con OpenSearch como una solución de búsqueda y análisis de código abierto que ofrece capacidades para manejar datos estructurados y no estructurados. La motivación inicial para explorar OpenSearch radicó en su robustez y versatilidad, especialmente en tareas de búsqueda textual y análisis avanzado. Se configuró un servicio de OpenSearch utilizando un contenedor Docker, donde se habilitaron los módulos necesarios para trabajar con datos vectoriales. OpenSearch requería un despliegue en un entorno Docker, lo que añadía una capa adicional de complejidad operativa al sistema. Además de esto, la configuración y mantenimiento del servicio demandaban recursos adicionales, especialmente en entornos de desarrollo locales o limitados.

Otra opción que revisamos fue ChromaDB como base de datos vectorial debido a sus capacidades de persistencia nativa y su diseño orientado a la gestión avanzada de datos embebidos. Al ser una herramienta de código abierto, ofrece independencia tecnológica y flexibilidad para futuras adaptaciones. También permite almacenar en ella toda la información: índice, datos y metadatos.

Otra alternativa planteada en la solución y que se puede parametrizar es FAISS, pero en este caso solo almacenamos el índice y los metadatos se almacenan en un fichero JSON separado.

Por tanto, la solución planteada permite parametrizar entre el uso de una base de datos ChromaDB o FAISS mediante un fichero de configuración de la aplicación.

3.5. Tecnologías utilizadas

El desarrollo del sistema RAG se fundamentó en la integración de tecnologías open-source para garantizar flexibilidad, reproducibilidad y escalabilidad. Los componentes principales de este sistema incluyen el modelo de lenguaje Llama 3.2: latest, ejecutado en local mediante Ollama, y la base de datos vectorial ChromaDB o FAISS para la recuperación semántica de información.

El despliegue de la aplicación nos trajo problemas para usar estos LLMs al tener altos requerimientos de hardware. Fue entonces cuando se planteó la opción de añadir la opción de OPENAI como alternativa que permite encontrar un servicio de hosting con menos requerimientos.

Por tanto, el sistema permite de una forma paramétrica en un fichero de configuración cambiar tanto el modelo generador de embeddings, la base de datos vectorial (ChromaDB o FAISS), el LLM (OLLAMA, OPENAI) y la elección de uno de los modelos que corre en cada una de estas plataformas. Además de esto la evaluación mediante RAGAS también se puede hacer tanto con Ollama como con OpenAI.

3.5.1. Embeddings: paraphrase-multilingual-mpnet-base-v2

Para la codificación de la información en embeddings se probaron diferentes modelos de HuggingFace (*paraphrase-multilingual-MiniLM-L12-v2*, *all-mpnet-base-v2*, *all-MiniLM-L6-v2*). Finalmente se optó por *paraphrase-multilingual-mpnet-base-v2* al estar optimizado para tareas de búsqueda semántica en múltiples idiomas al estar pre entrenado específicamente para capturar relaciones semánticas entre frases y textos, lo que mejora la capacidad de identificar información relevante en el contexto de consultas realizadas por el usuario.

3.5.2. Base de Datos Vectorial: ChromaDB y FAISS

Se ha implementado la posibilidad de usar ChromaDB como la base de datos vectorial, principalmente por ser Open Source y por su eficiencia y escalabilidad ofreciendo un tiempo de respuesta rápido en la búsqueda de similitudes entre vectores, lo que resulta esencial para aplicaciones de preguntas y respuestas. Además, también permite usar FAISS como base de datos vectorial alternativa.

3.5.3. LLMs: Ollama y OPENAI

Ollama: Llama 3.2: latest

Para generar respuestas basadas en las consultas del usuario, se han probado múltiples modelos soportados por Ollama, finalmente se utilizó el modelo de lenguaje *Llama 3.2: latest* y a partir de este se creó una copia en uno nuevo llamado *LlamaAyudas: latest*.

Este modelo fue seleccionado por permitir su despliegue con Ollama en local, por su compatibilidad con LangChain y por ser Open source, alineándose con los principios del proyecto de utilizar soluciones accesibles y sin dependencias de proveedores comerciales.

OPENAI: gpt-4o-mini

También se ha implementado la opción de utilizar OPENAI como LLM ya que fue necesario para que la aplicación fuera desplegable en Hugging Face Spaces.

En resumen, como se ha comentado anteriormente la elección del LLM y del modelo de este se puede parametrizar en la aplicación en el fichero *global_config.py* del directorio */config/*.

3.5.4. Despliegue en HuggingFace Spaces

Para posibilitar la prueba del proyecto realizado se ha desplegado la solución en HuggingFace Spaces cambiando el LLM para que se base en OPENAI con el modelo GPT-3.5 Turbo o cualquier otro. Esto nos permite una implementación accesible, potente y fácil de usar, que destaca por su capacidad para manejar consultas complejas en el contexto de recuperación de información. Esto es así por las limitaciones para publicar la solución en una arquitectura que tuviera el hardware necesario ya que el coste se elevaba considerablemente.

Con esta solución de compromiso de cambiar el LLM se permite probar la solución sin necesidad de tener que desplegar la solución en local facilitando así la evaluación de este TFM.

La URL de acceso es <https://huggingface.co/spaces/DesiMarti/TFMCienciaDatos> y en la Figura 15 tenemos una imagen de la página principal de esta web publicada.



Figura 15: Imagen web accesible en HuggingFace Spaces

3.6. Interfaz de usuario

El diseño de la interfaz del chatbot se centró en la simplicidad y la accesibilidad para garantizar que usuarios con diversos niveles de experiencia técnica pudieran interactuar fácilmente con el sistema. Se buscó facilitar la interacción en tiempo real, permitiendo consultas y respuestas instantáneas, y presentar la información de manera estructurada, destacando tanto las respuestas generadas por el modelo como las fuentes recuperadas, lo que mejora la comprensión y transparencia del sistema.

Para desarrollar la interfaz, se eligió Streamlit debido a sus características que favorecen un desarrollo rápido y un despliegue sencillo. Esta herramienta permite un renderizado dinámico, actualizando la interfaz según las interacciones del usuario, y ofrece una integración fluida con Python, lo que facilita la conexión con componentes clave del pipeline RAG, como el modelo Llama y la base de datos vectorial ChromaDB. Además, su capacidad para ejecutarse localmente o en la nube con requisitos mínimos amplía su versatilidad.

El diseño incluyó elementos esenciales como un campo de entrada para que los usuarios introduzcan consultas en lenguaje natural, un historial de conversación que permite revisar interacciones previas y una sección dedicada a visualizar las fuentes recuperadas, mostrando enlaces a sus ubicaciones originales. Este enfoque estructurado garantiza una experiencia de usuario clara y eficiente.

El flujo de interacción se diseñó de manera intuitiva: los usuarios ingresan una consulta en el campo de texto, la cual es procesada por el pipeline RAG para recuperar información relevante y generar una respuesta mediante Llama 3.2:1b. Los resultados se presentan junto con los fragmentos de las fuentes recuperadas,

y el historial de consultas se actualiza dinámicamente para facilitar el seguimiento de la conversación.

Entre las ventajas de usar Streamlit destaca su capacidad para realizar iteraciones rápidas en la interfaz, incorporando feedback de manera ágil durante el desarrollo. También es compatible con herramientas nativas de Python, lo que simplifica la integración con los componentes técnicos del sistema, y su accesibilidad mediante un navegador web elimina la necesidad de instalaciones adicionales, mejorando la experiencia del usuario final.

El objetivo principal es ofrecer una experiencia sencilla y funcional, organizada en dos secciones:

- **Página de Chat:** Actúa como asistente virtual, permitiendo a los usuarios interactuar con el modelo de lenguaje (LLM) y obtener información basada en los documentos cargados.
- **Página de Administración:** Facilita la subida y gestión de documentos, que posteriormente son procesados para su indexación en la base de datos vectorial utilizada en el sistema RAG.

Página de Chat

La página de chat es el componente principal de interacción para los usuarios finales. Está diseñada con un enfoque minimalista para maximizar la usabilidad y permitir una experiencia fluida al consultar la información. En la Figura 17 se puede ver una imagen de prototipo a bajo nivel y cuyas características principales son:

- **Entrada de Texto:** Campo para que el usuario introduzca preguntas en lenguaje natural.
- **Salida del Modelo:** Respuestas generadas por el modelo LLM, enriquecidas con contexto recuperado desde la base de datos vectorial (ChromaDB).
- **Historial de Conversación:** Registro dinámico de la interacción para facilitar la consulta de respuestas previas.
- **Indicadores de Estado:** Información sobre el estado del sistema, como la conexión con el modelo o la base de datos.

Página de subida de documentos

Se ha desarrollado una página de subida de documentos para que los usuarios administradores puedan alimentar la base de datos vectorial sin necesidad de depender del módulo de webscraping. Inicialmente permitía la subida manual de documentos en formato YAM. Esta primera aproximación fue de gran utilidad al inicio del proyecto ya que nos permitió hacer pruebas del almacenamiento de documentos en la base de datos vectorial en diferentes formatos de una forma

ágil y eficiente. Permitiendo añadir, listar y eliminar documentos de la base de datos vectorial.

Finalmente se ha evolucionado esta página para que permita adjuntar ayudas en formato pdf quedando como se muestra en la Figura 16.



Figura 16: Imagen de página de carga de ayudas en pdf

Página de listado de documentos

Se trata de una página que permite ver los documentos cargados en la base de datos vectorial y ver el contenido de cada uno de ellos. Para el despliegue en HuggingFace Spaces solo se han cargado 100 documentos por limitaciones de espacio. En la Figura 17 se puede ver la visualización de la página:



Figura 17: Imagen de página de listado de documentos

Página Ragas - Datasets

En esta página se pueden visualizar los datasets generados bien con el interfaz gráfico de la aplicación o bien mediante el proceso batch desatendido que se explicará en la siguiente sección del presente TFM. Como se muestra en la Figura

18, mediante esta página se puede visualizar el contenido de dicho dataset tanto en formato tabla como en formato JSON.



Figura 18: Imagen de página Ragas - Datasets

Página Ragas - Resultados

Mediante esta página se puede visualizar el resultado de las evaluaciones de los diferentes datasets. En la Figura 19 podemos ver la imagen con información del dataset de resultados tanto en formato csv como en un pequeño análisis descriptivo donde podemos seleccionar cada una de las métricas y ver más detalle de esta.



Figura 19: Imagen de página Ragas - Resultados

Página Aviso legal

En esta página, como muestra la Figura 20, podemos ver el aviso legal al que nos obliga la utilización del API de la web de ayudas del Gobierno de España, además de poner un descargo de responsabilidad de cara a quitarnos posibles problemas legales de la utilización de la información proporcionada por el chatbot.



Figura 20: Imagen de página Aviso legal

4. Evaluación y resultados

4.1. Evaluación con RAGAS

4.1.1. Planteamiento general y objetivos

Este proyecto se orienta a abordar una evaluación integral del sistema utilizando la metodología RAGAS. Este enfoque metodológico permite no solo analizar el rendimiento del sistema desde una perspectiva cuantitativa, sino también garantizar la alineación con objetivos específicos de interpretabilidad y efectividad. El planteamiento general en la evaluación con RAGAS se estructurará en tres fases clave:

- Diseño de Escenarios de Evaluación: Creación de casos de prueba representativos, así como definición de métricas de evaluación alineadas con los objetivos específicos.
- Implementación del Proceso de Evaluación: Utilización de RAGAS para medir automáticamente los resultados y comparar las diferentes estrategias de recuperación y generación para identificar configuraciones óptimas.
- Análisis y Documentación de Resultados: Interpretación de los resultados obtenidos, destacando fortalezas y debilidades del sistema para establecer propuestas de mejora basadas en los hallazgos.

En cuanto a los objetivos específicos que queremos obtener mediante la evaluación con RAGAS se encuentran los siguientes:

- Medir la precisión en la recuperación de la información: Evaluar la capacidad del sistema RAG para recuperar documentos relevantes y pertinentes a partir de la base de datos vectorial.
- Evaluar las respuestas generadas: Analizar la calidad, coherencia y relevancia de las respuestas generadas por el LLM.
- Tener coherencia con el contexto proporcionado: Verificar que las respuestas proporcionadas por el sistema se basen exclusivamente en información documentada en la base de datos, minimizando al máximo la alucinación.
- Optimizar y mejorar el desempeño: Identificar áreas de mejora en la arquitectura de RAG mediante un análisis de errores y rendimiento, optimizando las consultas y las respuestas generadas.

Por último, hay que comentar que en el anexo 8.4 se describe pormenorizadamente este método de evaluación.

4.1.2. Métricas utilizadas

Hemos utilizado RAGAS para evaluar la calidad y efectividad de un sistema RAG. Este framework se basa en unas métricas para evaluar diferentes aspectos clave de la generación de texto y la relación entre las respuestas generadas y el contexto proporcionado. Las métricas utilizadas para la evaluación de este modelo son las siguientes:

Métrica	Qué mide	Importancia
Context Recall	La proporción de contenido relevante del contexto que el LLM efectivamente incluye en su respuesta.	Evalúa si el modelo aprovecha la información del contexto para formular sus respuestas.
Factual Correctness	La precisión factual de la respuesta generada en comparación con el contexto recuperado	Valida si la información generada por el modelo coincide con los hechos presentes en las fuentes contextuales, ayudando a reducir errores de interpretación
Faithfulness	La fidelidad de la respuesta respecto al contexto.	Asegura que la respuesta se basa estrictamente en el contexto y no se invente datos.
Semantic Similarity	La similitud semántica entre la respuesta generada y el contexto, utilizando embeddings.	Evalúa si las ideas clave de la respuesta y el contexto están alineadas, incluso cuando no coincidan palabra por palabra
Answer Relevancy	La relevancia de la respuesta con respecto a la pregunta planteada, utilizando	Garantiza que las respuestas se enfoquen en resolver la consulta específica y no

	embeddings para calcular la proximidad semántica	divaguen hacia información irrelevante
Context Precision	La precisión con la que el LLM utiliza únicamente las partes relevantes del contexto.	Asegura un enfoque conciso y directo al responder, maximizando la claridad y relevancia

Tabla 3: Métricas RAGAS evaluadas

4.1.3. Generación de datasets de preguntas

4.1.3.1. Introducción

Para la evaluación del sistema mediante RAGAS se han diseñado tres estrategias distintas para la generación de datasets de evaluación:

- Generar datasets con datos reales: Recopilar interacciones reales de la aplicación para evaluarlas posteriormente.
- Generar datasets con datos sintéticos: Generar preguntas simuladas con diversos parámetros, almacenándolas junto con los datos reales.
- Generar datasets con datos generados por RAGAS: Usar el framework para crear datasets basados en la información de la base de datos vectorial, simulando casos de uso reales.

La primera de las estrategias está clara y se basa en almacenar la información utilizada en la aplicación por parte de usuarios reales. En las 2 siguientes secciones se explican las 2 últimas estrategias, así como los comandos para su generación. Por otra parte, en el Anexo 8.5 se describen con más detalle ejemplos concretos generados con cada una de estas 3 estrategias.

4.1.3.2. Datasets de preguntas sintéticos

La generación del dataset sintético de preguntas se ha realizado mediante un script con diferentes tipos de preguntas parametrizado de tal forma que hemos conseguido generar 87 preguntas distintas que nos sirvan para preguntar el sistema como si fuera un usuario y que nos den resultados a evaluar posteriormente. El script se puede ejecutar con el siguiente comando desde la raíz del proyecto:

```
python.exe generar_dataset_sintetico.py
```

Este dataset es más similar a como interactuaría un usuario ya que a priori el usuario no conoce la información almacenada en la bd vectorial. Esto tiene sus ventajas y sus inconvenientes ya que podemos estar preguntando por información que no tenemos y nos puede dar la sensación de que la información que nos devuelve no es correcta cuando si lo es.

4.1.3.3. Datasets de preguntas con RAGAS

Otra opción más recomendada es utilizar el propio framework de RAGAS para que a partir de los datos almacenados en la base de datos vectorial genere automáticamente los datasets de preguntas que sirvan de entrada para automatizar la consulta de datos como si lo hiciera un usuario y genere los datasets para la evaluación.

Para este caso se ha usado un script que se encarga de generar 100 preguntas distintas en un fichero csv a través de RAGAS y el LLM seleccionado. El script a ejecutar ha sido el siguiente

```
python.exe generar_dataset_ragas.py
```

A diferencia del punto anterior esta generación es más inteligente ya que se basa en la información almacenada y además de la pregunta incorpora información de la referencia (reference) la respuesta (response) y los contextos obtenidos (retrieved contexts) en la que se basa para poder ser evaluada a futuro. Un ejemplo de este fichero puede verse en la siguiente imagen.

A partir de estos ficheros podemos automatizar tanto la búsqueda de estas preguntas para obtener los datasets con los datos obtenidos en la consulta para poder evaluar el comportamiento de la aplicación.

4.1.4. Generación de Datasets para la evaluación

4.1.4.1. Generación de Datasets mediante IU

La aplicación está preparada para generar datasets de evaluación a partir de las conversaciones que se realizan en el chat. Esta información se almacena diariamente en un fichero dentro del directorio */ragas_eval/datasets* (por ejemplo, *2024_12_15_ragas.json*) en formato JSON. Un ejemplo se puede ver en la Figura 20 donde se pueden identificar los siguientes campos:

- **User_input:** Texto introducido por el usuario
- **Reference:** Referencia que usa para validar la respuesta. En este caso está vacío ya que será manualmente el usuario que evalúa el que decida el texto desde el cual se obtiene la información correcta.
- **Response:** Respuesta que realiza el LLM

- **Retrieved_contexts:** Contexto proporcionado por la llamada a la base de datos vectorial e incorporado en la llamada al LLM.

4.1.4.2. Generación de Datasets mediante búsquedas desatendidas

La aplicación contempla la posibilidad de simular la interacción del usuario con el chatbot mediante una batería de preguntas almacenada en un fichero de texto en el directorio `/ragas_eval/questions`.

Datasets preguntas sintéticos

Para simular las búsquedas de cada una de las líneas del fichero sintético `dataset_sintetico.txt` se debería lanzar este comando desde la raíz del proyecto

```
python.exe buscar_batch_sintetico.py dataset_sintetico.txt
```

Este fichero tiene 87 preguntas y el resultado de estas se almacenará en el mismo dataset de evaluación del día en que se ejecuta (como si hubieran sido ejecutadas por un usuario por la interfaz web). En la Figura XX se puede ver una sección del dataset de evaluación generado tras la ejecución del proceso anterior.

Datasets de preguntas con RAGAS

Otra opción que tenemos es ejecutar las búsquedas desatendidas a partir de un fichero csv generado a partir de RAGAS mediante el comando:

```
python.exe buscar_batch_ragas.py  
dataset_ragas_2024_12_21_15_56_21_ragas_results.csv
```

Este fichero tiene 100 preguntas y el resultado de estas se almacenará en el mismo dataset de evaluación del día en que se ejecuta (como si hubieran sido ejecutadas por un usuario por la interfaz web). En la Figura XX se puede ver una sección del dataset de evaluación generado tras la ejecución del batch anterior

4.1.5. Evaluación de la solución

Una vez tenemos los datasets de evaluación en formato JSON, ya sean por interacción de un usuario real o mediante ejecución desatendida, los podemos evaluar mediante el siguiente comando:

```
python.exe evaluar.py 2024_12_14_ragas.json
```

El fichero con los resultados se almacena en el directorio `ragas_eval/results` en formato CSV. Un fichero de ejemplo se puede ver en la Figura 21 donde se identifican los campos que se utilizan para poder evaluar (`user_input`,

retrieved_contexts, response, reference) junto con los campos resultado de las métricas evaluadas (context_recall, factual_correctness, faithfulness, semantic_similarity, answer_relevancy, context_precision)

4.1.6. Resultados Cuantitativos – Dataset sintético

Los resultados obtenidos en formato csv se pueden cargar en pandas y hacer un pequeño resumen para ver cómo se han comportado cada una de las métricas para un fichero csv mediante el siguiente script

```
python.exe mostrar_eval.py ..._ragas_results.csv
```

En nuestro caso hemos evaluado un dataset con 87 registros

	faithfulness	semantic_similarity	answer_relevancy	context_precision
count	100,00	100,00	100,00	100,00
std	0,29	0,03	0,45	0,50
min	0,00	0,82	0,00	0,00
25%	0,50	0,88	0,00	0,00
50%	0,79	0,90	0,84	0,00
75%	0,93	0,93	0,91	1,00
max	1,00	0,97	0,98	1,00

Tabla 4: Estadísticas descriptivas del dataset sintético.

En la Figura 21 se puede ver la visualización en gráfico de histograma.

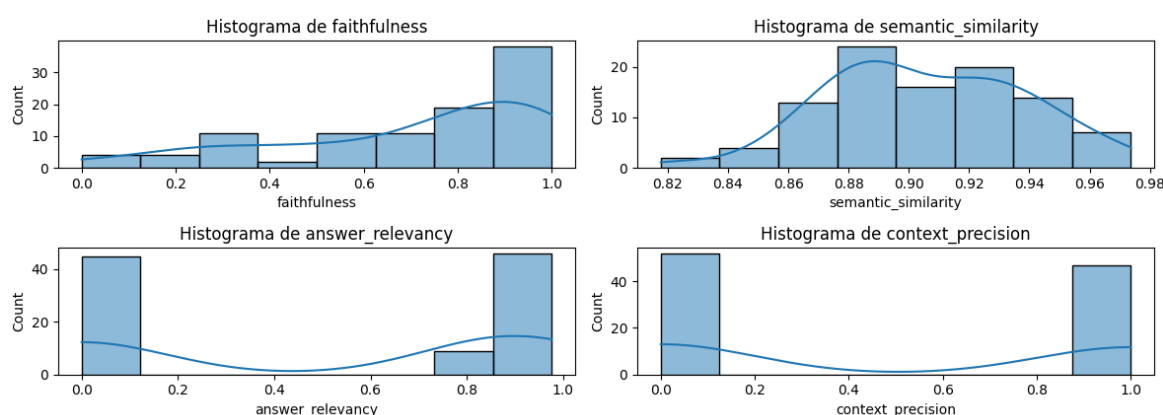


Figura 21: Histograma de las métricas dataset sintético.

Métricas Analizadas

- Faithfulness mide si las respuestas son fieles al contexto. Con una media de 0.6827, las respuestas son fieles en promedio un 68.3%. La desviación

estándar de 0.2856 muestra una variabilidad moderada, mientras que el rango de 0 a 1 refleja respuestas más y menos fieles al contexto.

- Semantic Similarity evalúa la similitud semántica con el texto de referencia. La media de 0.7160 indica un 71.6% de similitud promedio, con una desviación estándar muy baja (0.0064) que muestra gran consistencia. El rango, de 0.7000 a 0.7336, confirma estas pequeñas variaciones.
- Answer Relevancy mide la relevancia de las respuestas respecto a las preguntas. La media de 0.6032 indica un 60.3% de relevancia promedio, aunque la alta desviación estándar (0.4526) refleja respuestas muy variables. El rango oscila entre 0 (irrelevantes) y 0.9805 (altamente relevantes).
- Context Precision mide la relevancia del contexto recuperado. La media de 0.5172 sugiere una precisión moderada del 51.7%. Sin embargo, la desviación estándar de 0.5026 y el rango de 0 a 1 muestran casos de precisión perfecta y otros completamente imprecisos.

Análisis Cuantitativo

- Consistencia: La métrica Semantic Similarity es consistentemente alta, con una desviación estándar muy baja. Esto sugiere que el modelo genera respuestas que son semánticamente similares al texto de referencia de manera uniforme.
- Variabilidad alta: Métricas como Answer Relevancy y Context Precision presentan desviaciones estándar elevadas, lo que indica comportamientos inconsistentes del sistema en términos de relevancia y precisión contextual.
- Rendimiento razonable: Faithfulness tiene una media moderadamente alta (68.3%), pero hay margen para mejorar la fidelidad al contexto en algunos casos.

En el Anexo 8.7 se pueden ver ejemplos de resultados satisfactorios y no satisfactorios con preguntas sintéticas.

4.1.7. Resultados Cuantitativos – Dataset generado por RAGAS

Los resultados obtenidos en formato csv se pueden cargar en pandas y hacer un pequeño resumen para ver cómo se han comportado cada una de las métricas para un fichero csv mediante el comando:

```
python.exe mostrar_eval.py 2024_12_21_16_47_34_ragas_results.csv
```

En este caso hemos evaluado un dataset con 100 registros

	context_recall	factual_correctness	faithfulness	semantic_similarity	answer_relevancy	context_precision
count	100,00	100,00	100,00	100,00	100,00	99,00

mean	0,46	0,20	0,70	0,91	0,49	0,47
std	0,42	0,23	0,29	0,03	0,45	0,50
min	0,00	0,00	0,00	0,82	0,00	0,00
25%	0,00	0,00	0,50	0,88	0,00	0,00
50%	0,50	0,15	0,79	0,90	0,84	0,00
75%	1,00	0,34	0,93	0,93	0,91	1,00
max	1,00	0,89	1,00	0,97	0,98	1,00

Tabla 4: Estadísticas descriptivas del dataset ragas.

En la Figura 22 se puede ver la visualización en gráfico de histograma.

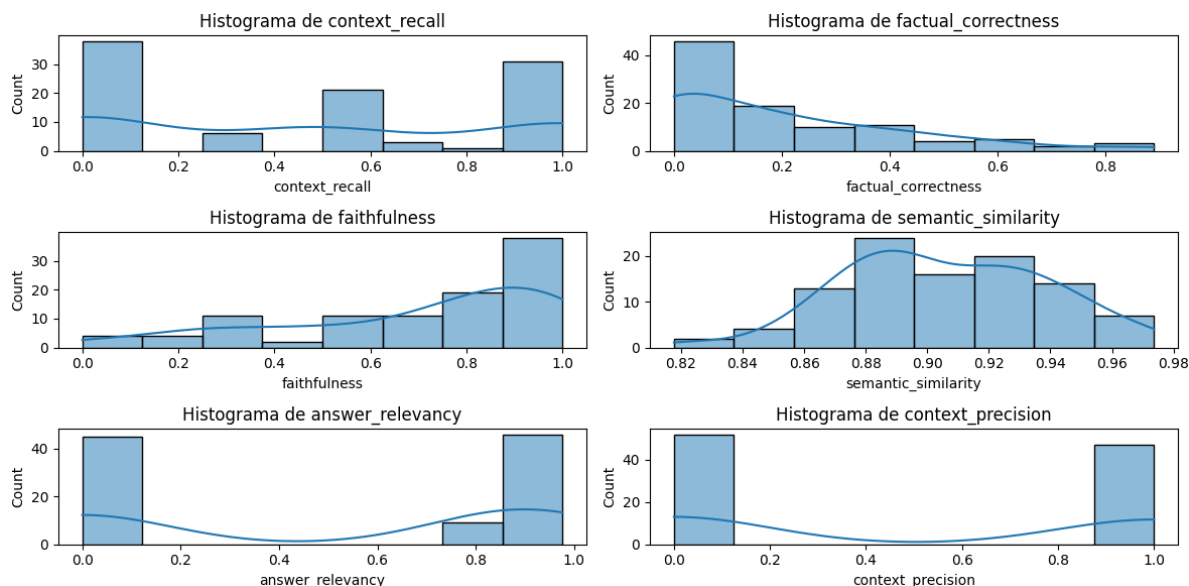


Figura 22: Histograma de las métricas dataset RAGAS

Según estos datos podemos evidenciar lo siguiente para cada métrica:

- *Context Recall* mide la capacidad del sistema para recuperar contexto relevante desde la base de datos. Con una media de 0.4625, recupera el 46% del contexto necesario en promedio. La desviación estándar de 0.4195 revela una alta variabilidad, y el rango, de 0 a 1, muestra desde fallos completos hasta recuperación total.
- *Factual Correctness* evalúa si las respuestas son factualmente correctas. La media de 0.20 indica que solo el 20% son correctas en promedio. La desviación estándar de 0.2261 refleja una moderada variabilidad, y el rango, de 0 a 0.89, evidencia que nunca alcanza una corrección total.
- *Faithfulness* analiza la fidelidad de las respuestas al contexto proporcionado. La media de 0.6958 indica una fidelidad promedio del 69.6%, mientras que la desviación estándar de 0.2904 señala cierta inconsistencia. El rango completo de 0 a 1 incluye respuestas infieles y totalmente fieles.

- *Semantic Similarity* mide la similitud semántica con el texto de referencia. Con una media de 0.9057, las respuestas son altamente similares, mientras que la baja desviación estándar de 0.0327 indica gran consistencia. El rango, de 0.8177 a 0.9737, refleja valores consistentemente altos.
- *Answer Relevancy* evalúa la relevancia de las respuestas respecto a las preguntas. La media de 0.4935 indica relevancia promedio del 49.4%, pero la desviación estándar de 0.4497 muestra una variabilidad significativa. El rango, de 0 a 0.9762, evidencia respuestas desde irrelevantes hasta altamente relevantes.
- *Context Precision* mide la precisión del contexto recuperado. Con una media de 0.4747, el sistema tiene un desempeño moderado. Sin embargo, la alta desviación estándar de 0.5019 y el rango de 0 a 1 muestran casos de precisión nula y perfecta.

En el anexo 8.6 se pueden ver ejemplos de resultados satisfactorios y no satisfactorios con preguntas sintéticas.

4.1.8. Resultados finales de evaluación.

Los resultados de la evaluación destacan varios aspectos clave del rendimiento del sistema:

- **Consistencia:** La métrica de *Semantic Similarity* muestra un desempeño consistentemente alto, con una desviación estándar muy baja. Esto indica que el modelo genera respuestas semánticamente similares al texto de referencia de manera uniforme y confiable.
- **Alta Variabilidad:** Métricas como *Answer Relevancy* y *Context Precision* presentan desviaciones estándar elevadas, lo que refleja comportamientos inconsistentes en términos de relevancia y precisión del contexto recuperado.
- **Rendimiento Razonable:** La métrica de *Faithfulness* tiene un valor promedio moderadamente alto (68.3%), lo que sugiere que las respuestas suelen ser fieles al contexto proporcionado. Sin embargo, aún hay margen para mejorar en casos específicos.
- **Rendimiento Bajo:** Métricas como *Factual Correctness* y *Context Recall* muestran valores promedio bajos acompañados de alta variabilidad. Esto pone en evidencia limitaciones significativas tanto en la recuperación efectiva del contexto como en la precisión factual de las respuestas generadas.

Fortalezas

El modelo destaca por generar respuestas consistentemente similares al texto de referencia, lo que indica un buen ajuste para captar el significado semántico. Además, en al menos el 75% de los casos, muestra un alto nivel de relevancia y precisión en la recuperación del contexto, ofreciendo una experiencia sólida en estas áreas. Algunas respuestas alcanzan niveles de relevancia y precisión completamente satisfactorios.

Debilidades

Existen inconsistencias notables en la relevancia de las respuestas y la precisión contextual, lo que puede impactar negativamente la experiencia del usuario. En casos extremos, tanto la relevancia como la precisión pueden ser nulas, lo que evidencia la necesidad de abordar estas deficiencias.

Áreas de Mejora

Es necesario mejorar la capacidad para recuperar contexto relevante y garantizar una mayor precisión factual en las respuestas. También resulta crítico reducir la variabilidad en métricas clave como *Answer Relevancy* y *Context Precision* para ofrecer un desempeño más consistente.

4.2. Resultados

Los resultados obtenidos tras la implementación del sistema RAG han sido satisfactorios en términos generales, logrando cumplir con gran parte de los objetivos planteados inicialmente. El sistema ha demostrado ser capaz de responder preguntas específicas relacionadas con el dominio de las ayudas y subvenciones públicas de forma precisa, utilizando información actualizada y relevante extraída mediante web scraping.

A nivel técnico, los modelos de lenguaje (LLMs) integrados con la base de datos vectorial han mostrado un rendimiento adecuado en la mayoría de los casos. El pipeline de RAG permite generar respuestas que combinan las capacidades de razonamiento del modelo con la precisión derivada de los documentos recuperados. Esto se traduce en un sistema que ofrece una experiencia de usuario coherente y útil.

No obstante, se han identificado áreas de mejora que permitirían optimizar tanto la experiencia de usuario como la eficacia del sistema:

- **Identificación de consultas:** Aunque el sistema maneja adecuadamente preguntas directas, su desempeño puede verse afectado cuando se plantean consultas ambiguas, demasiado generales o fuera del contexto previsto. Sería beneficioso implementar técnicas más avanzadas de clasificación y reformulación de consultas para mejorar su interpretación inicial.
- **Recuperación de datos de contexto:** En ocasiones, la información recuperada de la base de datos vectorial no es correcta y coherente con la pregunta proporcionada, lo que nos obliga necesariamente a mejorar el desempeño de esta.
- **Experiencia de usuario:** En ocasiones, las respuestas proporcionadas no son completamente claras o están redactadas de manera excesivamente técnica para ciertos perfiles de usuario. Esto indica la necesidad de mejorar la

generación de respuestas mediante ajustes en el modelo LLM o en el formato de presentación de la información recuperada.

A pesar de estos desafíos, el sistema ofrece un punto de partida sólido para futuras investigaciones y desarrollos en el área de RAG aplicado a sistemas de información especializados. En resumen, los resultados respaldan la viabilidad de esta tecnología en el dominio explorado, evidenciando su potencial como herramienta para la consulta y el acceso a grandes volúmenes de información estructurada y no estructurada.

5. Conclusiones y trabajos futuros

En cuanto a la consecución de los objetivos planteados inicialmente, se puede afirmar que la mayoría de los objetivos relacionados con el modelo LLM y la recuperación de información a través del API fueron alcanzados con éxito. Sin embargo, el evaluar el rendimiento de la base de datos vectorial no ofreció el resultado esperado. Se abordaron diferentes estrategias para mitigar este problema, pero el resultado no es del todo el esperado. Algunas de las adaptaciones realizadas han sido:

- Modificar el contenido del documento en diferentes formatos, JSON, YAML, un texto plano con toda la información relevante.
- Quitar información no relevante para mejorar el desempeño.
- Cambiar el modelo de embeddings.
- Cambiar el algoritmo de cercanía en la búsqueda.
- Probar varias bases de datos vectoriales diferentes.

En cuanto a la planificación, podemos decir que se ha seguido según lo previsto, aunque con algunos ajustes normales en algunas de las tareas que se han visto atrasadas por los problemas comentados.

El principal impacto no previsto fue la falta de funcionamiento óptimo de la base de datos vectorial, lo que ha afectado la integración y ha retrasado el trabajo. Este desafío ha sido mitigado parcialmente al continuar con la investigación y prueba de soluciones alternativas para mejorar el almacenamiento y la consulta de los datos.

La elección de las herramientas Ollama + ChromaDB fue acertada para la implementación del sistema de recuperación y generación de información basada en código Open Source siendo satisfactoria al ejecutarse en local. Sin embargo, al plantear el despliegue para su evaluación nos encontramos con muchos problemas para tener infraestructura con potencia para poder ejecutar esos modelos. Fue entonces cuando planteamos la posibilidad de desplegar en HuggingFace Spaces y vimos que era factible cambiando la estrategia utilizando

OPENAI + FAISS. Al haber hecho el sistema muy modularizado y parametrizable esta adaptación no nos llevó mucho tiempo.

Trabajos futuros

A continuación, se detallan los trabajos futuros que planteamos y clasificamos como prioritarios y secundarios.

- Trabajos prioritarios

Bajo nuestro punto de vista el trabajo prioritario se debe centrar en la mejora de rendimiento de base de datos vectorial. Se hace necesario abordar y revisar el comportamiento general de la base de datos vectorial abordándolo desde varias perspectivas:

- **Embeddings:** Usar otros embeddings o ajustar los parámetros de recuperación. Se evaluaron varios modelos de embedding, la mayoría multilingües, pero creemos que revisando un modelo de generación de embeddings solo en español y más de nicho puede mejorar el rendimiento considerablemente.
- **Preguntas y prompt:** Se hace necesario revisar y normalizar el texto introducido por el usuario con el objetivo de reducir el ruido de palabras no importantes para la consulta. Así como investigar alternativas en la generación del prompt para el LLM.
- **Enriquecimiento de datos:** Agregar metadatos relevantes para mejorar las búsquedas, como por ejemplo más datos de geolocalización informando siempre comunidad, provincia, municipio, población, etc. de forma que mejoren las búsquedas.
- **Comparativas tecnologías:** Sería interesante hacer una comparativa de las diferentes tecnologías (embeddings, LLMs, modelos) así como de los costes de los modelos no OpenSource.

- Trabajos secundarios

Mejoras de usabilidad y desempeño del chatbot:

- Hacer un asistente de preguntas y opciones prefijadas en base a los datos almacenados que posibilite la creación de un prompt concreto que nos permita hacer búsquedas también por metadatos y mejore el rendimiento de la base de datos vectorial.

Mejoras de IU

- Implementar autenticación de usuarios para almacenar información histórica, prefijar preguntas, hacer recomendaciones, etc...
- Añadir opciones para exportar las conversaciones o resultados en formatos como PDF o CSV.
- Incluir un módulo de retroalimentación para mejorar continuamente la calidad de las respuestas generadas.

6. Glosario

APIs	Abstract Programming Interfaces, APIs
BERT	Bidirectional Encoder Representations from Transformers
DOM	Document Object Model
HTML	HyperText Markup Language
IA	Inteligencia Artificial
IR	Information retrieval
LLM	Large Language Model
LSTM	Long Short-Term Memory
ML	Machine Learning
MVP	Minimum Viable Product
NLP	Natural Language Processor
Open LLM	Open Large Language Model
PLN	Procesamiento del lenguaje Naturals
POS	Part-of-Speech
RAG	Retrieval-Augmented Generation
RAGAS	Retrieval-Augmented Generation Analytics and Scoring
RNN	Recurrent Neural Network

7. Bibliografía

- A Comparison of Top Embedding Libraries for Generative AI [consulta: octubre de 2024] Disponible en <https://www.marktechpost.com/2024/07/28/a-comparison-of-top-embedding-libraries-for-generative-ai/>
- Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., ... & Simonyan, K. (2022). Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35, 23716-23736.
- Arasu, A., & Garcia-Molina, H. (2003, June). Extracting structured data from web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (pp. 337-348).
- Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463, No. 1999). New York: ACM press.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5, 135-146.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.
- Chang, C. H., Kayed, M., Girgis, M. R., & Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE transactions on knowledge and data engineering*, 18(10), 1411-1428.
- Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory*, 2(3), 113-124.
- Designing high-performing RAG systems [consulta: noviembre de 2024] <https://medium.com/@bijit211987/designing-high-performing-rag-systems-464260b76815>
- Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Evolución de los modelos de lenguaje de gran escala (LLMs: Large Language Models) [consulta: noviembre de 2024] Disponible en <https://borjafernandez.es/evolucion-de-los-modelos-de-lenguaje-de-gran-escala-llms-large-language-models/>
- Ferrara, E., De Meo, P., Fiumara, G., & Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-based systems*, 70, 301-323.

- Grandes modelos de lenguaje (LLM): una explicación con un mínimo de matemáticas y tecnicismos [consulta: octubre de 2024] Disponible en <https://www.understandingai.org/p/grandes-modelos-de-lenguaje-llm-una>
- Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020, November). Retrieval augmented language model pre-training. In International conference on machine learning (pp. 3929-3938). PMLR.
- Hochreiter, S. (1997). Long Short-term Memory. Neural Computation MIT-Press.
- Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., ... & Wei, F. (2023). Language is not all you need: Aligning perception with language models. Advances in Neural Information Processing Systems, 36, 72096-72109.
- Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. International Journal of Advances in Soft Computing & Its Applications, 13(3).
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. Advances in neural information processing systems, 35, 22199-22213.
- Laender, A. H., Ribeiro-Neto, B. A., Da Silva, A. S., & Teixeira, J. S. (2002). A brief survey of web data extraction tools. ACM Sigmod Record, 31(2), 84-93.
- Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.
- Li, J., Li, D., Savarese, S., & Hoi, S. (2023, July). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In International conference on machine learning (pp. 19730-19742). PMLR.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys, 55(9), 1-35.
- Liu, Y. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 364.
- Manning, C. D. (2009). An introduction to information retrieval.
- Mikolov, T. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 3781.
- Modelo extenso de lenguaje. Wikipedia [consulta: octubre de 2024] Disponible en: https://es.wikipedia.org/wiki/Modelo_extenso_de_lenguaje

- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35, 27730-27744.
- Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-286.
- Radford, A. (2018). Improving language understanding by generative pre-training.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748-8763). PMLR.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140), 1-67.
- Ragas [consulta: octubre de 2024] Disponible en: <https://docs.ragas.io/en/stable/>
- Retrieval-augmented generation. Wikipedia [consulta: octubre de 2024] Disponible en: https://en.wikipedia.org/wiki/Retrieval-augmented_generation
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533-536.
- Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613-620.
- Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4), 35-43.
- Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., ... & Wang, G. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.
- Streamlit [consulta: diciembre de 2024] <https://streamlit.io/>
- Team, G., Anil, R., Borgeaud, S., Alayrac, J. B., Yu, J., Soricut, R., ... & Blanco, L. (2023). Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

- Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. arXiv preprint arXiv:2104.08663.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
- Vaswani, A. (2017). Attention is all you need. Advances in Neural Information Processing Systems.
- Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., ... & Wei, F. (2023). Image as a foreign language: Beit pretraining for vision and vision-language tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 19175-19186).
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35, 24824-24837.
- Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), 36-45.
- What are Vector Embeddings? - Revolutionize Your Search Experience [consulta: octubre de 2024] Disponible en <https://qdrant.tech/articles/what-are-embeddings/>
- Why You Shouldn't Invest In Vector Databases? [consulta: noviembre de 2024] Disponible en <https://blog.det.life/why-you-shouldnt-invest-in-vector-databases-c0cd3f59d23c>
- Xu, G., Jin, P., Hao, L., Song, Y., Sun, L., & Yuan, L. (2024). LLaVA-o1: Let Vision Language Models Reason Step-by-Step. arXiv preprint arXiv:2411.10440.

8. Anexos

8.1. Planificación detallada

A continuación, se muestra un listado detallado de fases y tareas:

FASE DE PLANIFICACIÓN

- Definición del tema y objetivos del TFM
- Estudio de viabilidad (web scraping y LLMs)
- **HITO:** Definición del TFM: enunciado y entrega (M1)
- **HITO:** Entrega comité ética y convenios (M1)
- Revisión de la literatura sobre chatbots, web scraping y LLM
- **HITO:** Estado del arte: enunciado y entrega de la actividad (M2)
- Definición del alcance (funcionalidades del chatbot)
- Análisis de herramientas y tecnologías necesarias (bibliotecas para scraping, APIs de LLMs, infraestructura)
- Planificación inicial (cronograma, recursos, riesgos, legalidad del scraping)

ANÁLISIS DE REQUISITOS

- Requisitos funcionales del chatbot (interacciones, tareas, etc.)
- Requisitos del modelo LLM (Open LLM específico, recursos computacionales)
- Requisitos de web scraping (fuentes web, frecuencia, protección anti-bloqueo)
- Especificación de casos de uso del chatbot y escenarios

DISEÑO DEL SISTEMA

- Diseño de la arquitectura del chatbot
- Módulo de web scraping
- Módulo de integración con el LLM
- Módulo de interacción con el usuario
- Definición de flujos de datos (recopilación de datos, procesamiento y respuestas)
- Diseño del pipeline de scraping (cómo y cuándo se extraen datos)
- Definición de las APIs o interfaces necesarias
- Diseño de la interfaz de usuario (interfaz de texto o visual si aplica)

DESARROLLO DEL PROYECTO

- Preparación del entorno de entornos (instalación de herramientas para web scraping y LLM)
- Desarrollo
 - Desarrollo del módulo web scraping
 - Desarrollo almacenamiento en bd de conocimiento
 - Desarrollo e integración del LLM
 - Desarrollo de app web
- Pruebas unitarias y ajustes en cada módulo

FASE DE PRUEBAS

- Pruebas de WebScraping
- Pruebas del LLM
- Ajuste de parámetros del LLM
- Pruebas de integración entre los módulos
- Pruebas funcionales del chatbot
- **HITO:** Implementación: enunciado y entrega de la actividad (M3)

DOCUMENTACIÓN

- Documentación técnica del sistema
- Integración del LLM y diseño del chatbot
- Guía de usuario del chatbot
- Documentación de las pruebas realizadas (unitarias, integración)

REVISIÓN FINAL DEL TFM

- Redacción del documento final del TFM
- **HITO:** Redacción de la memoria: entrega preliminar (M4)
- Revisión con el tutor
- Correcciones sugeridas por el tutor
- **HITO:** Redacción de la memoria: entrega final (M4)
- Preparación del video TFM
- **HITO:** Presentación audiovisual del trabajo (M4)

DEFENSA DEL TFM

- Preparación de la presentación (diapositivas, demos)
- Ensayos de la defensa (simulaciones)
- **HITO:** Presentación ante el tribunal

Planificación - Diagrama de Gantt

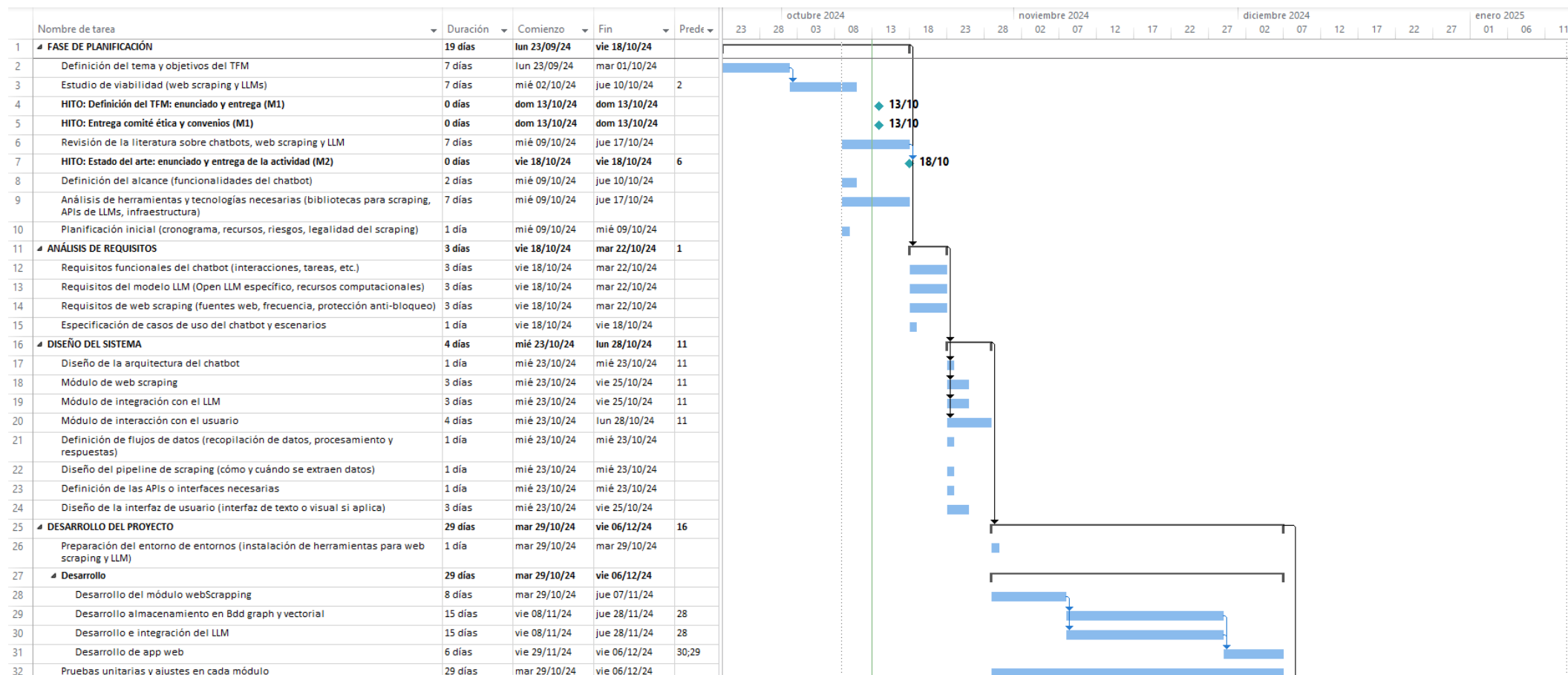


Figura 23: Diagrama de Gantt (parte 1)

Planificación - Diagrama de Gantt (continuación)

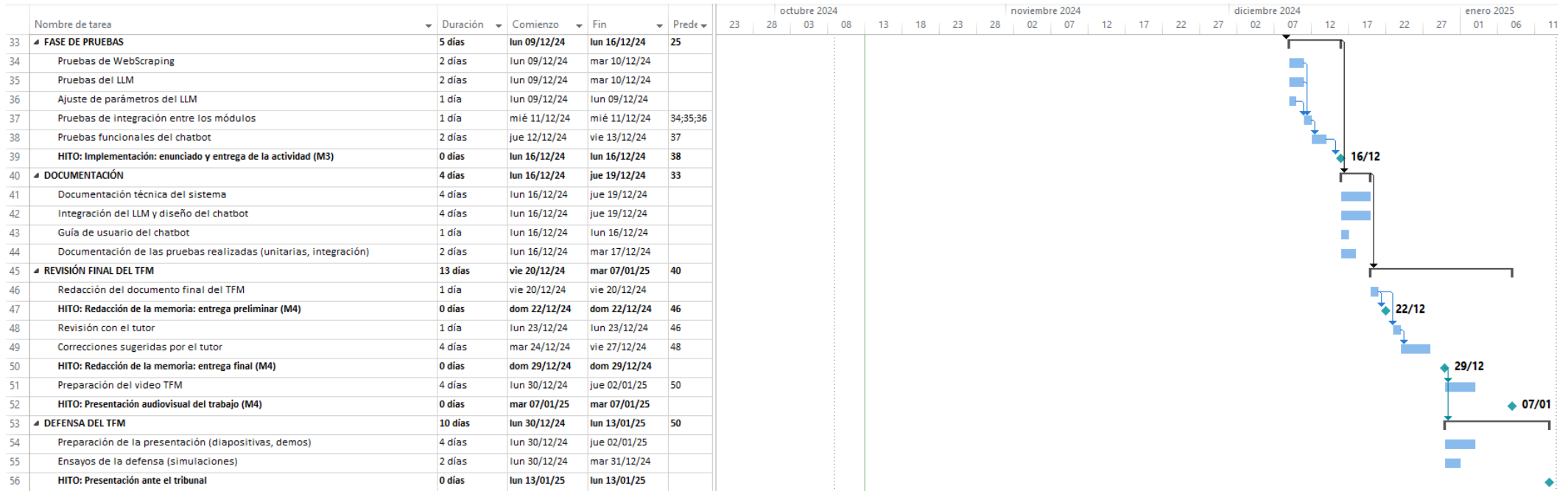


Figura 24: Diagrama de Gantt (parte 2)

8.2. Embeddings

Librerías de Embeddings

Existen múltiples librerías que implementan y optimizan estas técnicas de embeddings, cada una con particularidades en términos de rendimiento, facilidad de uso y capacidad de integración en diferentes plataformas de aprendizaje automático. En (A Comparison of Top Embedding Libraries for Generative AI, 2024) podemos ver una comparación reciente, de Julio del 2024, donde se comparan las 15 librerías de embeddings más utilizadas. A continuación, se hace un resumen de este trabajo:

OpenAI Embeddings

Son representaciones vectoriales generadas por modelos como GPT-3 y posteriores, que se entrenan para entender y generar texto de manera coherente. Estos modelos transforman texto en vectores de alta dimensión que capturan tanto el significado como el contexto de las palabras y frases en el texto.

Técnicas utilizadas.

Utiliza la arquitectura de Transformers permitiendo que el modelo considere el contexto completo de una palabra en relación con las palabras circundantes en una oración, lo que resulta en embeddings más contextuales. Se basan en el entrenamiento no supervisado con grandes volúmenes de texto sin etiquetar, lo que les permite aprender patrones de lenguaje y significado sin intervención humana. Estos modelos se pueden ajustar para tareas específicas mediante el uso de conjuntos de datos etiquetados, mejorando su rendimiento en aplicaciones concretas.

Ventajas.

Capturan el contexto y el significado dinámico de las palabras en función de su uso en oraciones, lo que los hace más precisos para tareas complejas de NLP.

Pueden ser utilizados para una amplia variedad de aplicaciones, como generación de texto, traducción automática, resumen de texto, y respuesta a preguntas.

Limitaciones.

Los modelos de OpenAI son complejos y requieren una cantidad significativa de recursos computacionales, tanto para el entrenamiento como para la inferencia, lo que puede ser un obstáculo para su implementación en entornos de bajo recursos.

La calidad de los embeddings depende del corpus de texto utilizado para el entrenamiento. Open AI ofrece acceso gratuito a sus modelos para un número limitado de consultas. También ofrece métodos de pago por consulta que ofrecen un consumo ilimitado del modelo.

HuggingFace Embeddings

Librería que proporciona una amplia variedad de modelos de embeddings contextuales. Es especialmente útil para aplicaciones que requieren interpretaciones dependientes del

contexto. HuggingFace ofrece una biblioteca llamada Transformers que permite a los usuarios acceder a numerosos modelos de lenguaje entrenados previamente, incluidos aquellos que generan embeddings de palabras y oraciones. Estos modelos pueden ser utilizados para diversas tareas de NLP, como clasificación de texto, análisis de sentimientos y generación de texto.

Técnicas utilizadas.

La biblioteca se basa en la arquitectura de Transformers, que utiliza mecanismos de atención para capturar relaciones de largo alcance en el texto. Modelos como BERT, GPT-2, RoBERTa y DistilBERT están disponibles y pueden generar embeddings contextuales. Los modelos se pueden ajustar con conjuntos de datos específicos para mejorar su rendimiento en tareas concretas.

Ventajas.

La biblioteca Transformers de HuggingFace es fácil de usar y está bien documentada, lo que permite a los desarrolladores implementar modelos de embeddings con poco esfuerzo. Permite elegir entre una gran cantidad de modelos entrenados previamente para diferentes idiomas y tareas, lo que ofrece flexibilidad según las necesidades específicas del proyecto. HuggingFace cuenta con una comunidad muy activa que contribuye con nuevos modelos, ejemplos y recursos.

Limitaciones.

Al igual que otros modelos de Transformers, los modelos de HuggingFace pueden ser intensivos en recursos, lo que puede ser un desafío para su implementación en dispositivos con capacidad limitada. La calidad de los embeddings generados depende de los datos utilizados para entrenar los modelos. Algunos modelos más avanzados pueden ser difíciles de ajustar o implementar correctamente sin una comprensión sólida de su funcionamiento interno.

Faiss

Faiss (Facebook AI Similarity Search) es una biblioteca de código abierto para la búsqueda eficiente y la agrupación de vectores de alta dimensión. Está optimizada para operaciones de similitud y clustering, y se utiliza ampliamente en sistemas de recomendación y búsqueda de información.

Técnicas utilizadas.

Realiza la indexación de vectores mediante árboles de productos y otras estructuras de datos optimizadas. Implementa de técnicas de reducción de dimensionalidad como PCA y LSH (Locality Sensitive Hashing).

Ventajas.

Está optimizada para búsquedas y recuperación semántica en tiempo real, especialmente en GPUs. Resulta muy eficiente en términos de velocidad y memoria, lo que permite el manejo de grandes conjuntos de datos. Es compatible con operaciones de búsqueda exacta

y aproximada. También permite la escalabilidad para hacer uso de ella en sistemas distribuidos.

Limitaciones.

No está orientada a generación de embeddings desde texto. Puede ser complejo de configurar y ajustar para casos de uso específicos. Requiere una comprensión profunda de los algoritmos de búsqueda de similitud para optimizar su uso.

FastText

FastText es una herramienta de procesamiento de lenguaje natural que permite la creación de representaciones vectoriales (embeddings) de palabras, considerando subpalabras para capturar mejor la morfología del lenguaje

Técnicas utilizadas.

Aprendizaje de representaciones de palabras a partir de n-gramas, lo que permite capturar variaciones morfológicas.

Ventajas.

Proporciona embeddings más robustos para lenguajes con rica morfología. Es rápido y eficiente, adecuado para grandes conjuntos de datos.

Limitaciones.

Limitado en embeddings contextuales; tamaño de modelo mayor

A menudo no alcanza la calidad de embeddings de modelos más complejos como BERT para tareas de contexto profundo.

Menos eficaz en el manejo de frases y oraciones comparado con modelos basados en Transformers.

Gensim Word

Es conocida por su facilidad de uso y eficiencia en la generación de embeddings. Su uso está orientado a tareas donde los embeddings contextuales no son indispensables. Gensim permite crear y trabajar con modelos de embeddings de palabras como Word2Vec, FastText y GloVe. Facilita la representación de palabras como vectores densos en un espacio de alta dimensión, donde la similitud entre las palabras se puede medir mediante distancias en ese espacio.

Técnicas utilizadas.

Word2Vec, FastText y GloVe

- Word2Vec: Utiliza dos arquitecturas principales, Continuous Bag of Words (CBOW) y Skip-Gram, para aprender representaciones de palabras a partir de grandes corpus de texto.
- FastText: Al igual que Word2Vec, pero tiene en cuenta subpalabras (n-gramas), lo que permite capturar mejor la morfología de palabras raras o compuestas.

- GloVe: Implementación de la técnica Global Vectors for Word Representation que se basa en la matriz de coocurrencia de palabras, lo que ayuda a construir embeddings considerando las relaciones globales en el corpus.

Ventajas.

Gensim tiene una interfaz simple y bien documentada, lo que facilita la creación y manipulación de modelos de embeddings. Está optimizada para trabajar con grandes conjuntos de datos. Permite entrenar modelos personalizados a partir de datos específicos, y también puedes utilizar modelos entrenados previamente, lo que ahorra tiempo y recursos. Además, permite ser integrada fácilmente con otras bibliotecas de procesamiento de texto, como NLTK y SpaCy.

Limitaciones.

Aunque es eficiente, entrenar modelos de embeddings en grandes corpus puede ser intensivo en recursos, especialmente con modelos complejos. Comparado con modelos más avanzados basados en Transformers (como BERT o GPT), los embeddings de Gensim pueden no captar adecuadamente las relaciones contextuales profundas entre palabras. Para algunas tareas de NLP, los embeddings de palabras generados por Gensim pueden no ser suficientes en comparación con enfoques más avanzados que utilizan atención o redes neuronales profundas.

Otras librerías para tener en cuenta son las siguientes:

- **GTE-Base, GTE-Large, GTE-Small:** Son modelos de embeddings que se centran en tareas generales de texto, con variaciones en el tamaño para equilibrar la precisión y la eficiencia computacional.
- **E5-Small:** Un modelo optimizado para tareas de búsqueda y recuperación de información, que combina representaciones de texto eficaces en un tamaño compacto.
- **MultiLingual BERT:** Un modelo pre entrenado con múltiples conjuntos de datos de diferentes idiomas y diseñado para tareas de NLP en un contexto multilingüe.
- **RoBERTa (2022):** Una variante mejorada de BERT que utiliza más datos de entrenamiento y ajustes en la técnica de enmascaramiento, ofreciendo un mejor rendimiento en diversas tareas de NLP.
- **MPNet V2:** Un modelo que combina las ventajas de BERT y otros enfoques de aprendizaje por contraste, optimizando tanto la calidad de los embeddings como la eficiencia.
- **SciBERT Science-Vocabulary Uncased:** Un modelo específico para textos científicos, entrenado con vocabulario técnico y publicaciones académicas, lo que mejora la comprensión en este dominio.

- **Longformer Base 4096:** Diseñado para manejar secuencias largas, este modelo utiliza atención local y global para procesar textos extensos sin perder contexto.
- **DistilBERT Base Uncased:** Una versión más pequeña y rápida de BERT, que ofrece un buen equilibrio entre rendimiento y eficiencia, siendo ideal para aplicaciones donde se requiere velocidad.

Las librerías más generalistas, basadas en modelos como Word2Vec, GloVe y BERT, son ampliamente usadas debido a su eficacia en tareas generales de procesamiento de lenguaje natural (NLP), como clasificación, análisis de sentimientos y generación de texto. Por otro lado, las librerías y modelos más de nicho, como SciBERT y E5-Small, están diseñados para aplicaciones específicas, como puede ser la investigación científica o la búsqueda de información en texto, ofreciendo resultados mejores en contextos particulares.

Los embeddings han transformado significativamente el procesamiento del lenguaje al facilitar una representación vectorial semántica de palabras y frases, constituyendo la base de numerosas aplicaciones en modelos de lenguaje. La selección de la técnica y la biblioteca de embeddings a emplear está determinada por los requisitos específicos de cada proyecto.

8.3. Comparativa de Bases de Datos Vectoriales

En la actualidad existen varias bases de datos vectoriales que han sido optimizadas para su uso en sistemas RAG y LLM. A continuación, se presentan algunas de las más utilizadas:

Producto	Descripción
Pinecone	Pinecone es una base de datos vectorial de uso comercial que permite almacenar, indexar y consultar vectores de alta dimensión en tiempo real. Su infraestructura permite también el escalado horizontal, lo que es ventajoso para aplicaciones de gran envergadura que requieren alta disponibilidad y rendimiento en consulta.
Weaviate	Weaviate es una base de datos vectorial de código abierto que permite almacenar y consultar datos en forma de grafos semánticos. Su arquitectura incorpora directamente las capacidades de embeddings de modelos de lenguaje, haciendo que la recuperación de datos sea altamente contextual.
Milvus	Milvus es otra base de datos vectorial de código abierto diseñada para soportar un gran volumen de datos y realizar consultas distribuidas de alta disponibilidad. Su arquitectura distribuida permite la consulta eficiente de millones de vectores. Milvus se destaca por su capacidad de escalado horizontal y su integración con ecosistemas de aprendizaje automático como TensorFlow y PyTorch.
Qdrant	Qdrant es una base de datos vectorial de alto rendimiento optimizada para la búsqueda de similitud de alta precisión. Su implementación en sistemas RAG permite consultas rápidas y precisas, y ofrece compatibilidad con modelos de embeddings de última generación.
Chroma	Chroma es una base de datos vectorial de código abierto diseñada para el almacenamiento y la recuperación de datos en forma de vectores, especialmente enfocada en aplicaciones de inteligencia artificial y aprendizaje automático. Su principal ventaja radica en su simplicidad y facilidad de uso.

Tabla 6: Comparativa de bases de datos vectoriales

En la Figura 25 hay una copia de la representación gráfica que aparece en (Det Life, 2024) donde se puede ver una comparación entre algunas de las bases de datos vectoriales desde 2 perspectivas distintas, por un aparte si es una base de datos vectorial al uso o bien es una base de datos más generalista que soporta el almacenamiento de vectores y por otra parte si son de código abierto o bien requiere de una licencia comercial.

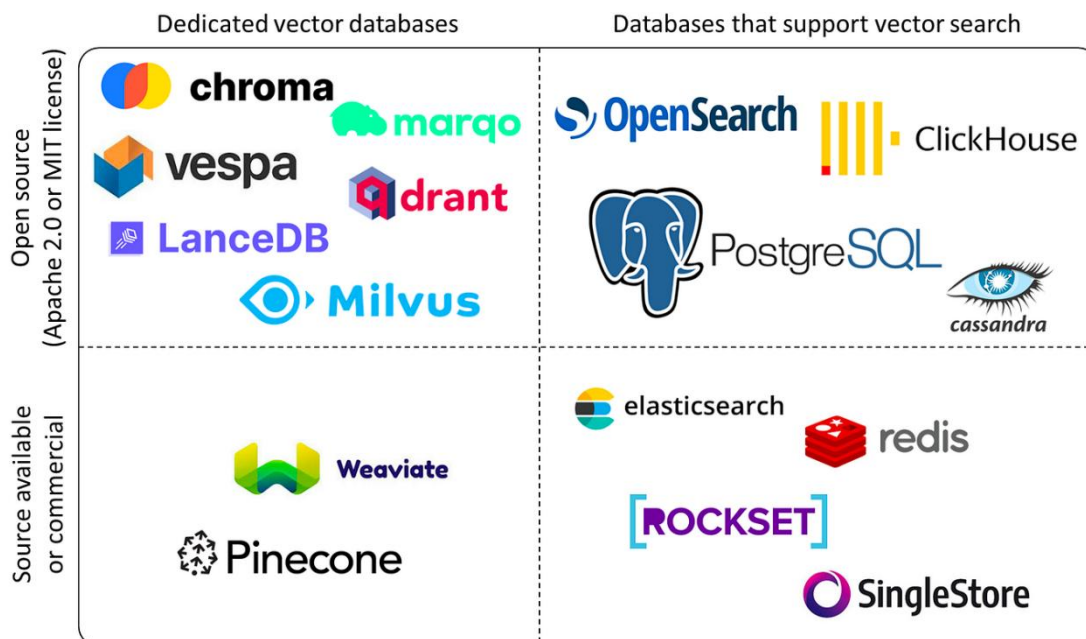


Figura 25: Comparativa BD Vectoriales desde 2 perspectivas

Las bases de datos vectoriales han permitido el desarrollo de aplicaciones avanzadas en diversos sectores:

- **Asistentes Virtuales y Chatbots:** Sistemas como Chroma, FAISS y Pinecone se utilizan para optimizar la recuperación de datos relevantes en asistentes virtuales, mejorando la precisión de las respuestas generadas por el LLM.
- **Motores de Recomendación:** Weaviate y Qdrant han sido empleados en motores de recomendación para personalizar sugerencias basadas en la similitud semántica entre preferencias del usuario y contenidos almacenados.
- **Búsqueda Semántica en Empresas:** Milvus ha sido adoptado en aplicaciones empresariales de búsqueda semántica, facilitando el acceso rápido y contextualizado a grandes volúmenes de documentación corporativa.

8.4. Retrieval-Augmented Generation Analytics and Scoring (RAGAS)

Descripción

La técnica de Retrieval-Augmented Generation Analytics and Scoring (RAGAS) es un método reciente para evaluar la calidad de los sistemas LLMs, especialmente aquellos que incorporan técnicas de generación aumentada con recuperación (Retrieval-Augmented Generation, RAG). Este método, basado en métricas específicas, permite evaluar la precisión y efectividad de los modelos en tareas de recuperación de información y generación de lenguaje natural a partir de datos recuperados.

Métricas

Una métrica es una medida cuantitativa que se utiliza para evaluar el rendimiento de una aplicación de IA. Las métricas ayudan a evaluar el rendimiento de la aplicación y de los componentes individuales que la componen en relación con los datos de prueba proporcionados. Proporcionan una base numérica para la comparación, la optimización y la toma de decisiones durante todo el proceso de desarrollo e implementación de la aplicación. Las métricas son fundamentales para:

- Selección de componentes: las métricas se pueden utilizar para comparar diferentes componentes de la aplicación de IA como LLM, Retriever, configuración del agente, etc. con sus propios datos y seleccionar el mejor entre diferentes opciones.
- Diagnóstico y depuración de errores: las métricas ayudan a identificar qué parte de la aplicación está causando errores o un rendimiento subóptimo, lo que facilita la depuración y el refinamiento.
- Monitorización y mantenimiento continuos: las métricas permiten rastrear el rendimiento de una aplicación de IA a lo largo del tiempo, lo que ayuda a detectar y responder a problemas como la desviación de datos, la degradación del modelo o los cambios en los requisitos del usuario.

Diferentes tipos de métricas

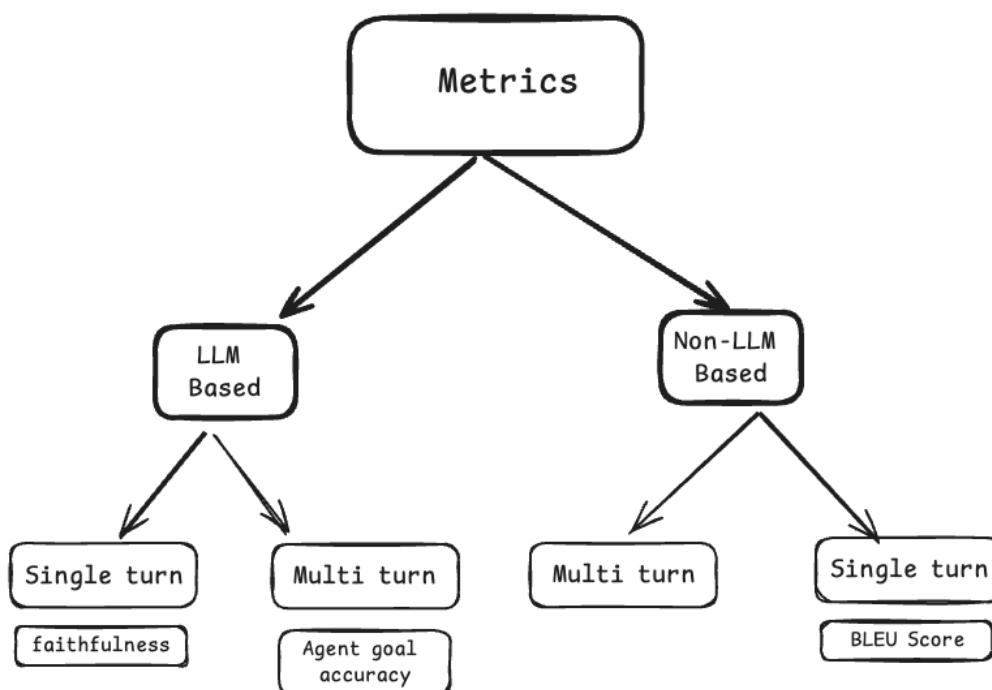


Figura 26: Árbol Métricas RAGAS

Las métricas se pueden clasificar en dos categorías según el mecanismo utilizado y según el tipo de datos que avalúan. En la Tabla 5 se puede ver esta clasificación:

Clasificación	Métrica	Descripción
Según mecanismo utilizado	Basadas en LLM	Utilizan LLM para realizar la evaluación. Puede ser necesario una o más llamadas al LLM para llegar a la puntuación o resultado.
	No Basadas en LLM	Son deterministas y se pueden utilizar para evaluar el rendimiento de la aplicación de IA sin utilizar LLM.
Según tipo de dato a evaluar	De un solo turno	Evalúan el rendimiento de la aplicación de IA en función de un solo turno de interacción entre el usuario y la IA.
	Multi-turno	Evalúan el rendimiento de la aplicación de IA en función de múltiples turnos de interacción entre el usuario y la IA.

Tabla 7: Clasificación Métricas RAGAS

Lista de Métricas disponibles

RAGAS ofrece un conjunto de métricas de evaluación que se pueden utilizar para medir el rendimiento de su aplicación LLM. Estas métricas están diseñadas para ayudarlo a medir objetivamente el rendimiento de su aplicación. Hay métricas disponibles para diferentes aplicaciones y tareas, como flujos de trabajo RAG y Agentic.

Cada métrica es esencialmente un paradigma diseñado para evaluar un aspecto particular de la aplicación. Las métricas basadas en LLM pueden utilizar una o más llamadas LLM para llegar a la puntuación o al resultado. También se pueden modificar o escribir métricas propias utilizando ragas.

Recuperación Generación Aumentada

- Precisión del contexto (*Context Precision*). La precisión del contexto es una métrica que mide la proporción de fragmentos relevantes en el contexto.
- Recordatorio del contexto (*Context Recall*). El recuerdo de contexto mide cuántos documentos relevantes (o fragmentos de información) se recuperaron correctamente. Se centra en no omitir resultados importantes.
- Recordatorio de entidades de contexto (*Context Entities Recall*). La métrica proporciona la medida de recuperación del contexto recuperado, en función de la cantidad de entidades presentes.
- Sensibilidad al ruido (*Noise Sensitivity*). La métrica mide la frecuencia con la que un sistema comete errores al proporcionar respuestas incorrectas al utilizar documentos recuperados relevantes o irrelevantes.
- Relevancia de la respuesta (*Response Relevancy*). La métrica se centra en evaluar la pertinencia de la respuesta generada para la pregunta planteada. Se asigna una puntuación más baja a las respuestas que están incompletas o contienen información redundante y las puntuaciones más altas indican una mayor relevancia.
- Fidelidad (*Faithfulness*). La métrica mide la coherencia fáctica de la respuesta generada en relación con el contexto dado. Se calcula a partir de la respuesta y el contexto recuperado.
- Fidelidad multimodal (*Multimodal Faithfulness*). La métrica mide la coherencia fáctica de la respuesta generada en relación con el contexto visual y textual. Se calcula a partir de la respuesta, el contexto textual recuperado y el contexto visual.
- Relevancia multimodal (*Multimodal Relevance*). La métrica mide la relevancia de la respuesta generada en relación con el contexto visual y textual. Se calcula a partir de la entrada del usuario, la respuesta y los contextos recuperados (tanto visuales como textuales).

Existen otros tipos de medidas agrupadas por la funcionalidad que cubren:

- Casos de uso de agentes o herramientas
 - Adherencia al tema
 - Llamada de herramienta Precisión
 - Precisión de los objetivos del agente
- Comparación de lenguaje natural
 - Corrección fáctica
 - Similitud semántica
 - Similitud de cadenas no LLM
 - Puntuación BLEU
 - Puntuación de ROUGE
 - Presencia de cuerdas
 - Coincidencia exacta
- SQL
 - Puntuación de Datacompy basada en ejecución
 - Equivalencia de consultas SQL
- Propósito general
 - Crítica de aspectos
 - Puntuación de criterios simples
 - Puntuación basada en rúbricas
 - Puntuación de rúbricas específicas de cada instancia

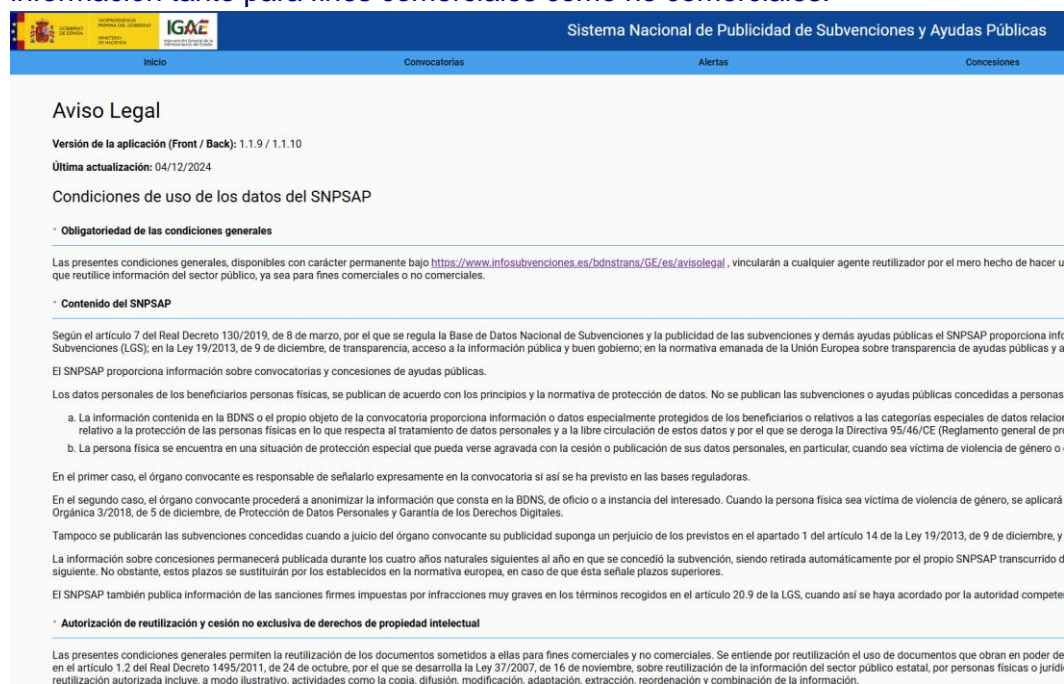
Implementación de RAGAS para Evaluación de LLMs implica la creación de un conjunto de pruebas que contengan preguntas y respuestas generadas por el modelo. Estas respuestas se comparan contra un conjunto de respuestas esperadas usando las métricas mencionadas en los puntos anteriores.

8.5. Consideraciones legales al obtener datos de API

Durante las llamadas al API si se revisó la respuesta, y en todas las llamadas se incluye una advertencia que se detalla a continuación:

"advertencia": "La reutilización de los datos del Sistema Nacional de Publicidad de Subvenciones y Ayudas Públicas está sujeta a una serie de restricciones y consideraciones legales. Consulte el aviso legal en <https://www.infosubvenciones.es/bdnstrans/GE/es/avisolegal> .Además, el usuario debe ser consciente que la información presentada es de naturaleza dinámica, y que los datos pueden ser sometidos a correcciones, inserciones, modificaciones y eliminaciones en momentos posteriores a su extracción."

En la Figura 27 vemos el aviso legal y observamos que se permite la reutilización de la información tanto para fines comerciales como no comerciales.



Sistema Nacional de Publicidad de Subvenciones y Ayudas Públicas

Inicio Convocatorias Alertas Concesiones

Aviso Legal

Versión de la aplicación (Front / Back): 1.1.9 / 1.1.10
Última actualización: 04/12/2024

Condiciones de uso de los datos del SNPSAP

- Obligatoriedad de las condiciones generales**

Las presentes condiciones generales, disponibles con carácter permanente bajo <https://www.infosubvenciones.es/bdnstrans/GE/es/avisolegal>, vincularán a cualquier agente reutilizador por el mero hecho de hacer u que reutilice información del sector público, ya sea para fines comerciales o no comerciales.

- Contenido del SNPSAP**

Según el artículo 7 del Real Decreto 130/2019, de 8 de marzo, por el que se regula la Base de Datos Nacional de Subvenciones y la publicidad de las subvenciones y demás ayudas públicas el SNPSAP proporciona información sobre convocatorias y concesiones de ayudas públicas.

Los datos personales de los beneficiarios personas físicas, se publican de acuerdo con los principios y la normativa de protección de datos. No se publican las subvenciones o ayudas públicas concedidas a personas físicas.

a. La información contenida en la BDNS o el propio objeto de la convocatoria proporciona información o datos especialmente protegidos de los beneficiarios o relativos a las categorías especiales de datos relativos a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos).

b. La persona física se encuentra en una situación de protección especial que pueda verse agravada con la cesión o publicación de sus datos personales, en particular, cuando sea víctima de violencia de género o víctima de trata de seres humanos.

En el primer caso, el órgano convocante es responsable de señalarlo expresamente en la convocatoria si así se ha previsto en las bases reguladoras.

En el segundo caso, el órgano convocante procederá a anonimizar la información que consta en la BDNS, de oficio o a instancia del interesado. Cuando la persona física sea víctima de violencia de género, se aplicará la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y Garantía de los Derechos Digitales.

Tampoco se publicarán las subvenciones concedidas cuando a juicio del órgano convocante su publicidad suponga un perjuicio de los previstos en el apartado 1 del artículo 14 de la Ley 19/2013, de 9 de diciembre, y de la Ley 19/2013, de 9 de diciembre, de Transparencia.

La información sobre concesiones permanecerá publicada durante los cuatro años naturales siguientes al año en que se concedió la subvención, siendo retirada automáticamente por el propio SNPSAP transcurrido el siguiente. No obstante, estos plazos se sustituirán por los establecidos en la normativa europea, en caso de que ésta señale plazos superiores.

El SNPSAP también publica información de las sanciones firmes impuestas por infracciones muy graves en los términos recogidos en el artículo 20.9 de la LGS, cuando así se haya acordado por la autoridad competente.

- Autorización de reutilización y cesión no exclusiva de derechos de propiedad intelectual**

Las presentes condiciones generales permiten la reutilización de los documentos sometidos a ellas para fines comerciales y no comerciales. Se entiende por reutilización el uso de documentos que obran en poder de terceros, en el ámbito de la actividad económica, por personas físicas o jurídicas, que no sean las personas físicas o jurídicas titulares de los derechos de propiedad intelectual, para la explotación económica de la información.

Figura 27: Página de Aviso Legal

En las condiciones generales para la reutilización de los datos vemos que indica lo siguiente:

Son de aplicación las siguientes condiciones generales para la reutilización de los documentos sometidos a ellas:

- *Está prohibido desnaturalizar el sentido de la información.*
- *Debe citarse la fuente de los documentos objeto de la reutilización. Esta cita podrá realizarse de la siguiente manera: "Origen de los datos: Intervención General de la Administración del Estado".*
- *Debe mencionarse la fecha de la última actualización de los documentos objeto de la reutilización, siempre cuando estuviera incluida en el documento original.*

- *En caso de disociación de datos personales deberá citarse su realización y quién la efectúa.*
- *No se podrá indicar, insinuar o sugerir que la Intervención General de la Administración del Estado participa, patrocina o apoya la reutilización que se lleve a cabo con ella.*
- *Deben conservarse, no alterarse ni suprimirse los metadatos sobre la fecha de actualización y las condiciones de reutilización aplicables incluidos, en su caso, en el documento puesto a disposición para su reutilización.*

8.6. Datasets para evaluación con RAGAS

Descripción

Para la evaluación del sistema mediante RAGAS necesitamos datasets de evaluación con la información necesaria. Se ha abordado esta estrategia desde 3 vertientes distintas:

- Evaluación con dataset con datos reales: Básicamente recopilamos todas las interacciones reales con la aplicación y las almacenamos para evaluarlas posteriormente.
- Evaluación con dataset sintético: En este caso simulamos preguntas típicas con diferentes parámetros y los utilizamos como si fuera un usuario real. Estos datos se almacenarán junto con los del uso de la aplicación.
- Evaluación con dataset generado con RAGAS: El framework de RAGAS permite, a partir de información almacenada en la base de datos vectorial, generar datasets que se utilizan para simular el uso como el de un usuario final.

En las siguientes secciones veremos ejemplos de cada uno de estos usos.

Dataset datos reales

En este caso no tenemos como tal un dataset de preguntas, sino que es el usuario el que mediante el uso de la aplicación genera esos datasets de preguntas, algunos ejemplos serían los siguientes:

- puedes listarme todas las ayudas que tengas?
- qué ayudas tienes para autónomos?
- que ayudas tienes de Gipuzkoa?

Dataset sintético

Para generar estos datasets sintéticos nos hemos basado en plantillas de preguntas genéricas parametrizadas que vamos cambiando el valor del parámetro. Estas son las plantillas que se han preparado:

```
plantillas = {
    "región": "¿Qué ayudas hay disponibles en {item}?",
    "sector": "¿Hay subvenciones para el sector {item}?",
    "tipo_beneficiario": "¿Existen ayudas para {item} en alguna región?",
    "organismo": "¿Qué ayudas hay de {item}?",
    "palabras_clave": "Busco subvenciones para {item}."
}
```

Los parámetros de región, sector, beneficiario, organismo y palabras clave están también parametrizados en JSON con estos valores:

```
regiones = ["Murcia", "Cataluña", "Madrid", "Valencia", "Barcelona", "San Juan de Alicante",
"Castellón", "Paiporta"]
```

sectores = ["Agricultura", "ganadería", "Industria", "tecnológica", "Salud", "bienestar", "Educación", "formación", "Construcción", "vivienda", "Energías renovables", "Transporte", "logística", "Industria manufacturera", "Turismo", "hostelería", "Arte", "cultura", "Comercio", "distribución", "Sector financiero", "Servicios sociales", "Innovación", "desarrollo", "Medio ambiente", "sostenibilidad", "Pesca", "acuicultura", "Automoción", "movilidad sostenible", "Ciencias", "tecnología espacial", "Industria alimentaria", "Telecomunicaciones", "medios digitales"]

tipos_beneficiarios = [
 "PYME",
 "Autónomos",
 "Personas físicas",
 "Grandes empresas",
 "Startups",
 "Asociaciones sin ánimo de lucro",
 "Entidades públicas",
 "Comunidades de vecinos",
 "Agricultores y ganaderos",
 "Cooperativas",
 "Fundaciones",
 "Estudiantes",
 "Investigadores",
 "Emprendedores",
 "Personas en situación de vulnerabilidad",
 "ONGs",
 "Administraciones locales",
 "Centros educativos",
 "Pequeñas y medianas empresas del sector tecnológico",
 "Personas mayores o jubilados"
]

organismos_emisores = [
 "Diputación de Alicante",
 "Ayuntamiento de Barcelona",
 "Gobierno de Andalucía",
 "Consejería de Educación de Madrid",
 "Ministerio de Industria, Comercio y Turismo",
 "Diputación de Valencia",
 "Ayuntamiento de Sevilla",
 "Junta de Extremadura",
 "Gobierno de Canarias",
 "Gobierno de Castilla y León",
 "Diputación de Málaga",
 "Consejería de Sanidad de Galicia",
]

*"Ayuntamiento de Bilbao",
"Gobierno Vasco",
"Consejo Insular de Mallorca",
"Gobierno de Aragón",
"Cabildo de Tenerife",
"Consejería de Agricultura de La Rioja",
"Ministerio para la Transición Ecológica y el Reto Demográfico",
"Consejería de Fomento de Castilla-La Mancha"*

]

palabras_clave = [*"mejora de viviendas", "eficiencia energética", "innovación empresarial", "DANA"*]

A través de un script en python somos capaces de generar un fichero de texto plano que utilizaremos como datos para simular el uso de la aplicación como si fuera un usuario real. A continuación, puede ver un ejemplo de la generación de este fichero:

*¿Qué ayudas hay disponibles en Murcia?
¿Qué ayudas hay disponibles en Cataluña?
¿Qué ayudas hay disponibles en Madrid?
¿Qué ayudas hay disponibles en Valencia?
¿Qué ayudas hay disponibles en Barcelona?
¿Qué ayudas hay disponibles en San Juan de Alicante?
¿Qué ayudas hay disponibles en Castellón?
¿Qué ayudas hay disponibles en Paiporta?
¿Hay subvenciones para el sector Agricultura?
¿Hay subvenciones para el sector ganadería?
¿Hay subvenciones para el sector Industria?
¿Hay subvenciones para el sector tecnológica?
¿Hay subvenciones para el sector Salud?
¿Hay subvenciones para el sector bienestar?
...*

Como se ha comentado en la sección 4.2.4.2 este fichero se generaría con el siguiente comando:

python.exe buscar_batch_sintetico.py dataset_sintetico.txt

Dataset generado con RAGAS

En este caso nos basamos en la información almacenada en la base de datos vectorial para que mediante el LLM sea capaz de generar las preguntas pertinentes junto con la respuesta y de dónde ha sacado la información.

Como se ha comentado anteriormente en la sección 4.2.4.2 la generación de este dataset se realiza mediante el siguiente comando:

```
python.exe buscar_batch_ragas.py
dataset_ragas_2024_12_21_15_56_21_ragas_results.csv
```

Un ejemplo de generación de este dataset se muestra en estas 2 preguntas generadas:

```
"user_input";"reference_contexts";"reference";"synthesizer_name"
"¿Qué tipo de ayuda se ofrece en la convocatoria del ayuntamiento de Pontes de García Rodríguez?";"[ detalle de la convocatoria de ayuda o subvención: 804940 enlace a convocatoria: https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/804940 órgano, comunidad, autonomía, provincia o ayuntamiento convocante: pontes de garcía rodríguez, as - ayuntamiento de as pontes de garcía rodríguez fecha de recepción: 2024-12-20t18:38:26+01:00 tipo de ayuda: subvención y entrega dineraria sin contraprestación tipo de convocatoria: concurrencia competitiva - canónica presupuesto total: 34630 euros descripción: convocatoria de subvenciones para la organización de eventos deportivo-turísticos para el año 2024 tipos de beneficiarios: personas jurídicas que no desarrollan actividad económica sectores involucrados: actividades artísticas, recreativas y de entretenimiento región de impacto: es111 - a coruña finalidad: otras prestaciones económicas bases reguladoras: bases reguladoras de subvenciones para la organización de eventos deportivo-turísticos para el año 2024 publicación en diario oficial: sí estado de convocatoria abierta: no inicio de convocatoria: al día siguiente de la publicación del extracto de la convocatoria en el bop fin de convocatoria: cinco días naturales a partir de la publicación del extracto en el bop ]";"La convocatoria del ayuntamiento de Pontes de García Rodríguez ofrece subvenciones y entrega dineraria sin contraprestación para la organización de eventos deportivo-turísticos para el año 2024."; "single_hop_specifc_query_synthesizer"
```

```
"¿Qué es la concurrencia competitiva en el contexto de la convocatoria de ayudas del Ayuntamiento de Salamanca?";"[ detalle de la convocatoria de ayuda o subvención: 804880 enlace a convocatoria: https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/804880 órgano, comunidad, autonomía, provincia o ayuntamiento convocante: salamanca - ayuntamiento de salamanca fecha de recepción: 2024-12-20t14:48:37+01:00 tipo de ayuda: subvención y entrega dineraria sin contraprestación tipo de convocatoria: concurrencia competitiva - canónica presupuesto total: 33000 euros descripción: convocatoria 2025 apoyo a la creación escénica tipos de beneficiarios: pyme y personas físicas que desarrollan actividad económica sectores involucrados: actividades artísticas, recreativas y de entretenimiento región de impacto: es415 - salamanca finalidad: cultura bases reguladoras: presupuesto fundación salamanca ciudad de cultura y saberes publicación en diario oficial: sí estado de convocatoria abierta: no inicio de convocatoria: al día siguiente de su publicación en el bob salamanca fin de convocatoria: ]";"La concurrencia competitiva es el tipo de convocatoria mencionada en la ayuda o subvención del Ayuntamiento de Salamanca, que se refiere a
```

un proceso donde múltiples solicitantes compiten por un número limitado de ayudas disponibles."; "single_hop_specifc_query_synthesizer"

Los 3 primeros campos son los importantes ya que son los utilizados posteriormente para de una forma desatendida ver la respuesta del sistema para posteriormente evaluarlo mediante RAGAS.

Como se pude ver, las preguntas están más bien orientadas a contenidos concretos de los documentos almacenados en la base de datos y no tanto a como un usuario realizaría las preguntas. Parece poco probable que un usuario haga una pregunta como la primera

"¿Qué tipo de ayuda se ofrece en la convocatoria del ayuntamiento de Pontes de García Rodríguez?"

Es por eso por lo que se decidió abordar la evaluación del sistema mediante estas 3 perspectivas distintas con el objetivo de que la evaluación fuera lo más realista.

8.7. Ejemplos de resultados de la evaluación

Ejemplos satisfactorios en la evaluación

- **Ejemplo satisfactorio 1: (Generada con RAGAS)**

Fichero resultados: `ragas_eval/results/2025_01_03_19_45_10_ragas_results.csv`

context_recall	factual_correctness	faithfulness	semantic_similarity	answer_relevancy	context_precision
1.0	0.56	0.73	0.90	0.90	0.99

Pregunta

¿qué tipo de ayuda ofrece la consejería de fomento, cooperación local y prevención de incendios en asturias?

Respuesta

La Consejería de Fomento, Cooperación Local y Prevención de Incendios del Principado de Asturias ofrece varios tipos de ayudas, principalmente en forma de subvenciones y entregas dinerarias sin contraprestación. A continuación, te detallo las convocatorias más relevantes:

1. ****Ayudas para la transformación de flotas de mercancías y viajeros (achatarramiento y adquisición de vehículos)****

- ****Tipo de ayuda:**** Subvención y entrega dineraria sin contraprestación.
- ****Presupuesto total:**** 315,000 euros (achatarramiento) y 777,332.7 euros (adquisición de vehículos).
- ****Tipos de beneficiarios:**** PYMEs y personas físicas que desarrollan actividad económica.
- ****Sectores involucrados:**** Transporte y almacenamiento.
- ****Finalidad:**** Subvenciones al transporte.
- ****Fecha de inicio de solicitud:**** 25 de mayo de 2022.
- ****Fecha de fin de solicitud:**** 30 de abril de 2024.
- ****Enlace a la convocatoria de achatarramiento:**** [Convocatoria 806554](https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/806554)
- ****Enlace a la convocatoria de adquisición de vehículos:**** [Convocatoria 806556](https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/806556)

2. ****Subvenciones a corporaciones locales en el área de influencia socioeconómica del Parque Nacional de los Picos de Europa****

- ****Tipo de ayuda:**** Subvención y entrega dineraria sin contraprestación.
- ****Presupuesto total:**** 248,000 euros.
- ****Tipos de beneficiarios:**** Personas jurídicas que no desarrollan actividad económica.
- ****Sectores involucrados:**** Agricultura, ganadería, caza y servicios relacionados.

- **Finalidad:** Agricultura, pesca y alimentación.
- **Fecha de inicio de solicitud:** No especificada.
- **Fecha de fin de solicitud:** No especificada.
- **Enlace a la convocatoria:** [Convocatoria 806684](https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/806684)

Estas ayudas son parte de las iniciativas para mejorar y transformar el sector del transporte y apoyar a las comunidades locales en Asturias. Si necesitas más información sobre alguna convocatoria específica, no dudes en preguntar.

- **Ejemplo satisfactorio 2: (Generada con RAGAS)**

Fichero resultados: ragas_eval/results/2025_01_03_19_45_10_ragas_results.csv

context_recall	factual_correctness	faithfulness	semantic_similarity	answer_relevancy	context_precision
1.0	0.5	1.0	0.97	0.83	0.99

Pregunta

¿qué tipo de ayuda ofrece la secretaría general de cultura en extremadura?

Respuesta

La Secretaría General de Cultura en Extremadura ofrece varias ayudas en forma de subvenciones y entregas dinerarias sin contraprestación, bajo un régimen de concurrencia competitiva. Aquí te presento los detalles de las convocatorias disponibles:

1. **Ayudas a la programación de artes escénicas y de música (Primer Procedimiento)**
 - **Presupuesto Total:** 1,300,000 euros
 - **Descripción:** Subvenciones a la programación de artes escénicas y de música en la red de teatros y otros espacios escénicos de Extremadura.
 - **Tipos de beneficiarios:** Personas jurídicas que no desarrollan actividad económica.
 - **Finalidad:** Cultura.
 - **Bases Reguladoras:** Decreto 152/2024, de 12 de diciembre.
 - **Enlace a la Convocatoria:** [Convocatoria 806525](https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/806525)
2. **Ayudas a la programación de artes escénicas y de música (Segundo Procedimiento)**
 - **Presupuesto Total:** 300,000 euros
 - **Descripción:** Segunda convocatoria para la programación de artes escénicas y de música en la red de teatros y otros espacios escénicos de Extremadura.
 - **Tipos de beneficiarios:** Personas jurídicas que no desarrollan actividad económica.
 - **Finalidad:** Cultura.
 - **Bases Reguladoras:** Decreto 152/2024, de 12 de diciembre.
 - **Enlace a la Convocatoria:** [Convocatoria 806526](https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/806526)

3. **Ayudas a entidades locales menores de dos mil habitantes para la contratación de espectáculos**

- **Presupuesto Total:** 407,300 euros
- **Descripción:** Subvenciones para la contratación de espectáculos a representar en los escenarios móviles de la Junta de Extremadura.
- **Tipos de beneficiarios:** Personas jurídicas que no desarrollan actividad económica.
- **Finalidad:** Cultura.
- **Bases Reguladoras:** Decreto 63/2016, de 17 de mayo.
- **Enlace a la Convocatoria:** [Convocatoria 805018](https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/805018)

Si necesitas más información sobre alguna de estas ayudas o el proceso de solicitud, no dudes en preguntar.

- **Ejemplo satisfactorio 3: (Generada sintéticamente)**

Fichero resultados: ragas_eval/results/2025_01_03_17_25_42_ragas_results.csv

faithfulness	semantic_similarity	answer_relevancy	context_precision
0,77	0.71	0.94	0.99

Pregunta

¿hay subvenciones para el sector salud?

Respuesta

Sí, hay varias subvenciones dirigidas al sector salud. A continuación, te detallo algunas que podrían ser de interés:

1. **Subvenciones para asociaciones sin ánimo de lucro en Cantabria**

- **Descripción:** Convocatoria de subvenciones para asociaciones que realicen actividades de interés sanitario en el ámbito de la atención sanitaria en Cantabria durante 2025.
- **Presupuesto total:** 70,000 euros.
- **Tipo de beneficiarios:** Personas jurídicas que no desarrollan actividad económica.
- **[Enlace a la convocatoria]**(https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/804115)

2. **Subvenciones a entes locales para obras o helisuperficies de centros sanitarios en Extremadura**

- **Descripción:** Subvenciones 2025 para la realización de obras o helisuperficies en centros sanitarios de Extremadura.
- **Presupuesto total:** 889,290 euros.
- **Tipo de beneficiarios:** Personas jurídicas que no desarrollan actividad económica.

- **[Enlace a la convocatoria]**(<https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/805928>)******

3. **Apoyo a programas de promoción de la salud en Bilbao**

- **Descripción**: Convocatoria para el apoyo a programas destinados a la promoción de la salud.

- **Presupuesto total**: 92,000 euros.

- **Tipo de beneficiarios**: Personas jurídicas que no desarrollan actividad económica.

- **[Enlace a la convocatoria]**(<https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/803984>)******

Estas ayudas están en diferentes fases de convocatoria y tienen distintos objetivos dentro del ámbito de la salud. Te recomiendo revisar cada enlace para obtener información más detallada sobre los requisitos y procesos de solicitud.

Ejemplos NO Satisfactorios en la evaluación

- Ejemplo NO satisfactorio 1: (Generada sintéticamente)**

Fichero resultados: `ragas_eval/results/2025_01_03_17_25_42_ragas_results.csv`

faithfulness	semantic_similarity	answer_relevancy	context_precision
0,6	0.71	0	0

Pregunta

¿hay subvenciones para el sector industria manufacturera?

Respuesta

Actualmente, no he encontrado convocatorias específicas dirigidas de manera explícita al sector de la industria manufacturera. Sin embargo, hay algunas subvenciones que podrían ser de interés si se relacionan con actividades de investigación, desarrollo o innovación, como es el caso de la siguiente:

- Convocatoria de ayudas para proyectos de I+D sobre almacenamiento energético**:
 - Convocante**: Extremadura - Secretaría General de Ciencia, Tecnología e Innovación
 - Presupuesto total**: 3.000.000 euros
 - Descripción**: Ayudas 2024 a empresas para proyectos de I+D sobre almacenamiento energético en régimen de colaboración público-privada.
 - Tipos de beneficiarios**: Gran empresa, PYME y personas físicas que desarrollan actividad económica.
 - Región de impacto**: ES43 - Extremadura
 - Finalidad**: Investigación, desarrollo e innovación.

- ****Enlace a la convocatoria****: [Ayudas I+D

Extremadura](<https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/805281>)

Si tu interés está más orientado hacia otro tipo de subvenciones o si deseas información específica sobre algún proyecto dentro del sector de la industria manufacturera, por favor házmelo saber.

➔ Explicación: En este caso, al no encontrar resultados, el contexto que se contesta es sobre algo relacionado y no tanto con la pregunta realizada.

- **Ejemplo NO satisfactorio 2: (Generada con RAGAS)**

Fichero resultados: `ragas_eval/results/2025_01_03_19_45_10_ragas_results.csv`

Resultado obtenido en evaluación

context_recall	factual_correctness	faithfulness	semantic_similarity	answer_relevancy	context_precision
0,33	0	0,86	0.92	0	0

Pregunta

¿qué tipo de ayuda se ofrece en la convocatoria del ayuntamiento de Puebla de Alfindén?

Respuesta

Lamentablemente, no tengo información sobre una convocatoria de ayudas específica del ayuntamiento de Puebla de Alfindén. Sin embargo, puedo ofrecerte detalles sobre otras convocatorias disponibles, como las que se mencionan a continuación:

1. ****Convocatoria de Ayuda en Poble de Vallbona****:

- ****Tipo de Ayuda****: Subvención y entrega dineraria sin contraprestación.

- ****Descripción****: Concesión de prestaciones económicas individuales municipales a personas en situación de emergencia social (PEIM), ejercicio 2025.

- ****Presupuesto Total****: 15,000 euros.

- ****Tipos de Beneficiarios****: Personas físicas que no desarrollan actividad económica.

- ****Finalidad****: Servicios sociales y promoción social.

- ****Enlace a la convocatoria****: [Convocatoria Poble de Vallbona](<https://www.pap.hacienda.gob.es/bdnstrans/ge/es/convocatorias/804627>)

Si necesitas información sobre una convocatoria específica, no dudes en preguntar o especificar el área de interés.

➔ Explicación: En este caso el contexto proporcionado no devuelve la información correcta proporcionando el LLM una respuesta de lo que el entiende que es algo similar.