TRABAJOS FIN DE GRADO

Universidad de Jaén

Escuela Politécnica Superior de Jaén

Prototipo de Sistema de Acceso a Información Oficial de la Comunidad Autónoma de Andalucía con Técnicas de PLN

Autor: Juan Fernando Brocal Mora

Grado: Ingeniería en Informática

Directores: Prof. D. Eugenio Martínez Cámara y Prof. D. Luis Alfonso Ureña López
Departamento del director: Departamento de Informática

Fecha: 19/2/2025

UNIVERSIDAD DE JAÉN

D./Dª Prof. D. Eugenio Martínez Cámara y D./Dª Prof. D. Luis Alfonso Ureña López, tutor(es) del Trabajo Fin de Grado titulado: **Prototipo de Sistema de Acceso a Información Oficial de la Comunidad Autónoma de Andalucía con Técnicas de PLN**, que presenta Juan Fernando Brocal Mora, autoriza(n) su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, 19/2/2025

El estudiante

Los tutores

BROCAL MORA JUAN FERNANDO - 75100976L
Firmado digitalmente por BROCAL MORA JUAN FERNANDO - 75100976L
Fecha: 2025.02.19 20:40:42 +01'00'

Juan Fernando Brocal Mora

Firmado digitalmente por MARTINEZ CAMARA EUGENIO - 77355503V
Fecha: 2025.02.19 20:05:44 +01'00'

Prof. D. Eugenio Martínez Cámara

UREÑA LOPEZ LUIS ALFONSO - 52552054K
Firmado digitalmente por UREÑA LOPEZ LUIS ALFONSO - 52552054K
Fecha: 2025.02.19 20:11:25 +01'00'

Prof. D. Luis Alfonso Ureña López

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de este trabajo. Este proyecto ha sido un desafío intelectual y personal, y no habría sido posible sin el apoyo y la guía de quienes han estado a mi lado a lo largo del proceso.

En primer lugar, quiero agradecer a mis tutores, los profesores Eugenio Martínez Cámara y Luis Alfonso Ureña López, por brindarme la oportunidad de plasmar mis conocimientos en este proyecto. Su orientación, apoyo y dedicación han sido fundamentales para el desarrollo de esta investigación, guiándome en cada etapa con su experiencia y sabiduría. Gracias a su paciencia y consejos, he podido enfrentar cada reto con mayor seguridad y confianza.

También deseo expresar mi gratitud a cada uno de los compañeros del grupo SINAI, cuyo apoyo y colaboración han sido esenciales a lo largo de este proceso. Vuestras aportaciones y debates han enriquecido mi aprendizaje, permitiéndome profundizar en aspectos clave del proyecto y mejorar constantemente. Compartir este camino con vosotros ha hecho que el esfuerzo valga aún más la pena.

A mi familia, y en especial a mi madre, por su amor incondicional, paciencia y constante ánimo. Sin su apoyo inquebrantable, este logro no habría sido posible. Su confianza en mí ha sido una fuente de motivación constante, recordándome la importancia del esfuerzo y la perseverancia.

Asimismo, quiero agradecer a mi pareja por su comprensión, motivación y por acompañarme y ayudarme en cada paso de este recorrido académico. Su apoyo ha sido un pilar fundamental, brindándome la estabilidad y el ánimo necesarios para superar los momentos de dificultad. Su paciencia y aliento han sido clave para mantenerme enfocado en mis objetivos.

A mi hija, por su amor incondicional y por comprender mi situación cuando no podía dedicarle más tiempo. Su paciencia y cariño han sido una fuente de inspiración constante, dándome fuerzas para seguir adelante y terminar este trabajo con el mejor esfuerzo posible.

A todos vosotros, gracias por vuestro respaldo, vuestra confianza y por formar parte de este viaje. Este logro es también vuestro.

Tabla de contenidos

Lista de figuras

Lista de tablas

Lista de algoritmos

Capítulo 1

INTRODUCCIÓN

En la actualidad, la transformación digital ha dado lugar a una creciente necesidad de gestionar el conocimiento de manera eficiente para impulsar la investigación y el progreso tecnológico. En este escenario, las herramientas basadas en inteligencia artificial han cobrado protagonismo al ofrecer soluciones para automatizar y optimizar procesos complejos. Entre estas tecnologías, el Procesamiento del Lenguaje Natural (PLN) destaca por su capacidad para analizar, organizar y extraer información relevante de grandes conjuntos de datos, incrementando notablemente su accesibilidad y utilidad.

En este marco, la automatización de la extracción de cualquier cuestión relacionada con documentos de cualquier índole (ya sea de dominio público como privado) se perfila como una herramienta esencial para garantizar que el conocimiento sea fácilmente accesible con tan solo pedir la información necesaria escribiendo qué se necesita conocer. Esta solución no solo beneficia a profesionales que requieren información precisa y específica en su labor diaria, sino que también resulta invaluable para cualquier persona que quiere y necesita conocer y comprender de una manera cercana ese conocimiento.

El presente proyecto se fundamenta en los avances más recientes del PLN, integrando metodologías avanzadas de extracción de información. Además, hace uso de grandes modelos de lenguaje (LLM) combinados con un enfoque innovador de Generación Aumentada por Recuperación, con el objetivo de obtener respuestas exactas y contextualizadas que respondan a las necesidades de los usuarios, además de una evaluación de dichos modelos.

1.1. Planteamiento del problema

En las últimas décadas, la gestión y el análisis de datos han ganado relevancia, especialmente en el ámbito de las administraciones públicas. Este Trabajo de Fin de Grado tiene como objetivo principal el diseño e implementación de un sistema de Recuperación y Generación Aumentada (RAG) basado en los presupuestos de la Junta de Andalucía de los últimos 20 años. Este sistema permitirá extraer y sintetizar información relevante de manera eficiente, ofreciendo una herramienta innovadora y funcional para el análisis de esta información permitiendo comprender y acceder dentro de grandes volúmenes de datos económicos, financieros y estructurales.

1.2. Objetivos

El propósito de este proyecto es el diseño e implementación del acceso de la información de todo lo relacionado con los presupuestos de la Junta de Andalucía.

Así pues, los objetivos específicos de este trabajo son los siguientes:

- **Mejorar la accesibilidad y estructuración de la información presupuestaria:** Dado el volumen masivo de datos en las leyes de presupuestos, este trabajo busca desarrollar un sistema que **facilite la localización, comprensión y uso de la información**, garantizando un acceso rápido y preciso a los datos relevantes. La información presupuestaria no siempre se presenta de manera estructurada o intuitiva, dificultando su interpretación y análisis. Para superar estas limitaciones, se implementará una solución que no solo recupere datos de manera eficiente, sino que también ofrezca referencias claras a las fuentes originales, mejorando la transparencia y fiabilidad del sistema.
- **Recolectar y estructurar datos presupuestarios:** Obtener y organizar los datos históricos de los presupuestos de la Junta de Andalucía, asegurando su calidad y estructura para su procesamiento eficiente.
- **Optimizar el acceso a la información:** Diseñar una interfaz que permita a los usuarios navegar de forma intuitiva y eficiente por los datos presupuestarios, asegurando su fácil comprensión y recuperación.
- **Desarrollar una aplicación web robusta basada en buenas prácticas y patrones de diseño:** Garantizar que la aplicación sea modular, eficiente y fácil de

mantener, facilitando futuras ampliaciones o adaptaciones a nuevos requerimientos.

- **Evaluar el desempeño de modelos de lenguaje en el análisis presupuestario:** Realizar pruebas exhaustivas utilizando preguntas de referencia (*"source of truth"*) para medir la precisión y relevancia de las respuestas generadas por diferentes modelos de lenguaje.
- **Adquirir conocimientos avanzados en Procesamiento del Lenguaje Natural (PLN):** Fortalecer competencias en el diseño, desarrollo y evaluación de sistemas basados en inteligencia artificial, con el objetivo de enfrentar y resolver desafíos técnicos complejos en este campo.

1.3. Hipótesis principal

La hipótesis central de este trabajo sostiene que la implementación de un **sistema de Recuperación y Generación Aumentada (RAG)** puede transformar significativamente el análisis y la interpretación de los **presupuestos generales de cualquier comunidad autónoma, como la de Andalucía**. Mediante la integración de técnicas avanzadas de recuperación de información y modelos de lenguaje de última generación (*Large Language Models*, LLMs), el sistema permitirá **extraer, procesar y presentar información sobre las leyes del presupuesto de manera más rápida, precisa y adaptada a las necesidades específicas de los usuarios**.

Para ello, se plantea el desarrollo de una metodología que **combine la recuperación de documentos relevantes con la generación automática de respuestas fundamentadas en datos oficiales**. Este enfoque no solo facilitará la comprensión y el análisis de la información financiera y administrativa, sino que también optimizará la accesibilidad a datos complejos, garantizando una interpretación clara y contextualizada.

1.4. Estructura de la memoria

La estructura de la siguiente memoria está organizada de la siguiente manera, cumpliendo las típicas fases del ciclo de vida del desarrollo de software (SDLC):

- **Capítulo 2. Antecedentes:** Proporciona un marco teórico que incluye el estado

del arte en PLN, destacando las tecnologías clave y los avances más significativos.

- **Capítulo 3. Planificación:** Esta fase incluye tareas como análisis de costos y beneficios, programación, estimación de recursos y asignación. También aparece un plan detallado para conseguir los objetivos. Todos los requisitos son plasmados en un documento de especificaciones con los requisitos del software que establece las especificaciones y define los objetivos comunes que ayudan a planificar el proyecto.
- **Capítulo 4. Análisis.** En este capítulo se identifican y describen los requisitos del sistema, además de presentar diagramas que explican las interacciones y los casos de uso propuestos.
- **Capítulo 5. Diseño:** Se expone la arquitectura del sistema, el diseño de los datos y las salidas esperadas, así como las tecnologías que respaldan su implementación.
- **Capítulo 6. Implementación:** Detalla el desarrollo práctico del sistema, abarcando el procesamiento de datos, la extracción de la información, la creación de la respuesta a partir de preguntas con la metodología RAG, junto con la evaluación de los modelos empleados.
- **Capítulo 7. Resultados:** Detalla los resultados de evaluación y los motivos que dan lugar a esos resultados.
- **Capítulo 8. Conclusiones:** Se sintetizan los hallazgos del proyecto, se examinan sus limitaciones y se plantean posibles líneas de investigación futura.

Esta estructura está diseñada para garantizar una comprensión integral del proyecto, desde sus fundamentos teóricos hasta su implementación y evaluación final.

Capítulo 2

ANTECEDENTES

2.1. Planteamiento del problema

Este capítulo se dedica a ofrecer un análisis exhaustivo del estado actual del Procesamiento del Lenguaje Natural (PLN) y las tecnologías relacionadas, estableciendo un marco teórico esencial para este proyecto. Se aborda la evolución histórica de esta disciplina, destacando los avances en herramientas y métodos que han permitido a los sistemas computacionales interpretar, procesar y generar lenguaje humano de manera cada vez más sofisticada.

También se examinan los Grandes Modelos de Lenguaje (LLM), explorando su capacidad para generar y comprender texto con un nivel de similitud al humano. Se resalta su impacto en diversas áreas, desde la automatización de tareas de generación de contenido hasta la mejora en la accesibilidad y el entendimiento de información compleja.

En este contexto, se presenta la metodología principal empleada en el proyecto: la Generación Aumentada por Recuperación (RAG). Este enfoque combina técnicas avanzadas de recuperación de datos relevantes con modelos generativos, incrementando la precisión y el valor contextual de las respuestas a partir de preguntas generadas. Además, se destacan las bases de datos vectoriales como pilares clave para organizar y recuperar información semántica, aprovechando algoritmos avanzados de aprendizaje automático para mejorar la eficiencia en el manejo de grandes volúmenes de datos.

Por último, se evalúa la tarea de extracción de información procedente del histórico de presupuestos y leyes de la Junta de Andalucía. Esta exploración permite al

lector familiarizarse con los conceptos clave que serán profundizados en los capítulos siguientes.

2.2. Estado del arte

El estado del arte [1] se refiere al análisis y síntesis del conocimiento más avanzado y actual dentro de un área de estudio específica. En el contexto del Trabajo de Fin de Grado (TFG), el estado del arte tiene como objetivo:

- **Revisar trabajos previos:** Explorar investigaciones, proyectos y avances relevantes en el ámbito del TFG.
- **Identificar lagunas:** Detectar problemas no resueltos o áreas de oportunidad que pueden ser abordadas en el trabajo.
- **Contextualizar el proyecto:** Proporcionar un marco teórico que fundamente el desarrollo del trabajo y lo vincule con investigaciones previas. Un buen estado del arte no solo describe las investigaciones existentes, sino que las compara, contrasta y las relaciona con los objetivos del proyecto.

Por lo que se llevará a cabo un análisis exhaustivo del estado del arte en relación con los conceptos fundamentales, con el propósito de proporcionar un marco de referencia sólido que facilite la comprensión y contextualización de los contenidos desarrollados en este estudio.

2.2.1. Procesamiento del Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural (PLN) es una rama de la inteligencia artificial que se centra en el estudio y desarrollo de sistemas capaces de comprender, interpretar y generar lenguaje humano. Este campo combina principios de lingüística, informática y aprendizaje automático (machine learning [2]) para crear herramientas que interactúan con el lenguaje de forma efectiva. Entre sus aplicaciones destacan la traducción automática, el análisis de sentimientos, la generación de texto y la extracción de información relevante de grandes volúmenes de datos (Hernández et al., 2014) [3] entre otras aplicaciones.

Desde sus inicios, el PLN ha evolucionado significativamente gracias al desarrollo de algoritmos más avanzados y la disponibilidad de grandes volúmenes de datos textuales. La introducción de los Grandes Modelos de Lenguaje (LLMs), como GPT y BERT, ha revolucionado la capacidad de los sistemas para generar textos coherentes y relevantes en múltiples contextos (Goldberg, 2017) [4].

No obstante, este campo enfrenta desafíos éticos relacionados con la privacidad, el sesgo en los datos y el uso responsable de estas tecnologías (Molina Montoya, 2014) [1].

2.2.2. Large Language Models (LLM)

Los Large Language Models o Grandes Modelos del Lenguaje (LLMs) son modelos avanzados de inteligencia artificial diseñados para procesar y generar lenguaje natural. Estos modelos destacan por su capacidad para comprender y generar lenguaje con un nivel de naturalidad y fluidez muy cercano al humano. Además, ofrecen funcionalidades como la clasificación, simplificación de textos y generación de contenido estructurado. Su funcionamiento se basa en el manejo de unidades de texto denominadas tokens, que pueden ser palabras, fragmentos de palabras o caracteres individuales. Estos tokens son procesados en secuencias para permitir que el modelo entienda el contexto y genere respuestas coherentes. La unidad mínima de procesamiento en los LLMs es fundamental para garantizar precisión y relevancia en las tareas de comprensión y generación de texto (Gutiérrez López y Pérez Gómez, 2019) [5].

La historia de los LLMs está estrechamente ligada al desarrollo de los **transformadores**, una arquitectura introducida por Vaswani en 2017 [6] con el famoso trabajo Attention Is All You Need. Los transformadores emplean mecanismos de autoatención, lo que les permite evaluar las relaciones entre todos los tokens de una secuencia de entrada de manera simultánea.

Con la llegada de modelos como GPT, BERT y sus sucesores, los LLMs comenzaron a entrenarse en bases de datos masivas que incluyen libros, artículos y páginas web. Este entrenamiento, realizado con billones de tokens, permitió a los modelos generar texto coherente y relevante en múltiples idiomas y contextos (Martínez y Pérez, 2020) [2].

Un ejemplo temprano de estos avances fue el modelo GPT-1 de OpenAI, seguido por GPT-2 en 2019, cuyo lanzamiento generó debates debido a su potencial para mal uso. Posteriormente, GPT-3 y GPT-4 ampliaron las capacidades de los LLMs, incluyen-

do aplicaciones multimodales y mayor precisión en tareas complejas (López García y Sánchez Rivera, 2021) [7]. En particular, GPT-4, lanzado en 2023, ha sido señalado por su versatilidad en aplicaciones multimodales y su impacto en la accesibilidad y la automatización de procesos (Martínez y Pérez, 2020) [2].

En conclusión, los LLMs han transformado la manera en que interactuamos con sistemas automatizados, habilitando aplicaciones como asistentes virtuales, traducción automática y generación de contenido personalizado. Su capacidad para manejar grandes volúmenes de datos y comprender complejidades lingüísticas los convierte en herramientas indispensables en el avance del Procesamiento del Lenguaje Natural (PLN).

2.2.3. Retrieval-Augmented Generation (RAG)

La **Generación Aumentada por Recuperación o Retrieval-Augmented Generation (RAG)** ha evolucionado a ritmo vertiginoso y en paralelo a los avances tecnológicos relacionados con los LLMs.

En su contexto actual, RAG se refiere a un enfoque que combina la recuperación dinámica de información con la generación de texto, permitiendo mejorar la precisión y relevancia de las respuestas generadas por los modelos sin necesidad de un reentrenamiento exhaustivo.

Este método es especialmente útil en tareas que requieren un uso intensivo del conocimiento, como la respuesta a preguntas basada en documentos, la generación de informes o la búsqueda semántica avanzada. Al integrar un sistema de recuperación con un modelo generativo sequence-to-sequence, sometido a un ajuste end-to-end fine-tuning, RAG optimiza la generación de respuestas y reduce problemas como las alucinaciones de los modelos (Martínez y Pérez, 2020) [2].

2.2.3.1. Componentes Principales de un Sistema RAG

Un sistema RAG está compuesto por tres elementos clave: recuperador, generador y codificador de información.

1. **Recuperador de Información:** El recuperador es un modelo de aprendizaje automático basado en redes neuronales que toma una consulta en lenguaje natural

como entrada y devuelve un conjunto de fragmentos relevantes desde una base de conocimientos. Para lograrlo, emplea métodos avanzados de recuperación semántica, tales como:

- **Búsqueda del Producto Interior Máximo (MIPS - Maximum Inner Product Search) [8]:** Identifica los fragmentos de texto más similares a la consulta con base en la similitud coseno en el espacio vectorial.
- **Búsqueda en Bases de Datos Vectoriales:** Utiliza técnicas de indexación como **Approximate Nearest Neighbors (ANN)** [9] para una búsqueda rápida y eficiente *(Véase en detenimiento en el próximo apartado).*
- **Recuperación Multi-documento:** Puede consultar diversas fuentes de información, como bases de datos empresariales, artículos científicos o repositorios documentales.

Con todo ello, el recuperador se entrena para optimizar tareas específicas, como respuesta a preguntas, generación de resúmenes o recuperación de información basada en contexto.

2. **Generador de Texto:** El generador es el modelo **sequence-to-sequence** [10] encargado de producir un texto de salida basado en los fragmentos recuperados y la consulta original. Este modelo sigue la arquitectura de **codificador-decodificador**, donde:

   - El codificador transforma la consulta y los pasajes en representaciones vectoriales en un espacio común.
   - El decodificador genera la respuesta en lenguaje natural a partir de esta representación.

   Al aprovechar la información recuperada, el generador mejora la coherencia y precisión de las respuestas, evitando errores comunes en modelos de lenguaje entrenados exclusivamente con datos estáticos.

3. **Codificador de Representaciones:** El codificador es una red neuronal que transforma la consulta en una representación numérica (embedding), capturando su significado e intención en un espacio vectorial. Su función principal es:

   - **Convertir consultas en vectores de longitud fija** para su comparación con fragmentos de texto almacenados en la base de datos.
   - **Facilitar la búsqueda semántica** al organizar la información en un espacio latente, permitiendo encontrar respuestas relevantes sin necesidad de coincidencias exactas de palabras clave.

Para esto, se pueden emplear modelos preentrenados como **AutoModelForSequenceClassification** [11], que permiten adaptar el sistema a distintas tareas como la clasificación de texto o el análisis de sentimientos.

2.2.4. Base de datos vectoriales

El almacenamiento y la organización de datos han evolucionado significativamente en las últimas décadas, especialmente en el ámbito de la recuperación de información. Tradicionalmente, las bases de datos relacionales han dominado este campo, estructurando los datos mediante modelos entidad-relación y utilizando búsquedas basadas en palabras clave. Sin embargo, con la creciente necesidad de procesar grandes volúmenes de datos no estructurados y mejorar la precisión en la búsqueda semántica, han surgido las bases de datos vectoriales como una solución más avanzada y eficiente.

Para lograrlo, emplean una representación numérica basada en vectores **(embeddings)** [12] , que permiten modelar datos en espacios de alta dimensión. Estos vectores capturan **relaciones semánticas** entre palabras, documentos, imágenes e incluso audio, lo que permite una recuperación más precisa y contextualizada de la información.

Gracias a esta capacidad para interpretar la relación semántica entre datos, las bases de datos vectoriales han encontrado aplicaciones en diversos ámbitos tecnológicos como Asistentes Virtuales y Chatbots, Búsqueda Semántica en Documentos Empresariales o Reconocimiento de Imágenes y Multimodalidad.

El proceso de funcionamiento de una base de datos vectorial sigue los siguientes pasos:

- **Conversión de datos a vectores:** Los textos, imágenes o cualquier otro tipo de dato se transforman en representaciones vectoriales utilizando modelos de machine learning.
- **Almacenamiento en una base vectorial:** Se indexan estos vectores en un espacio de representación matemática, donde pueden ser recuperados posteriormente mediante métricas de similitud.
- **Búsqueda y recuperación:** Cuando se introduce una consulta, se convierte en un vector y se compara con los vectores almacenados mediante algoritmos de búsqueda de vecinos más cercanos (Nearest Neighbors Search, NNS), como

Approximate Nearest Neighbors (ANN) o Búsqueda por Máxima Similitud en Espacio Vectorial (MIPS).

Actualmente existe distintas implementaciones de base de datos vectoriales en las que aparece un cuadro comparativo para saber cual elegir:

Base de Datos	**Características**	**Casos de Uso**
FAISS [13]	Optimizada para búsqueda rápida en grandes volúmenes de datos. Usa HNSW, IVF, PQ.	Búsqueda en grandes volúmenes de datos, IA, PLN, recomendadores.
Pinecone [14]	Base de datos vectorial como servicio (SaaS), escalable y sin gestión de infraestructura.	Producción escalable de búsquedas vectoriales, SaaS.
ChromaDB [15]	Orientada a IA y RAG, open-source, fácil de integrar con LangChain.	Aplicaciones de IA, RAG, almacenamiento en memoria y persistente.
Weaviate [16]	Open-source, escalable, compatible con OpenAI y Cohere para embeddings.	Búsqueda semántica, AI embeddings, motores de recomendación.
Milvus [17]	Optimizada para ML y Big Data, alto rendimiento y escalabilidad.	Escalabilidad en Big Data, búsqueda de alta velocidad en ML.
Qdrant [18]	Open-source, alto rendimiento, compatible con LangChain y Haystack.	IA, RAG, integración con frameworks de PLN y recuperación de información.
Annoy [19]	Desarrollado por Spotify para búsqueda rápida, eficiente pero menos preciso que Redis (Redis-Vector)	Sistemas de recomendación, búsqueda en grandes catálogos.
Redis (Redis-Vector) [20]	Módulos como Redis-Vector permiten búsquedas ANN y almacenamiento de embeddings.	Bases de datos híbridas, búsqueda rápida en memoria.
Elasticsearch (k-NN) [21]	No es nativa vectorial, pero con el plugin k-NN soporta búsqueda de vecinos más cercanos.	Indexación de texto y búsqueda vectorial dentro de Elasticsearch.
Vespa [22]	Alta escalabilidad, recomendada para búsqueda en motores y recomendaciones personalizadas.	Motores de búsqueda, sistemas de recomendación, personalización en tiempo real.

Tabla. 2.1: Comparativa de Base de Datos Vectoriales

A la vista de este análisis, se muestra una tabla de recomendación de uso según necesidades:

Necesidades	Base de Datos
Para RAG	ChromaDB, Weaviate, Qdrant, FAISS.
Para escalabilidad en producción	Pinecone, Milvus, Weaviate.
Para sistemas de recomendación	Annoy, FAISS, Vespa.
Para integraciones con bases de datos tradicionales	Redis, Elasticsearch.

Tabla. 2.2: Base de datos vectoriales - ¿Cual elegir según necesidades?

Es por ello que para el caso que nos ocupa, es una excepcional elección el uso de **Chroma**. Además de estar orientada a IA y RAG, siendo open-source y fácil de integrar con frameworks como Lang-Chain, posee la versatilidad en el manejo de datos multimodales (almacenar y recuperar embeddings no solo de texto, sino también de imágenes y, en futuras versiones, de audio y video). Por último destacar que permite realizar búsquedas semánticas eficientes sobre grandes conjuntos de datos por lo que la posiciona como una opción clave para aquellos desarrolladores que buscan optimizar la gestión de información en aplicaciones basadas en inteligencia artificial.

2.2.5. Resumen: Flujo de Trabajo en un Sistema RAG

Un vez analizado todos los conceptos necesarios del estado del arte de este trabajo, es el momento de explicar por completo el funcionamiento de un sistema RAG siguiendo un proceso estructurado en varias etapas:

1. **Indexación de la Base de Conocimiento:**
   - Se divide el corpus de texto en fragmentos y se generan representaciones vectoriales con un modelo de embeddings.
   - Estos vectores se almacenan en una base de datos vectorial, junto con su texto asociado con un índice de referencia *(Véase en el siguiente gráfico los puntos 1, 2, 3 y 4).*
2. **Procesamiento de la Consulta:** Cuando un usuario realiza una pregunta, esta se convierte en un vector de consulta mediante el mismo modelo de embeddings. Este vector ejecuta de nuevo una consulta en el índice de la base de datos vectorial *(Véase en el siguiente gráfico puntos 5 y 6).*
3. **Búsqueda Semántica:** Se ejecuta una búsqueda en la base de datos vectorial usando técnicas como MIPS o ANN, recuperando los fragmentos de texto más

relevantes dentro del espacio de latencias del embedding *(Véase en el siguiente gráfico un ejemplo con ANN puntos 7 y 8).*

4. **Generación de la Respuesta:** La información recuperada se pasa como contexto al modelo de lenguaje, y junto a la pregunta del usuario, se construye un prompt adaptado que genera la respuesta basándose únicamente en el contenido recuperado *(Véase en el siguiente gráfico punto 9).*

5. **Presentación del Resultado:** La respuesta generada se devuelve al usuario en formato de texto estructurado y mostrada en el interface del usuario *(Véase en el siguiente gráfico punto 10).*

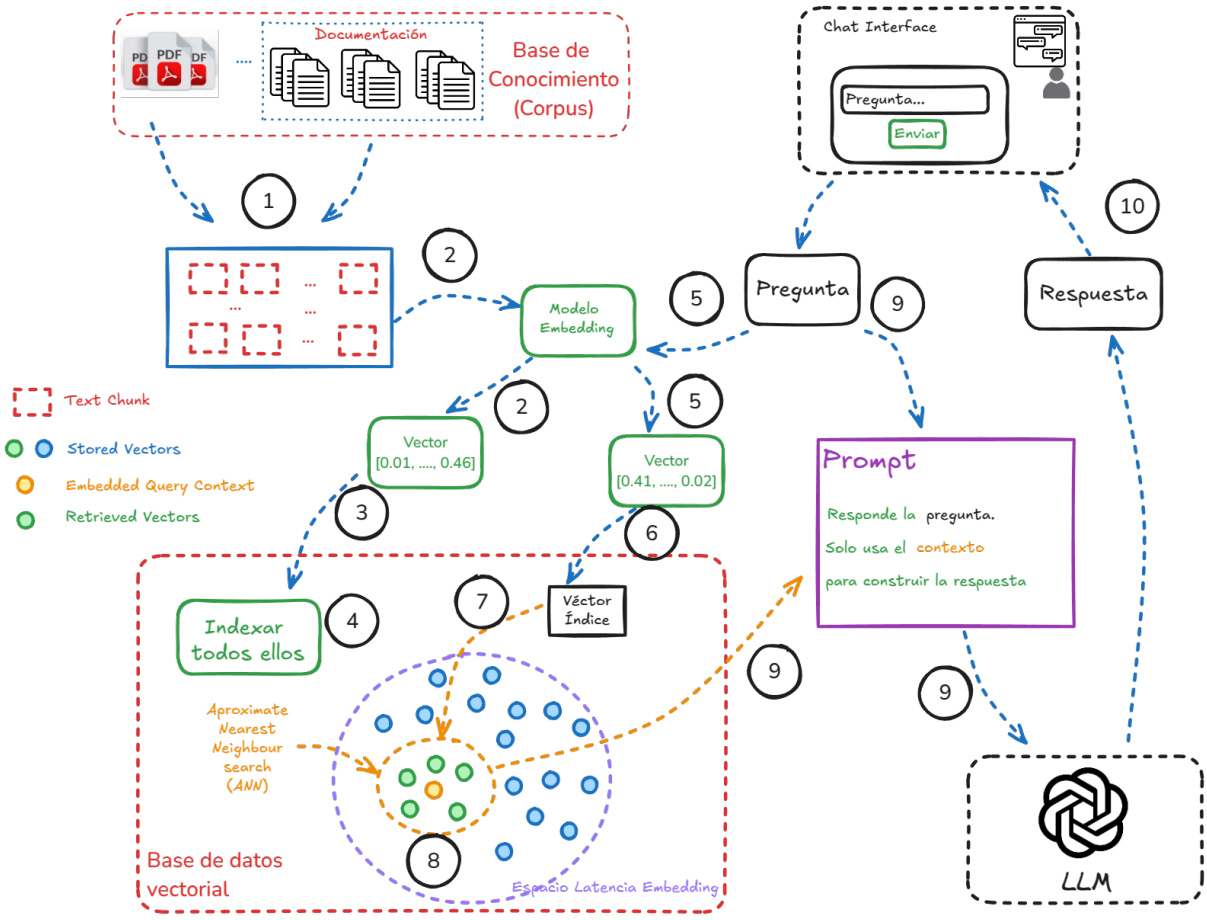


Figura 2.1: Esquema de Retrieval-Augmented Generation (RAG)

2.3. Sistema de Acceso a la Información Oficial de la Comunidad Autónoma

El diseño e implementación de un **sistema de Recuperación y Generación Aumentada (RAG)** basado en los presupuestos de la Junta de Andalucía supone un reto significativo en el ámbito del **Procesamiento del Lenguaje Natural (PLN)** y la gestión de **grandes volúmenes de datos estructurados**. La finalidad de este proyecto es desarrollar una herramienta capaz de extraer, **sintetizar y organizar información de manera eficiente,** facilitando el acceso y la interpretación de datos jurídicos, económicos y financieros de las últimas dos décadas. Además de mejorar la disponibilidad de información oficial, este sistema contribuirá a la **transparencia administrativa,** permitiendo a ciudadanos, analistas y organismos públicos acceder a los datos de forma intuitiva y contextualizada.

Uno de los principales retos radica en la **transformación de información estructurada en un formato óptimo para su recuperación semántica.** Los modelos de búsqueda basados en **embeddings vectoriales** han demostrado su eficacia en la recuperación de datos no estructurados, pero la conversión de tablas y y otras estructuras en representaciones adecuadas para modelos RAG presenta una complejidad adicional. Para abordar este desafío, es fundamental desarrollar estrategias que optimicen la generación de embeddings sin perder la estructura lógica de la información original.

Otro aspecto crítico es la **generación de respuestas a partir de datos cuantitativos, financieros y presupuestarios.** Estos presupuestos incluyen valores numéricos y relaciones complejas entre partidas presupuestarias, lo que dificulta la generación automática de respuestas precisas. Mientras que los modelos de lenguaje son altamente efectivos en la manipulación de texto, presentan **limitaciones en el razonamiento sobre datos numéricos,** lo que puede afectar la exactitud de la información generada.

Además, la **evaluación de la precisión y relevancia de las respuestas** representa otro desafío fundamental. La validación de la información recuperada es un proceso inherentemente subjetivo, especialmente en modelos no supervisados donde los resultados pueden no estar claramente definidos desde el inicio. La ausencia de un **estándar universal para la evaluación de respuestas en documentos oficiales** puede generar variabilidad en la interpretación de los resultados y dificultar la medición de la calidad del sistema. Para mitigar este problema, será necesario establecer **métricas de evaluación objetivas,** garantizar la verificabilidad de las respuestas me-

diante referencias explícitas a las fuentes de origen consideradas como “source of truth”.

Por último, la implementación del sistema deberá abordar el problema de **generación de respuestas fuera de contexto,** evitando las conocidas alucinaciones de los modelos de lenguaje. Un sistema RAG debe estar diseñado para reconocer consultas irrelevantes o fuera del dominio del conjunto de datos, evitando generar información errónea o sin respaldo documental. A esto se suman los desafíos de **seguridad y privacidad,** ya que la manipulación de información pública requiere estrictos criterios de integridad y accesibilidad para prevenir sesgos, distorsiones o malinterpretaciones de los datos extraídos. La combinación de estos enfoques permitirá el desarrollo de un **sistema robusto, confiable y escalable,** que optimice el acceso a la información presupuestaria de la **Comunidad Autónoma de Andalucía** de manera eficiente y transparente.

Capítulo 3

PLANIFICACION

La fase de planificación en el desarrollo de software es fundamental para garantizar la viabilidad y éxito del proyecto. Consiste en definir objetivos, alcance, recursos, costos, cronograma y estrategias de evaluación para alinear el trabajo con los requisitos establecidos.

Según Sommerville (2020) [23], la planificación en proyectos de software debe incluir la *gestión de riesgos, la asignación eficiente de recursos y el diseño de estrategias de implementación y validación*.

Este proyecto sigue una planificación estructurada para garantizar el desarrollo eficiente del sistema de Recuperación y Generación Aumentada (RAG), aplicado a las leyes de presupuestos de la Junta de Andalucía. Se definirán objetivos claros, entregables bien especificados, una estimación de costos detallada y un cronograma preciso para coordinar el esfuerzo dentro del período 09/09/2024 - 19/02/2025.

3.1. Alcance del Proyecto

Basado en los objetivos descritos en la sección de antecedentes y objetivos (Véase 1.2), este proyecto desarrollará un sistema basado en modelos de lenguaje LLMs y técnicas de recuperación de información para optimizar la accesibilidad a los datos presupuestarios.

3.1.1. Entregables del Proyecto

Los entregables son los productos finales o parciales que se deben entregar en cada fase del proyecto. Existen diversos tipos:

- **Documentación**: Informes de análisis, diseño, implementación y evaluación.
- **Código Fuente**: Implementación del sistema con frameworks asociados.
- **Interfaz de Usuario**: Aplicación web desarrollada con frameworks para el frontend como para el backend.
- **Pruebas y Validaciones**: Evaluación del sistema con métricas establecidas.
- **Despliegue y Manual de Usuario**: Implementación del sistema en un entorno accesible y guía de uso.

3.1.2. Tabla de Entregables

A continuación, se desglosa con más detalles estos entregables:

Entregable	Descripción	Fecha de Entrega
Informe de Planificación	Documento detallado con cronograma, costos y recursos.	30/09/2024
Análisis de Requisitos	Definición de necesidades del sistema y modelo RAG, estableciendo los requerimientos funcionales y no funcionales.	15/10/2024
Diseño del Sistema	Definición de la arquitectura, flujos de datos, selección de tecnologías y modelado de la base de datos.	01/11/2024
Implementación Backend	Desarrollo de la API bajo framework, integración con modelos de lenguaje (LLMs) y conexión con la base de datos.	30/11/2024
Implementación Frontend	Desarrollo de la interfaz gráfica bajo framework, integrando las funcionalidades de consulta y evaluación de modelos.	20/12/2024
Pruebas y Evaluación	Validación del sistema mediante pruebas unitarias, funcionales y de rendimiento, asegurando la correcta recuperación y generación de respuestas.	30/01/2025
Informe Final y Defensa	Redacción del documento final del TFG, incluyendo análisis de resultados, discusión y conclusiones, además de la preparación para la defensa.	19/02/2025

Tabla. 3.1: Entregables del Proyecto y Fechas de Entrega

3.2. Recursos del Proyecto

En la gestión de proyectos, los recursos representan todos los elementos esenciales necesarios para llevar a cabo las actividades planificadas de manera eficiente y efectiva. Estos recursos pueden ser humanos, materiales, tecnológicos y financieros, y su correcta gestión es clave para el éxito del proyecto. A continuación, se detallan los principales tipos de recursos involucrados en un proyecto de desarrollo de software:

3.2.1. Recursos Humanos

Los recursos humanos comprenden a todas las personas involucradas en el desarrollo del proyecto, cada una con roles y responsabilidades específicas. En proyectos

de software, estos pueden incluir desarrolladores, analistas, gestores de proyectos, diseñadores, QA y supervisores académicos. Una adecuada planificación y asignación de responsabilidades dentro del equipo es crucial para optimizar el tiempo y la productividad. En nuestro proyecto, veámos con mas detalle estos recursos:

Rol	**Responsabilidades**
Estudiante (Project Manager y Developer)	Planificación, desarrollo del sistema, documentación, pruebas y presentación final del proyecto.
Tutor Académico	Supervisión técnica y metodológica del proyecto, asegurando el cumplimiento de los objetivos académicos y de investigación.
Asesores Técnicos (Grupo SINAI)	Apoyo en la validación del modelo RAG, optimización de consultas y soporte en pruebas del sistema, proporcionando retroalimentación técnica.
QA del TFG (Estudiante)	Revisión final de la documentación, evaluación del trabajo presentado y validación de la defensa del TFG.

Tabla. 3.2: Roles y Responsabilidades en el Proyecto

3.2.2. Recursos Materiales y Tecnológicos

Los recursos materiales incluyen los equipos físicos necesarios para la implementación del proyecto. En el ámbito del software, esto suele referirse a hardware como computadoras de alto rendimiento, servidores, dispositivos de almacenamiento y periféricos necesarios para el desarrollo, pruebas y despliegue del sistema. Estos incluyen en el proyecto:

- **Software de Desarrollo**: Lenguajes de programación (Python), frameworks (se puede valorar el uso de los frameworks Flask, Streamlit) y bibliotecas especializadas (para la evaluación de resultados del modelo).
- **Plataformas y Servicios**: Servicios en la nube para almacenamiento y procesamiento de datos (Hugging Face, Vast.ai, AWS o Google Cloud).
- **Bases de Datos**: Motores de almacenamiento vectorial (A valorar Chromadb en sucesivas fases del proyecto).

- **Herramientas de Gestión**: Plataformas como GitHub para el control de versiones y Trello o Jira para la organización de tareas.

3.2.3. Recursos de Infraestructura

En algunos proyectos, especialmente aquellos que requieren una fase de despliegue, es fundamental contar con infraestructura que soporte el sistema en producción. Esto puede incluir servidores físicos o virtuales, configuraciones de redes, entornos de pruebas y medidas de seguridad informática. En nuestro caso, no vamos a desplegar en producción este sistema. Se implementará solo a nivel de pruebas y evaluación del mismo.

3.2.4. Estimación de recursos y costos

En concreto a nuestro proyecto, se prevee un costo total de **5.980€**, con la siguiente estimación de planificación:

- **Horas de dedicación:** 20 horas semanales desde el **9 de septiembre de 2024** hasta el **19 de febrero de 2025**.

- **Total de horas estimadas:** 460 horas.

A continuación se muestra una tabla detalle con todos los costos

Categoría	Costo Unitario	Unidad y Cantidad Utilizada	Costo Total
4.1. Costos de Infraestructura			
Servidor en la nube (Vast.ai)	300€/mes	1 servidor durante medio mes **[1]**	150€
Licencias de software (API OpenAI)	0€/mes	-	0€
Servicios en la nube (Hugging Face, API y otros servicios de IA)	0€	Uso de cuentas gratuitas	0€
Herramientas colaborativas (Trello para Desarrollo Ágil) **[2]**	10€/mes/cuenta	2 cuentas Premium 5 meses	100€
Subtotal Infraestructura			**250€**
4.2. Costos de Desarrollo			
Horas de trabajo (100h) Backend	20€/h costo neto **[3]** 1 programador	100 horas estimadas	2.000€
Horas de trabajo (80h) Frontend	20€/h costo neto 1 programador	80 horas estimadas	1.600€
Hardware (PC con GPU RTX 3060 de 8GB o ADA)	0€	Equipo personal o ADA	0€
Licencias de software (Visual Code)	0€/mes	Entorno libre	0€
Licencias de software (FrontEnd)	0€/mes	Entorno libre	0€
Licencias de software (Backend)	0€/mes	Entorno libre	0€
Subtotal Desarrollo			**3.600€**
4.3. Costos de Evaluación			
Pruebas de usuarios (Simulación con 5 usuarios)	-	5 pruebas simuladas	0€
Pruebas de modelos (Consumo de API OpenAI)	10€/mes	1 pago por uso	10€
Uso de servidores Vast.ai (GPU dedicada RTX 4090)	300€/mes	1 servidor durante 1 semana **[4]**	100€
Horas de trabajo (30h) Backend - Ajustes	20€/h costo neto 1 programador	30 horas estimadas	600€
Horas de trabajo (20h) Frontend - Ajustes	20€/h costo neto 1 programador	20 horas estimadas	400€
Horas de trabajo (50h) QA	20€/h costo neto 1 QA **[5]**	50 horas estimadas	1.000€
Subtotal Evaluación			**2.110€**
4.4. Costos de Documentación			
Impresión y encuadernación del TFG	20€	1 ejemplar impreso	20€
Subtotal Documentación			**20€**
4.5. Resto de Costos			
Resto Horas de trabajo (180h)	20€/h costo neto 1 Analista/Project Manager	180 horas estimadas	3.600€
TOTAL ESTIMADO			**5.980€**

Tabla. 3.3: Estimación Detallada de Costos del Proyecto

3.2.4.1. Justificación Estimación de recursos y costos

- **[1]**: El tiempo necesario para probar los modelos finales que necesitaban más recursos. En la implementación se optó por un modelo 2b que sí podía funcionar en la gráfica NVIDIA de 8GB del desarrollador y que no suponía coste.

- **[2]**: Necesario para poder crear tareas y un dashboard para utilizar metodología de Desarrollo Ágil.

- **[3]**: Costo neto por hora, después de impuestos. Se ha valorado el mismo costo para todos los roles que intervienen en el proyecto.

- **[4]**:El tiempo necesario para evaluar los modelos finales ya que necesitaban más recursos.

- **[5]**: Costo neto por hora, después de impuestos para este rol QA tester. Se ha valorado el mismo costo para todos los roles que intervienen en el proyecto.

3.3. Planificación Temporal

3.3.1. Introducción General

La planificación temporal en el desarrollo de software es un proceso esencial que permite organizar y gestionar las actividades necesarias para completar un proyecto dentro de un marco de tiempo definido. Su propósito es garantizar que los recursos se utilicen de manera eficiente, minimizando retrasos y asegurando que cada fase del desarrollo se lleve a cabo en el orden correcto. Para ello, se emplean herramientas como diagramas de Gantt, metodologías ágiles y marcos de trabajo que faciliten la ejecución de tareas de manera estructurada.

Este proceso abarca desde la fase de planificación y definición de objetivos hasta la entrega final del producto. Un cronograma bien definido establece plazos claros para cada tarea, permitiendo la identificación de dependencias entre actividades y la gestión de posibles riesgos. Además, facilita la supervisión del progreso y la toma de decisiones oportunas en caso de desviaciones en el calendario inicial.

3.3.2. Aplicación en el Sistema

Para la implementación del sistema de **Recuperación y Generación Aumentada (RAG)** aplicado al análisis de presupuestos, se ha diseñado un **cronograma detallado** que abarca todas las fases del desarrollo, desde la planificación hasta la entrega final. El período total de trabajo está comprendido entre el **9 de septiembre de 2024 y el 19 de febrero de 2025**, estructurando el proyecto en seis grandes fases bien definidas:

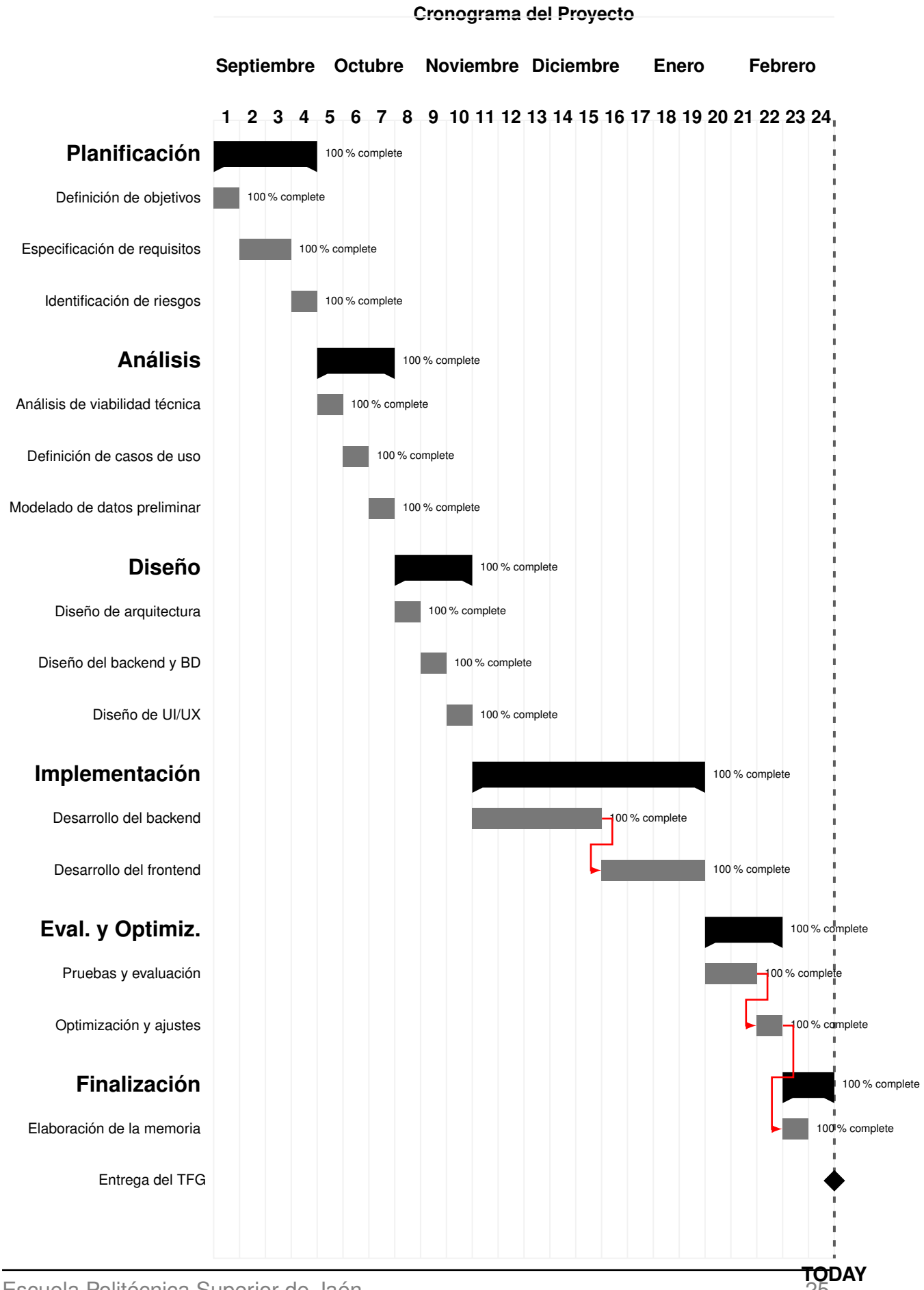
- **Fase de Planificación:** Se establecen los objetivos generales del proyecto, la especificación de requisitos del sistema y la identificación de riesgos. Esta fase es clave para garantizar una base sólida antes de proceder con el desarrollo.
- **Fase de Análisis:** Se analiza la viabilidad técnica, se definen los casos de uso y se realiza el modelado preliminar de los datos, asegurando que el sistema pueda cumplir con los requisitos establecidos.
- **Fase de Diseño:** Se define la arquitectura del sistema, la estructura del backend y frontend, y el diseño de la interfaz de usuario. Además, se modelan las bases de datos y se seleccionan las tecnologías clave para su implementación.

- **Fase de Implementación:** Se lleva a cabo el desarrollo del backend, asegurando su integración con la base de datos y los modelos de lenguaje, y posteriormente se implementa el frontend, garantizando una interfaz funcional y accesible para el usuario.
- **Fase de Evaluación y Optimización:** Se realizan pruebas exhaustivas para medir la precisión del sistema y la calidad de las respuestas generadas, optimizando el modelo y ajustando parámetros para mejorar su rendimiento.
- **Fase de Finalización:** Se documenta el proyecto en su totalidad, se prepara la defensa del *Trabajo de Fin de Grado (TFG)* y se realiza la entrega final del sistema, asegurando el cumplimiento de los requisitos establecidos.

Cada una de estas fases ha sido planificada con objetivos y tiempos bien definidos, utilizando herramientas de planificación como **diagramas de Gantt** para visualizar el avance del proyecto y detectar posibles desviaciones en el calendario. Este enfoque permite optimizar la gestión del tiempo y garantizar que el sistema cumpla con los plazos establecidos, asegurando un desarrollo estructurado y eficiente. En la **Tabla 3.4** se detalla la distribución de actividades y sus respectivas fechas de inicio y finalización, estableciendo así una visión clara del cronograma del proyecto.

Fase	Inicio	Fin	Duración
Fase de Planificación			
Definición de objetivos	09/09/2024	15/09/2024	1 semana
Especificación de requisitos	16/09/2024	29/09/2024	2 semanas
Identificación de riesgos	30/09/2024	06/10/2024	1 semana
Total Planificación	09/09/2024	06/10/2024	4 semanas
Fase de Análisis			
Análisis de viabilidad técnica	07/10/2024	13/10/2024	1 semana
Definición de casos de uso	14/10/2024	20/10/2024	1 semana
Modelado de datos preliminar	21/10/2024	27/10/2024	1 semana
Total Análisis	07/10/2024	27/10/2024	3 semanas
Fase de Diseño			
Diseño de la arquitectura del sistema	28/10/2024	03/11/2024	1 semana
Diseño del backend y base de datos	04/11/2024	10/11/2024	1 semana
Diseño de la interfaz gráfica (UI/UX)	11/11/2024	17/11/2024	1 semana
Total Diseño	28/10/2024	17/11/2024	3 semanas
Fase de Implementación			
Desarrollo del backend	07/10/2024	10/11/2024	5 semanas
Desarrollo del frontend	11/11/2024	08/12/2024	4 semanas
Total Implementación	07/10/2024	08/12/2024	9 semanas
Fase de Evaluación y Optimización			
Pruebas y evaluación	09/12/2024	29/12/2024	3 semanas
Optimización y ajustes	30/12/2024	12/01/2025	2 semanas
Total Evaluación y Optimización	09/12/2024	12/01/2025	5 semanas
Fase de Finalización			
Elaboración de la memoria	13/01/2025	09/02/2025	4 semanas
Entrega y defensa del TFG	10/02/2025	19/02/2025	1 semana
Total Finalización	13/01/2025	19/02/2025	5 semanas

Tabla. 3.4: Cronograma del Proyecto (9 de septiembre 2024 - 19 de febrero 2025)


Cronograma del Proyecto
Septiembre
Octubre
Noviembre
Diciembre
Enero
Febrero
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Planificación
Definición de objetivos
Especificación de requisitos
Identificación de riesgos
Análisis
Análisis de viabilidad técnica
Definición de casos de uso
Modelado de datos preliminar
Diseño
Diseño de arquitectura
Diseño del backend y BD
Diseño de UI/UX
Implementación
Desarrollo del backend
Desarrollo del frontend
Eval. y Optimiz.
Pruebas y evaluación
Optimización y ajustes
Finalización
Elaboración de la memoria
Entrega del TFG
100 % complete
TODAY

Capítulo 4

ANÁLISIS

En este capítulo, se examina una etapa fundamental dentro del desarrollo de proyectos en ingeniería informática: el análisis y modelado de requisitos. Este proceso abarca diversas actividades esenciales, desde la identificación y validación de las necesidades del sistema hasta su estructuración y jerarquización, con el objetivo de garantizar una implementación coherente y alineada con los objetivos del proyecto.

El propósito principal de este análisis es asegurar que los requisitos sean claros, detallados, viables y medibles, proporcionando una base firme para el diseño y desarrollo del sistema. Contar con una especificación precisa desde el inicio minimiza posibles ambigüedades y reduce riesgos asociados a cambios inesperados en etapas avanzadas del proyecto.

En este capítulo se definirán los requisitos y se presentarán diferentes diagramas y representaciones gráficas, los cuales facilitarán la comprensión de la arquitectura del sistema y sus interacciones, proporcionando una visión estructurada de sus funcionalidades y comportamiento esperado.

4.1. Análisis de requisitos

El análisis de requisitos permite comprender, definir y documentar las necesidades del proyecto con el fin de garantizar que su diseño, implementación y pruebas se realicen de manera estructurada y eficiente. Durante este proceso, se identifican tanto las funcionalidades que debe ofrecer el sistema como las restricciones y características que influyen en su comportamiento.

Para este proyecto, se han definido dos tipos de requisitos fundamentales: funcionales y no funcionales. Los requisitos funcionales establecen de manera detallada las acciones y procesos que el sistema debe ejecutar para cumplir con su propósito, especificando las operaciones que estarán disponibles para los usuarios. Por otro lado, los requisitos no funcionales describen las condiciones bajo las cuales el sistema debe operar, abordando aspectos como el rendimiento, la seguridad, la usabilidad y otras restricciones que afectan su calidad y eficiencia.

Este análisis no solo busca garantizar que los requisitos sean precisos, alcanzables y trazables, sino también que proporcionen una base sólida para todas las fases posteriores del desarrollo. Además, se establecerán criterios de validación para asegurar que cada especificación definida se corresponda con los objetivos del proyecto y pueda ser implementada de manera efectiva. [24]

4.1.1. Requisitos Funcionales

Los requisitos funcionales son descripciones explícitas del comportamiento que debe tener una solución de software y la información que debe manejar. Estos requisitos detallan las acciones y procesos que el sistema debe ejecutar para cumplir con su propósito, especificando las operaciones disponibles para los usuarios [25].

1. **Recopilación, Importación y Tratamiento de Datos:** El sistema debe extraer automáticamente los datos de los presupuestos estructurados de la Junta de Andalucía en formato PDF. Además debe de poder actualizarse con nuevos presupuestos o documentos relacionados sin necesidad de actualizar de nuevo todos los documentos.

2. **Recuperación de Información Basada en Semántica:** El sistema debe ser capaz de procesar consultas en **lenguaje natural** relacionadas con los presupuestos, proporcionando información relevante sin necesidad de conocimientos técnicos en informática. Además, debe contar con un mecanismo de validación que notifique al usuario cuando la información recuperada sea insuficiente o inexistente, evitando así respuestas erróneas o sin fundamento (*alucinaciones*). Finalmente, el sistema debe generar respuestas enriquecidas mediante la combinación de datos extraídos de múltiples documentos presupuestarios, garantizando una visión integral y contextualizada de la información.

3. **Análisis y Visualización de Datos en respuestas:** El sistema debe permitir la *visualización de datos históricos comparativos en gráficas de barras justo des-*

pués de la respuesta asociada cuando aparezcan dichos datos en la respuesta. Además debe de permitir seleccionar qué modelo desea utilizar, el aspecto de la misma y un historial de conversaciones del chatbot.

4. **Evaluación de Modelos**: El sistema debe incluir métricas de evaluación de los modelos seleccionados analizando respuestas frente a una fuente de respuestas consideradas *"source of truth"*. Esta evaluación consistirá en evaluar las respuestas individualmente, con valores promedio o con ponderaciones de las métricas para decidir qué modelo es el más apropiado. Además esta "source of truth"tiene que ser *obtenida de forma dinámica a partir de la importación de un fichero .CSV de preguntas y respuestas*.

4.1.2. Requisitos No Funcionales

Los requisitos no funcionales complementan a los requisitos funcionales previamente definidos, estableciendo limitaciones y criterios que afectan el diseño y la implementación del sistema.

1. **Escalabilidad y Capacidad de Expansión:** El sistema tiene que ser capaz de manejar grandes volúmenes de datos estructurados sin comprometer el rendimiento. Además tiene que ser capaz de cambiar de configuración con tan solo un cambio en un fichero de configuración. También tiene que estar preparado para que con unos mínimos cambios de programación pueda ingerir ficheros de cualquier tipo.

2. **Optimización:** Tiene que integrar técnicas de caching en el código para mejorar la velocidad de respuesta en consultas recurrentes a un mismo modelo.

3. **Interfaz de Usuario y Experiencia de Usuario (UX):** La interfaz de usuario debe ser intuitiva y accesible, permitiendo la interacción fluida con el sistema sin necesidad de conocimientos técnicos avanzados.

4. **Rendimiento y Eficiencia:** El sistema debe responder en un tiempo adecuado para la generación de respuestas.

5. **Robustez:** El sistema tiene que ser fácilmente mantenible utilizando la *arquitectura cliente-servidor* de una parte de frontend y otra de backend. Además debe contar con una arquitectura modular, permitiendo la incorporación de nuevas funciones sin afectar a los componentes existentes y sin apenas esfuerzos aplicando patrones de diseño.

6. **Mantenibilidad:** El código debe seguir buenas prácticas de desarrollo.

7. **Accesibilidad:** El sistema debe ser accesible desde dispositivos móviles y de escritorio, permitiendo su uso en cualquier entorno.

8. **Precisión:** La precisión del sistema será determinada por una serie de métricas de evaluación de generación de texto estandarizadas.

9. **Seguridad:** Se debe implementar un nivel adecuado de seguridad para proteger la integridad y confidencialidad de los documentos almacenados en el sistema. El sistema solo permite la interacción con el servidor, no la modificación del mismo.

10. **Compatibilidad con soluciones de terceros:** El sistema debe ofrecer la opción de descargar la información y resultados de la evaluación con toda la información en formatos como CSV o JSON para que pueda ser tratada o mostrada en herramientas de terceros para su posterior uso.

4.2. Casos de Uso

Los casos de uso son una herramienta fundamental en el desarrollo de software, ya que describen, en términos de acciones e interacciones, cómo un usuario interactúa con el sistema para lograr un objetivo específico. Estos casos permiten capturar de manera clara los requisitos funcionales, sin depender de la implementación técnica, asegurando que el sistema cumpla con las necesidades del usuario.

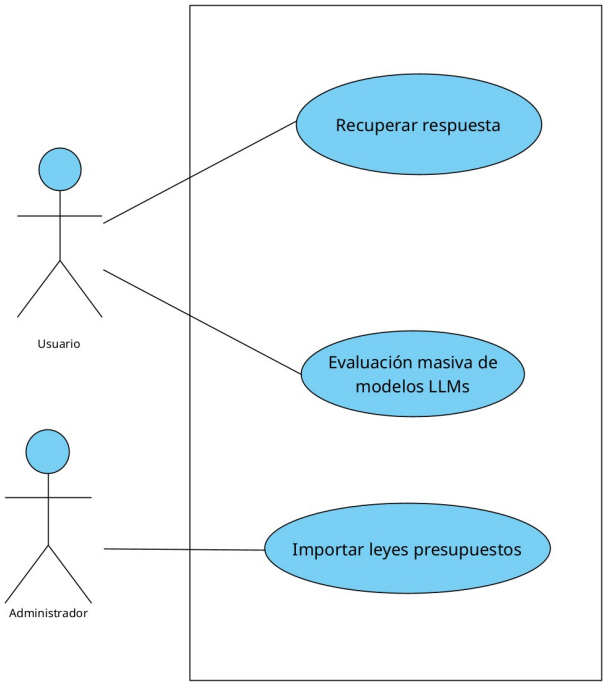


Figura 4.1: Diagrama de caso de uso general

A continuación, se presenta la representación textual de los casos de uso, proporcionando una descripción detallada de cada uno de ellos. La representación gráfica de estos casos se encuentra en la Sección 4.3 Diagramas de Casos de Uso y de Secuencia.

4.2.1. Recuperar respuesta

- **Actor principal:** Usuario
- **Objetivo:** Obtener una respuesta de una pregunta o concepto de los presupuestos especificado por el usuario.
- **Precondiciones:** El usuario debe tener acceso a la página web del sistema y haber accedido a la página principal. Opcionalmente puede seleccionar un modelo previamente.
- **Postcondiciones:** Mostrar la respuesta por pantalla utilizando el modelo seleccionado (por defecto siempre hay uno) y en el caso de que devuelva dicha respuesta datos comparativos con sus valores, visualizar un gráfico de barras. Seguidamente muestra la misma pregunta para una nueva interacción.
- **Flujo de eventos principal:**

1. El usuario accede a la página principal del sistema.
2. El usuario se sitúa en la parte central de la página principal del sistema.
3. El usuario hace click en el control de tabulación 'Chat'.
4. El sistema muestra una caja de texto y el usuario ingresa una pregunta o concepto.
5. El usuario pulsa Intro para enviar la pregunta al sistema.
6. El sistema devuelve al usuario el resultado de la pregunta visualizándolo seguidamente en pantalla, tras recuperar el contexto relacionado con la pregunta en la base de datos de leyes y tras tener seleccionado el modelo LLM.

El usuario, interesado en obtener una respuesta a partir de una pregunta, accede a la página principal. Una vez allí, el usuario se sitúa en la parte central de la página principal del sistema y hace click en el control de tabulación 'Chat'. A continuación muestra el inicio del chatbot con un mensaje de introducción. Seguidamente el sistema muestra una caja de texto y el usuario ingresa una pregunta o concepto. El usuario pulsa **Intro** para enviar la pregunta al sistema. Finalmente el sistema devuelve al usuario el resultado de la pregunta visualizándolo en pantalla tras recuperar el contexto relacionado con la pregunta en la base de datos de leyes, tener seleccionado el modelo LLM y tras confeccionar el LLM la respuesta. Si en la respuesta hay pares de información comparativos (datos-valores), el sistema visualiza seguidamente después de la respuesta un gráfico de barras. Por último, el sistema muestra la misma pregunta para una nueva interacción. A continuación se redactan los **sub-casos** de usos de heredan de este caso de uso principal:

4.2.1.1. Seleccionar el modelo para interactuar con el chatbot

- **Actor principal:** Usuario
- **Objetivo:** Cambiar el modelo del lenguaje (LLM) desde la interfaz de usuario para utilizarlo en el chatbot para la recuperación de la respuesta.
- **Precondiciones:** El usuario debe tener acceso a la página web del sistema y haber accedido a la página principal.
- **Postcondiciones:** Cambiar el modelo del lenguaje LLM de la interfaz del sistema. Solo puede estar activo simultáneamente uno que es el seleccionado.
- **Flujo de eventos principal:**

1. El usuario accede a la página principal del sistema.
2. El usuario se sitúa en la parte izquierda de la página principal del sistema.
3. El usuario hace click seleccionando un tema desde la sección 'Selecciona un modelo a evaluar'.

El usuario, interesado en cambiar el modelo LLM para evaluar individualmente el chatbot con preguntas, accede a la página principal. Una vez allí, se sitúa en la parte izquierda de la pantalla buscando la opción de 'Selecciona un modelo a evaluar'. En ese momento, el usuario hace click para cambiar al modelo que interactuar, pudiendo seleccionar solo uno del que el usuario desee. Finalmente, el modelo quedará seleccionado para poder interactuar a posteriori con el chatbot proporcionando al usuario la experiencia de elegir libremente el modelo que desee.

4.2.1.2. Mostrar historial de conversaciones

- **Actor principal:** Usuario
- **Objetivo:** Mostrar en pantalla el historial de conversaciones entre el sistema y el usuario del chat una vez que se ha producido la respuesta.
- **Precondiciones:** El usuario debe tener acceso a la página web del sistema y haber accedido a la página principal.
- **Postcondiciones:** Mostrar el historial de conversaciones por pantalla.
- **Flujo de eventos principal:**
  1. El usuario accede a la página principal del sistema.
  2. El usuario se sitúa en la parte izquierda de la página principal del sistema.
  3. El usuario hace scroll en la parte más inferior de la izquierda
  4. El sistema muestra el historial de conversaciones.

El usuario, interesado en no olvidar y saber en cada momento un resumen de lo que ha interactuado con el sistema, accede a la página principal. Una vez allí, se sitúa en la parte izquierda de la pantalla. En ese momento, el usuario realiza un scroll hacia abajo buscando la sección de 'Historial de Consultas (Chat). Finalmente, el usuario puede consultar y visualizar todo el historial del sistema con el chatbot.

4.2.1.3. Eliminar historial de conversaciones

- **Actor principal:** Usuario
- **Objetivo:** Mostrar en pantalla el historial de conversaciones entre el sistema y el usuario del chat una vez se ha producido la recuperación de la respuesta.
- **Precondiciones:** El usuario debe tener acceso a la página web del sistema y haber accedido a la página principal.
- **Postcondiciones:** Eliminar el historial de conversaciones por pantalla.
- **Flujo de eventos principal:**
  1. El usuario accede a la página principal del sistema.
  2. El usuario se sitúa en la parte izquierda de la página principal del sistema.
  3. El usuario hace scroll en la parte más inferior de la izquierda.
  4. El sistema muestra el historial de conversaciones.
  5. El sistema muestra justamente después un botón 'Limpiar Historial'.
  6. El usuario hace click en el botón 'Limpiar Historial'.

El usuario, interesado en limpiar el historial de conversaciones del chatbot, accede a la página principal. Una vez allí, se sitúa en la parte izquierda de la pantalla. En ese momento, el usuario realiza un scroll hacia abajo buscando la sección de 'Historial de Consultas (Chat). El usuario visualiza un botón para limpiar el historial llamado Limpiar Historial. Finalmente, el usuario hace click en el botón anterior permitiendo borrar todo el historial de conversaciones del sistema con el chatbot.

4.2.2. Evaluación masiva de modelos LLMs

- **Actor principal:** Usuario
- **Objetivo:** Evaluar en detalle por métricas los LLMs deseados.
- **Precondiciones:** El usuario debe tener acceso a la página web del sistema y haber accedido a la página principal. El usuario tiene que poseer en su poder un fichero .csv de preguntas y respuestas consideradas como 'source of truth'. Por último, el usuario debe seleccionar qué modelos LLM desea evaluar.

- **Postcondiciones:** Lanzar el proceso de evaluación en **segundo plano asíncrono**, para generar un fichero .csv identificando unívocamente con toda la información del proceso de la evaluación.
- **Flujo de eventos principal:**
  1. El usuario accede a la página principal del sistema.
  2. El usuario se sitúa en la parte central de la página principal del sistema.
  3. El usuario hace click en el control de tabulación 'Ejecución Evaluación masiva de Modelos'.
  4. El sistema muestra un primer paso para importar un fichero .csv con las preguntas y respuestas consideradas como 'source of truth'.
  5. El usuario arrastra el fichero/s al control de subida de ficheros. Alternativamente, el usuario puede hacer click en el botón 'Browse files' para subir ficheros, seleccionando el fichero que desea.
  6. El sistema muestra el fichero subido por el usuario indicando su nombre y su tamaño.
  7. El usuario pulsa el botón de 'Procesar archivos', para ejecutar el proceso de volcado de las preguntas y respuestas a evaluar en el sistema.
  8. El sistema muestra un mensaje 'Archivo/s procesados correctamente.' cuando este proceso de importación es un éxito.
  9. El sistema muestra un control de usuario para seleccionar qué modelos LLMs son los que quiere evaluar. Por defecto están todos seleccionados.
  10. El usuario selecciona/deselecciona los modelos LLMs que quiere evaluar.
  11. El sistema muestra al usuario un botón 'Evaluar'.
  12. El usuario hace click en el botón 'Evaluar' para iniciar el proceso de lanzamiento de evaluación.
  13. El sistema devuelve al usuario un mensaje de aviso indicando la ejecución con un ID único (Fecha, hora, segundos).
  14. El sistema devuelve al usuario un mensaje de aviso indicando que la visualización de las gráficas se realizan en el caso de uso 'Construcción de la Evaluación'.
- **Flujo alternativo:**
  1. Si el proceso de importación de ficheros de preguntas y respuestas produce un error, aparece en el sistema el mensaje 'Error al procesar los archivos.' y el motivo. Una vez solucionado el problema, volver a repetir proceso de arrastrar el fichero/s al control de subida de ficheros y seguir la secuencia.

2. Si el proceso de evaluación de modelos produce un error, aparece en el caso de uso ’Construcción de la Evaluación’ un listado de logs de errores. indicando el motivo del error. Una vez solucionado el error, puede repetir el proceso volviendo al paso inicial o reiniciando la aplicación por medio del caso de uso ’Reiniciar aplicación’.

El usuario, queriendo evaluar los modelos sobre las preguntas y respuestas consideradas como ’source of truth’, accede a la página principal del sistema. En esta página central, se le presenta un control de usuario en forma tabular. El usuario hace click en el control de tabulación ’Evaluación masiva de Modelos’. Una vez pulsada la opción, el sistema muestra una serie de pasos para poder iniciar la evaluación. El primer paso será importar un fichero .csv con las preguntas y respuestas consideradas como ’source of truth’. En este momento, el usuario arrastra un fichero/s al control de subida de ficheros. Alternati vamente, el usuario puede hacer click en el botón ’Browse files’ para subir ficheros, seleccionando el fichero que desea. Tras la selección, el sistema muestra el fichero subido por el usuario indicando su nombre y su tamaño. Tras mostrar esta información, el usuario pulsa el botón de ’Procesar archivos’, para ejecutar el proceso de volcado de las preguntas y respuestas a evaluar en el sistema. Si existe algún problema de formato o de importación, el sistema muestra un mensaje de error (’Error al procesar los archivos.’) y el motivo.

Si el proceso de importación es un éxito, el sistema muestra un mensaje de conformidad (’Archivo/s procesados correctamente.’). Seguidamente, el sistema muestra un control de usuario para seleccionar qué modelos LLMs son los que quiere evaluar. Por defecto están todos seleccionados. El usuario selecciona o deselecciona los modelos según desee. El sistema muestra al usuario un botón ’Evaluar’. Seguidamente, el usuario hace click en el botón ’Evaluar’ para iniciar el proceso de lanzamiento de la evaluación en **segundo plano asíncrono**. El sistema devuelve al usuario un mensaje de aviso indicando la ejecución con un ID único (Fecha, hora, segundos).Por último el sistema devuelve al usuario un mensaje de aviso indicando que la visualización de las gráficas se realizan en el caso de uso ’Construcción de la Evaluación’.

Si el proceso de ejecución de evaluación de modelos produce un error, aparece en pantalla en el caso de uso ’Construcción de la Evaluación’ un listado de logs de errores indicando el motivo del error. Una vez solucionado el error, puede repetir el proceso volviendo al paso inicial o reiniciando la aplicación por medio del caso de uso ’Reiniciar aplicación’.

A continuación se redactan los **sub-casos** de usos de heredan de este caso de uso principal:

4.2.2.1. Construcción Evaluación

- **Actor principal:** Usuario
- **Objetivo:** Renderizar en pantalla en cualquier momento que desee el usuario toda la información de datos y gráficas de cualquier evaluación ejecutada (caso de uso 'Evaluar modelos LLMs') desde un histórico de ejecuciones.
- **Precondiciones:** El usuario debe tener acceso a la página web del sistema y haber accedido a la página principal. El usuario tiene que lanzar previamente el proceso de evaluación de modelos (Caso de Uso Evaluar modelos LLMs).
- **Postcondiciones:** Mostrar datos y gráficas por métricas para una evaluación detallada por preguntas individuales, por media de scores de métricas y ponderando dichas métricas para obtener un resultado final de evaluación del modelo por pantalla.
- **Flujo de eventos principal:**
  1. El usuario accede a la página principal del sistema.
  2. El usuario se sitúa en la parte central de la página principal del sistema.
  3. El usuario hace click en el control de tabulación 'Construcción Evaluación'.
  4. El sistema muestra un listado de resumen de evaluaciones.
  5. El sistema muestra un control de selección para seleccionar cual evaluación desea mostrar las gráficas.
  6. El usuario selecciona una ejecución que será basada en nombres de ficheros .csv de ejecuciones.
  7. El sistema mostrará un resumen de la información que guarda.
  8. El sistema mostrará un botón 'Construcción Evaluación' para iniciar el proceso de construcción.
  9. El usuario hace click en el botón 'Construcción Evaluación'
  10. El sistema muestra la renderización en pantalla de toda la información, gráficas de análisis y el resultado de la evaluación por modelos.
- **Flujo alternativo:**
  1. Si el proceso de ejecución de modelos (Caso de uso Evaluar modelos LLMs) produce un error, aparece en el sistema el mensaje de advertencia 'Se han detectado errores en evaluaciones recientes' el motivo del error. Puede repetir el proceso volviendo al paso inicial de ese caso de uso o reiniciando la aplicación por medio del caso de uso 'Reiniciar aplicación'.

El usuario, queriendo conocer la evaluación de los modelos, accede a la página principal del sistema. En esta página central, se le presenta un control de usuario en forma tabular. El usuario hace click en el control de tabulación 'Construcción Evaluación'. Una vez pulsada la opción, el sistema muestra un listado de resumen de evaluaciones. Además el sistema debe de mostrar un control de selección para seleccionar cual evaluación desea mostrar las gráficas y sus datos. El usuario selecciona una ejecución ya realizada que será basada en nombres de ficheros .csv de ejecuciones por fecha y hora. Una vez seleccionado, el sistema mostrará un resumen mínimo de la información que guarda para sea más sencillo al usuario la verificación de la ejecución.

Después el sistema mostrará un botón 'Construcción Evaluación' para iniciar el proceso de construcción. El usuario hace click en el botón 'Construcción Evaluación' y el sistema muestra la renderización en pantalla de toda la información, gráficas de análisis y el resultado de la evaluación por modelos.

4.2.2.2. Exportar Evaluación

- **Actor principal:** Usuario
- **Objetivo:** Exportar en formato .CSV o .JSON en detalle el resultado de la evaluación de modelos.
- **Precondiciones:** El usuario debe tener acceso a la página web del sistema y haber concluido el proceso (caso de uso) de 'Construcción de la evaluación' de modelos.
- **Postcondiciones:** Exportar los resultados de la evaluación en un fichero formato .CSV o .JSON.
- **Flujo de eventos principal:**
  1. El usuario accede a la página principal del sistema.
  2. El usuario se sitúa en la parte central de la página principal del sistema.
  3. El usuario hace click en el control de tabulación 'Exportación datos métricas'.
  4. El sistema muestra un control de selección para elegir el formato de exportación.
  5. El usuario selecciona el formato que desea (.CSV o .JSON).
  6. El sistema muestra un botón para iniciar el proceso de exportación.
  7. El usuario pulsa el botón para iniciar el proceso de exportación.

8. El sistema muestra un mensaje de éxito si el proceso de exportación ha concluido correctamente y el usuario dispone de sus datos exportados en el formato deseado.

- **Flujo alternativo:**

    1. Si el proceso devuelve un error, el sistema muestra un mensaje. Si no hay datos de evaluación, muestra un mensaje de aviso.

El usuario, interesado en exportar los datos de la evaluación de forma conjunta, accede a la página principal. Una vez allí, se sitúa en la parte central de la pantalla buscando la opción de 'Exportar Datos Métricas'. En ese momento, el usuario selecciona qué formato desea exportar. Una vez seleccionado, el sistema muestra el botón de Exportar. El usuario hace click para iniciar el proceso exportación. Si el proceso devuelve un error, el sistema muestra un mensaje. También si no hay datos de evaluación, muestra un mensaje de aviso. El sistema muestra un mensaje de éxito si el proceso de exportación ha concluido correctamente. Finalmente el usuario obtiene el fichero de la exportación en el formato deseado en la carpeta configurada como descargas del navegador.

4.2.3. Importar leyes presupuestos

- **Actor principal:** Administrador
- **Objetivo:** Importar por primera vez todo el histórico de leyes de presupuestos en el sistema.
- **Precondiciones:** El administrador debe tener acceso a la consola o terminal del sistema operativo donde se aloja el sistema. Además el sistema debe acceder a la ruta de ficheros del sistema donde se alojan las leyes.
- **Postcondiciones:** Insertar en la base de datos toda la información de los documentos que forman parte del histórico de leyes del presupuesto y mover a una carpeta 'processed' el fichero importado, manteniendo la estructura de carpetas.
- **Flujo de eventos principal:**

    1. El administrador accede a la consola del sistemas operativo donde está alojado el sistema.
    2. El administrador ejecuta un comando o script para iniciar la importación indicando la ruta donde se hayan la ubicación de la información.

3. El sistema muestra en consola paso a paso (Fichero a fichero) el detalle del proceso de importación.
4. El administrador espera a que termine la ejecución del proceso. Una vez terminado, aparece un mensaje de éxito en la consola indicando que ha terminado el proceso.

- **Flujo alternativo:**

1. Si el sistema muestra un mensaje de error en consola, el administrador soluciona el error y vuelve a iniciar el flujo de eventos principal de este caso de uso.

El administrador desea proveer la base de conocimiento al sistema, necesitando importar los datos de las leyes del presupuesto. El administrador accede a la consola del sistema operativo donde está alojado el sistema. El administrador ejecuta un comando o script indicando la ruta donde se encuentran los ficheros del corpus. El sistema muestra en consola todo el proceso de importación paso a paso con detalle fichero a fichero. Una vez terminado, el sistema muestra un mensaje de éxito y mueve todas las leyes procesadas a una nueva carpeta 'processed'. Si por contra, el sistema muestra un mensaje de error en consola, el administrador soluciona el error y vuelve a iniciar el proceso de importación empezando por el inicio de los pasos de este caso de uso.

4.3. Diagramas de Casos de Uso y de Secuencia

Los diagramas de casos de uso son representaciones gráficas que muestran cómo los usuarios interactúan con un sistema y las funciones que este ofrece. Facilitan la comprensión de los procesos al visualizar de manera clara las relaciones entre los actores y sus acciones, proporcionando una perspectiva estructurada de la funcionalidad del sistema [26].

4.3.1. Diagramas de Casos de Uso

El diagrama de la Figura 4.1 representa la funcionalidad del sistema completo. El siguiente caso de uso ilustra el proceso de recuperación de información dentro del sistema, desglosando su funcionamiento en varias etapas clave. En primer lugar, el usuario introduce una pregunta para obtener el contexto relevante basado en leyes y

normativas. Posteriormente, este contexto, junto con el sub-caso de la selección del modelo LLM que se muestra a continuación, se emplea para generar una respuesta a través del modelo seleccionado. Finalmente, el sistema evalúa si la información obtenida requiere una representación visual mediante una gráfica comparativa, asegurando así una interpretación más clara y detallada de los datos.

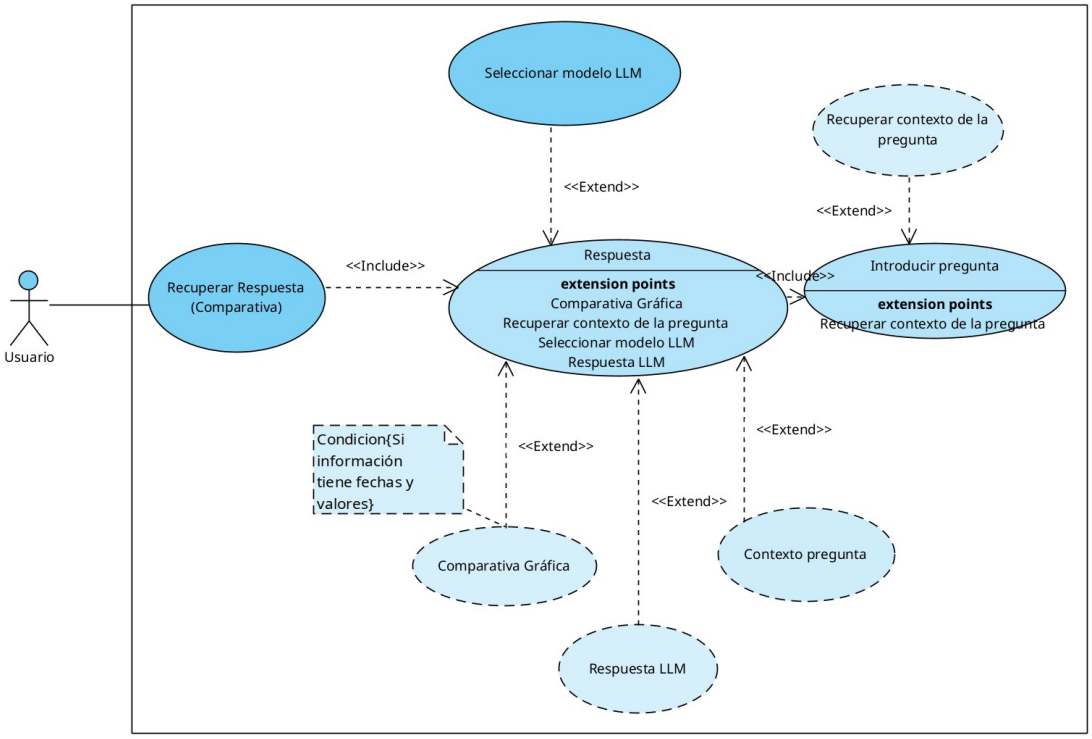


Figura 4.2: Diagrama de caso de uso: Recuperar respuesta (Comparativa)

A continuación aparece el siguiente sub-caso de uso, Figura 4.3, muestra cómo el usuario puede seleccionar el modelo LLM, necesitando para ello utilizar el caso de uso anterior Figura 4.2).

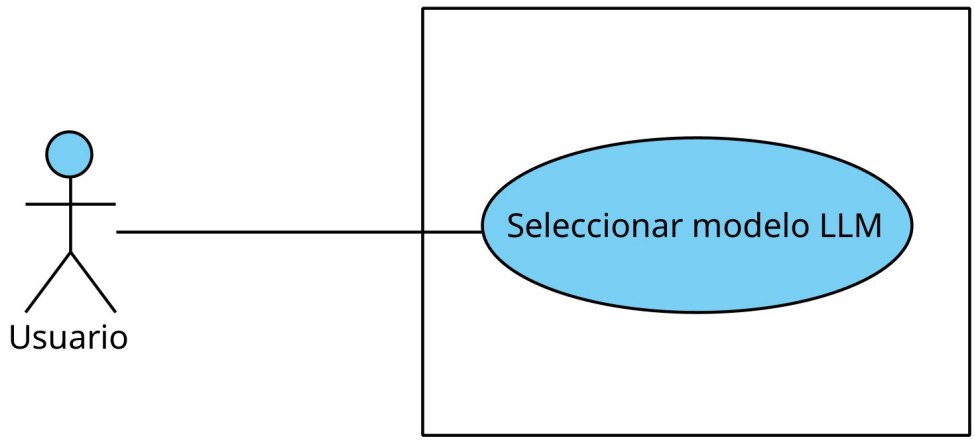


Figura 4.3: Diagrama de sub-caso de uso: Seleccionar el modelo LLM

Este nuevo sub-caso de uso nos enseña el proceso de mostrar el historial de conversaciones entre el chatbot y el usuario, necesitando para ello utilizar el caso de uso padre Figura 4.2).

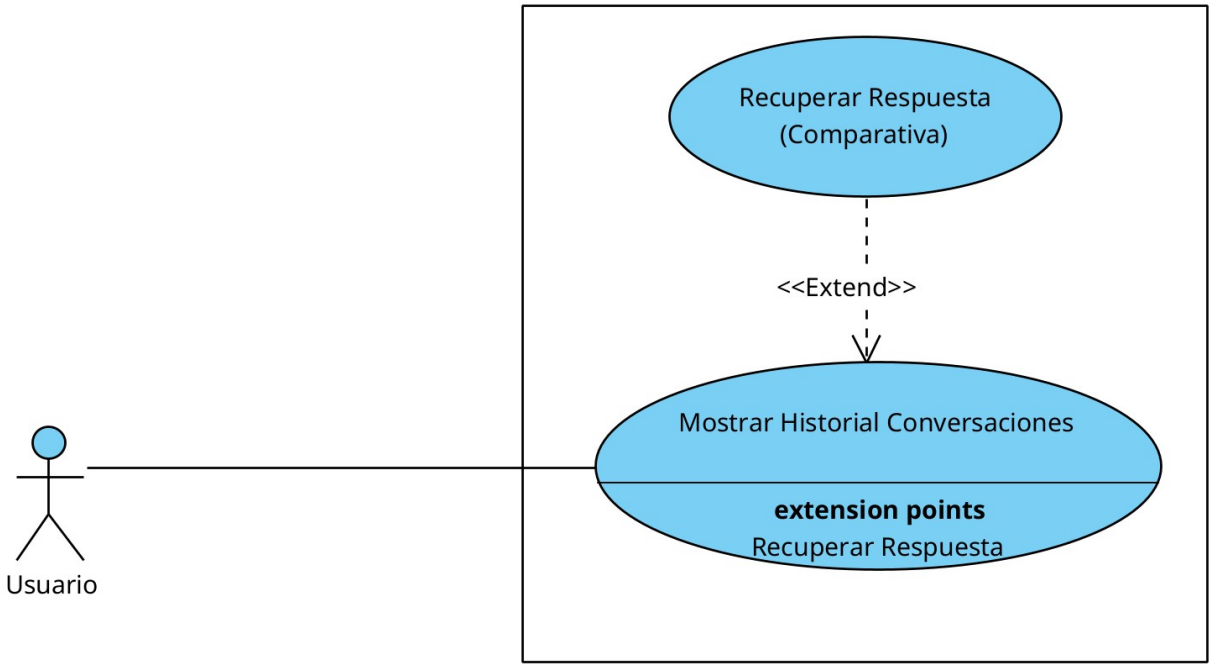


Figura 4.4: Diagrama de sub-caso de uso: Mostrar el historial de conversaciones

El siguiente sub-caso de uso, muestra como el usuario puede borrar el historial de conversaciones en cualquier momento. Para ello es necesario que previamente haya realizado del caso de uso padre Figura 4.2).

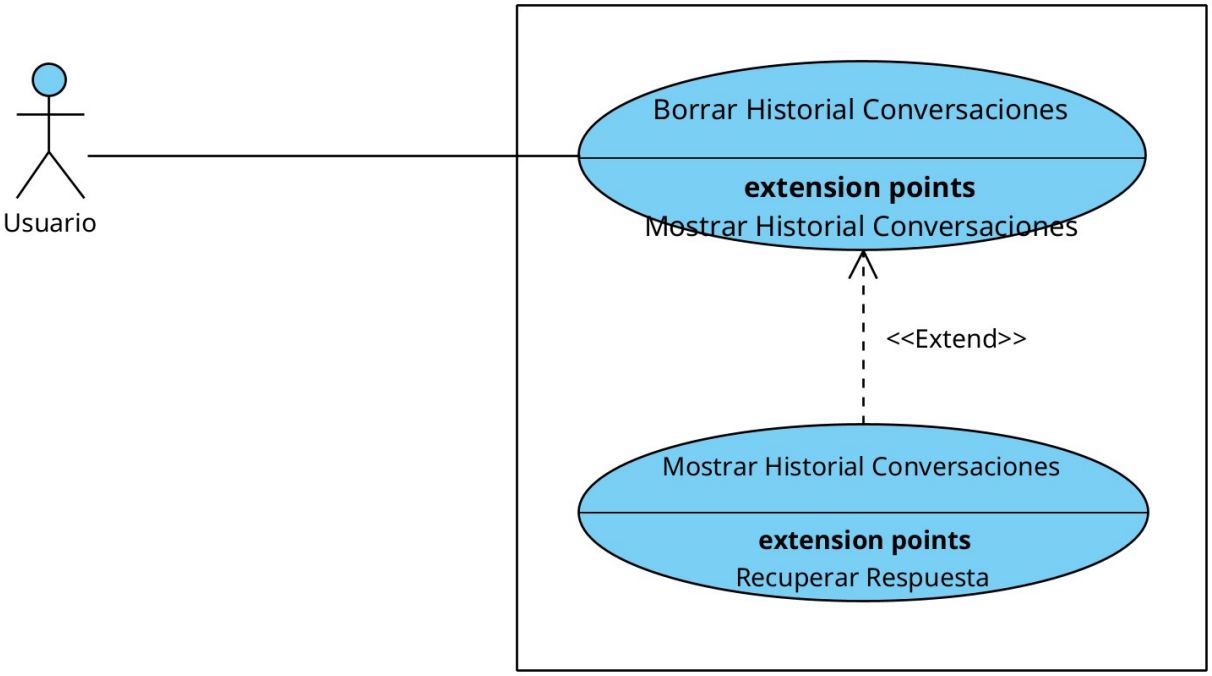


Figura 4.5: Diagrama de sub-caso de uso: Borrar el historial de conversaciones

Este nuevo caso de uso, muestra como se puede ejecutar el proceso de evaluación masiva de modelos LLMs. Para ello, necesita capturar los ficheros de preguntas y respuestas importando dicha información desde ficheros y seleccionando cualquier LLM de la lista que desee. Una vez seleccionado, se produce la ejecución en segundo plano de la evaluación, necesitando para ello el caso de uso Figura 4.2, generando el fichero final .csv de evaluación.

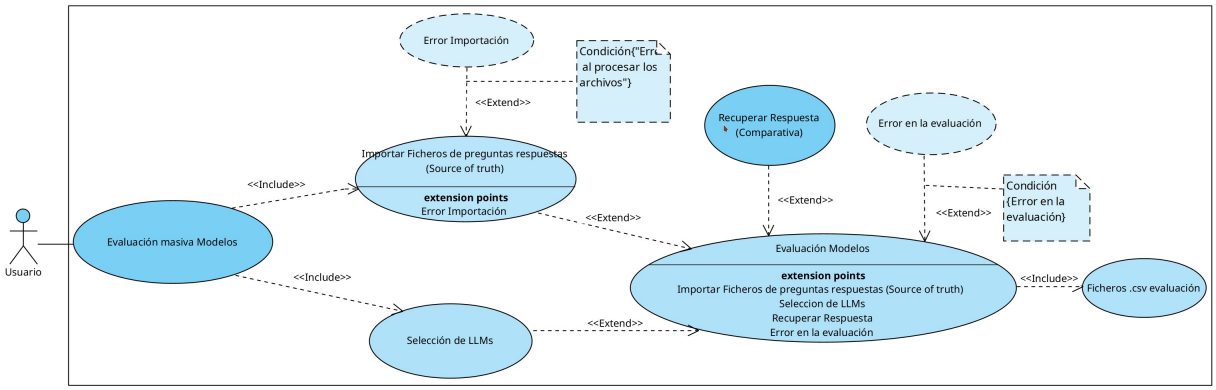


Figura 4.6: Diagrama de caso de uso: Evaluación masiva de Modelos

Este sub-caso de uso, muestra la construcción de la evaluación con toda la información y el renderizado de los gráficos cuando el usuario desee una determinada evaluación previamente ejecutada en el caso de uso Figura 4.6.

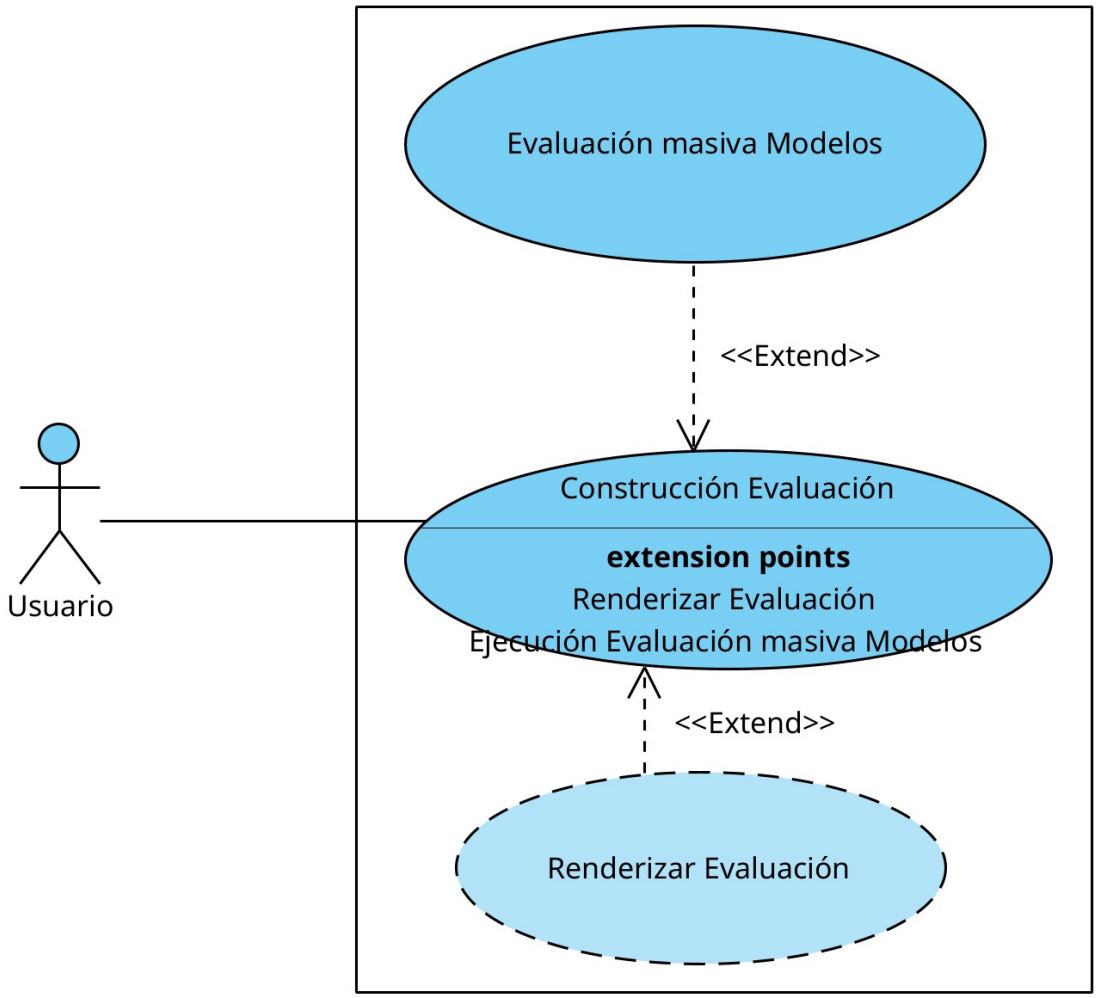


Figura 4.7: Diagrama de sub-caso de uso: Construcción Evaluación

Este sub-caso de uso, ilustra el proceso de exportación a los formatos establecidos de toda la información obtenida previamente del proceso de Construcción de la Evaluación (Figura 4.7).

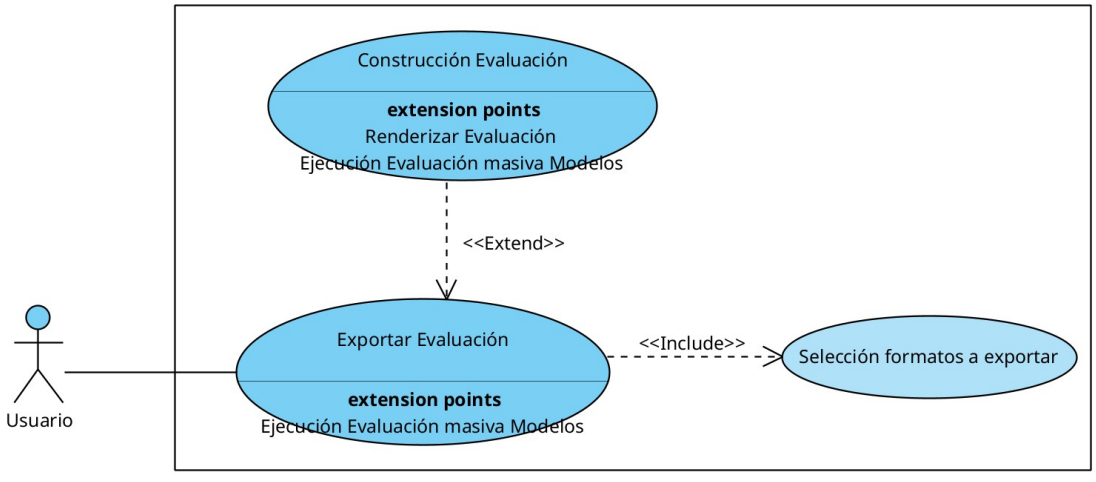


Figura 4.8: Diagrama de sub-caso de uso: Exportar Evaluación

Este último caso de uso, nos muestra como el administrador del sistema puede importar leyes del presupuesto al sistema RAG, para que el usuario de final de la aplicación pueda responder a todas la preguntas que necesite iniciando el caso de uso Figura 4.2.

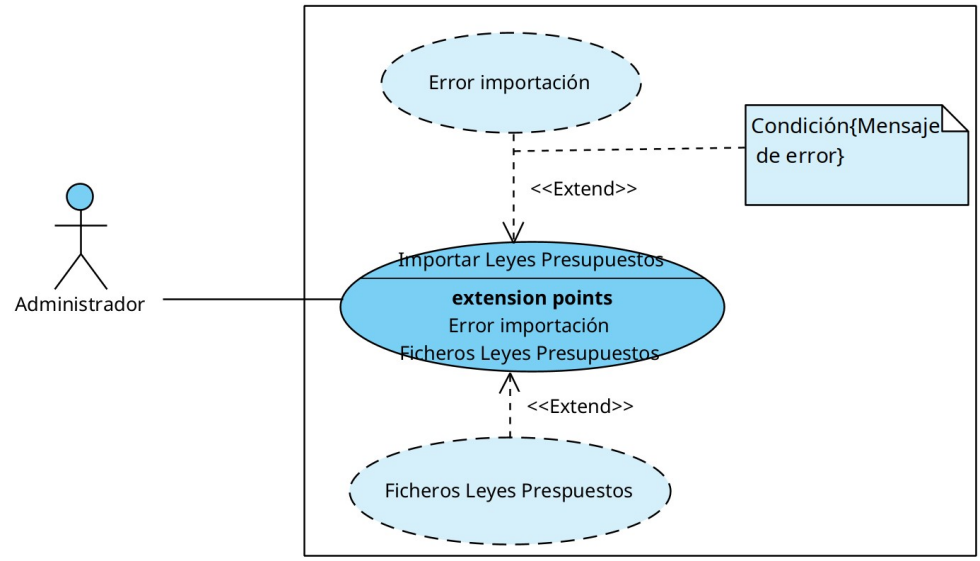


Figura 4.9: Diagrama de caso de uso: Importar leyes de presupuestos

4.3.2. Diagramas de Secuencia de los casos de uso

Los diagramas de secuencia son una herramienta fundamental en la modelización de sistemas, ya que permiten visualizar la interacción entre los distintos componentes de un sistema a lo largo del tiempo. Representan el flujo de mensajes entre los actores y los objetos involucrados, facilitando la comprensión del comportamiento dinámico del sistema y ayudando a identificar puntos críticos en la comunicación, optimizar procesos y garantizar una implementación coherente con los requisitos definidos (Booch, Rumbaugh y Jacobson, 2005) [27].

Los diagramas de secuencia complementan a los diagramas de casos de uso, ya que detallan cómo se llevan a cabo los escenarios definidos en los casos de uso y explican cómo se realizan esas interacciones paso a paso, proporcionando una visión detallada de la comunicación entre los actores y el sistema (Fowler, 2004) [28]. A continuación se muestran los diagramas de secuencia que tienen relación con cada uno de los diagramas de caso de uso:

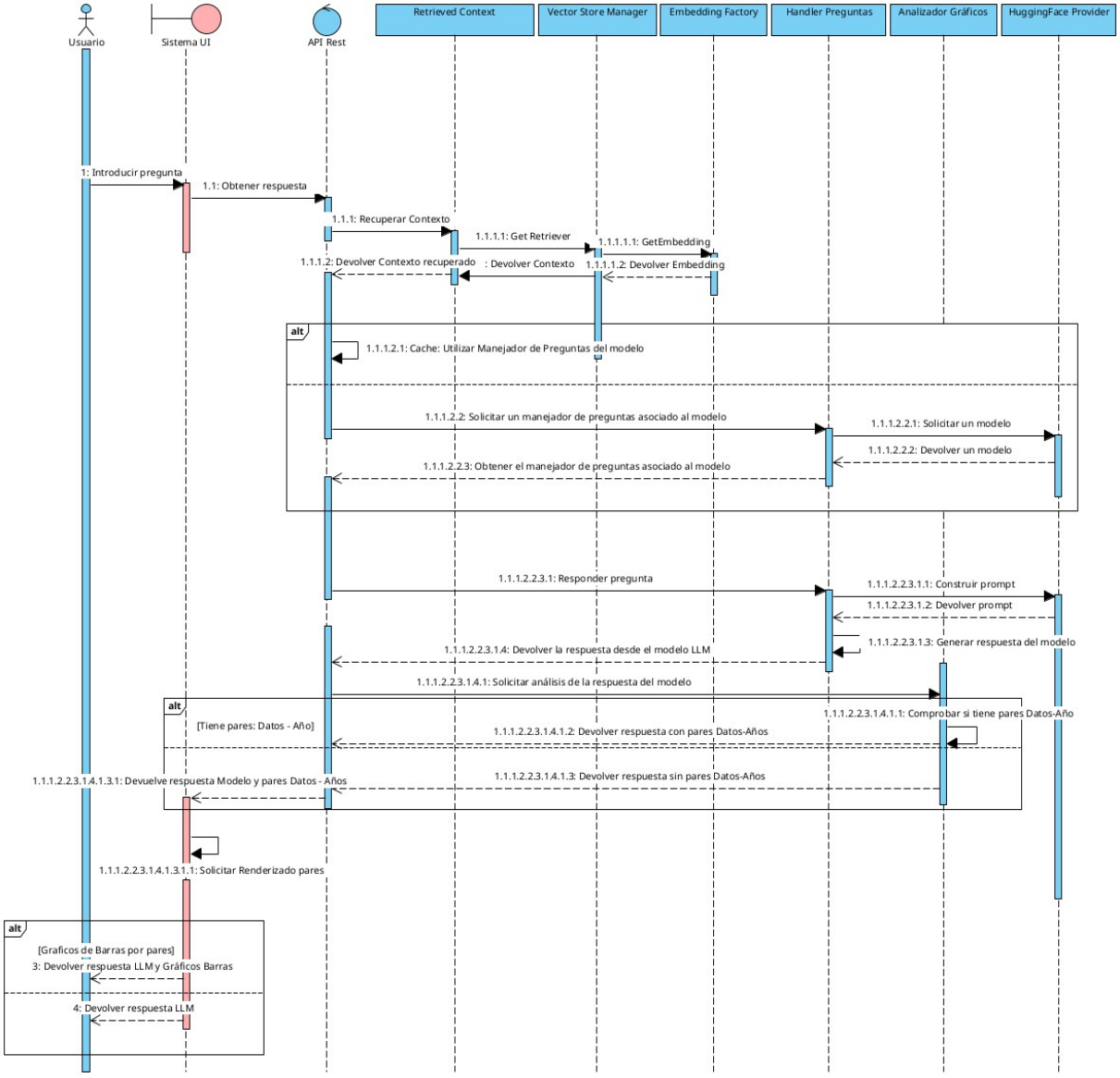


Figura 4.10: Diagrama de secuencia: Recuperar respuesta (Comparativa).

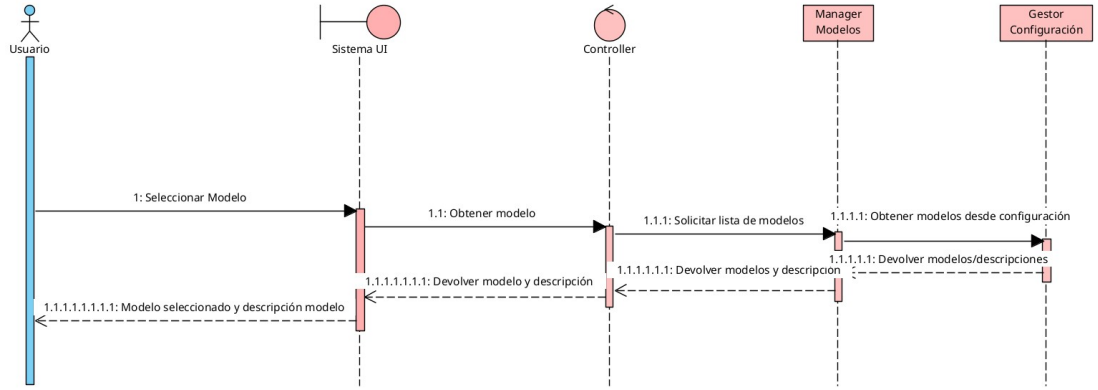


Figura 4.11: Diagrama de secuencia: Seleccionar el modelo LLM a usar.

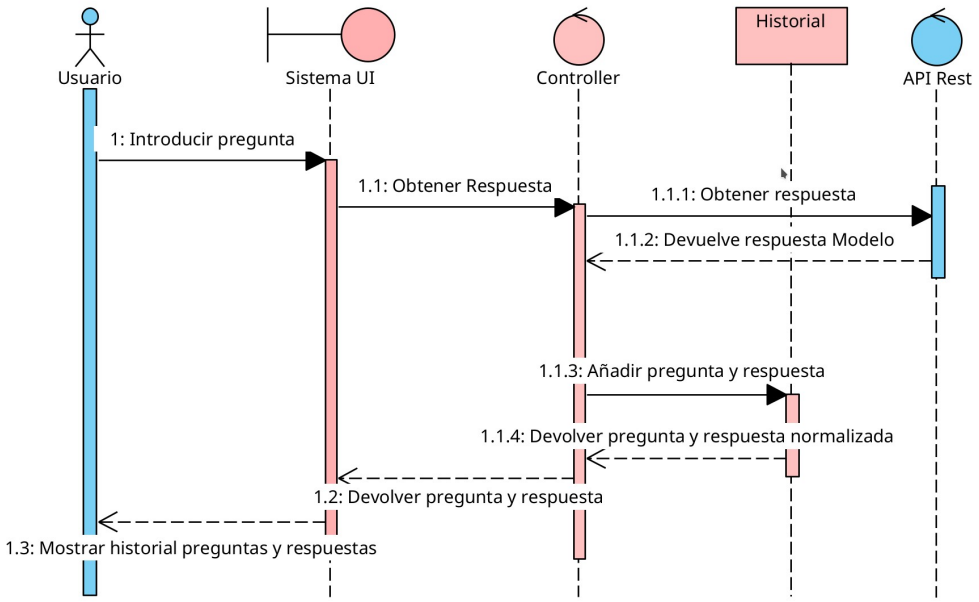


Figura 4.12: Diagrama de secuencia: Mostrar historial de conversaciones.

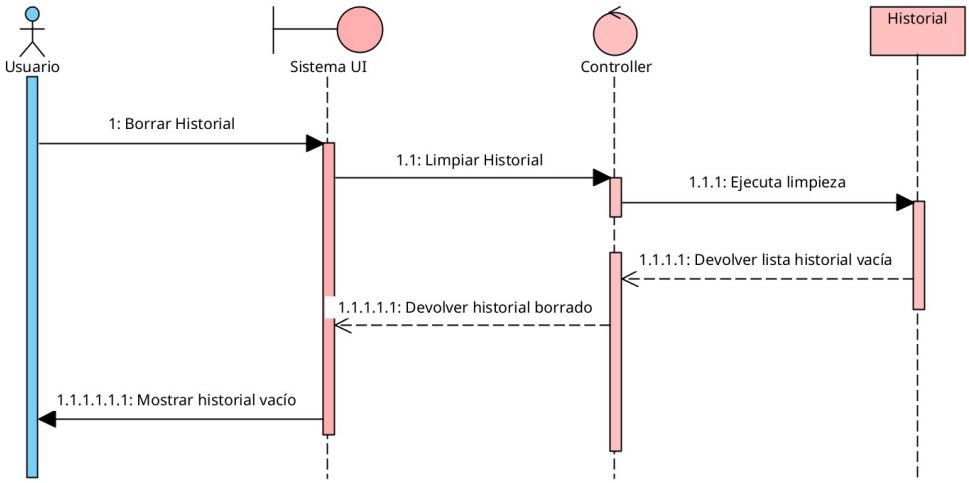


Figura 4.13: Diagrama de secuencia: Borrar historial de conversaciones.

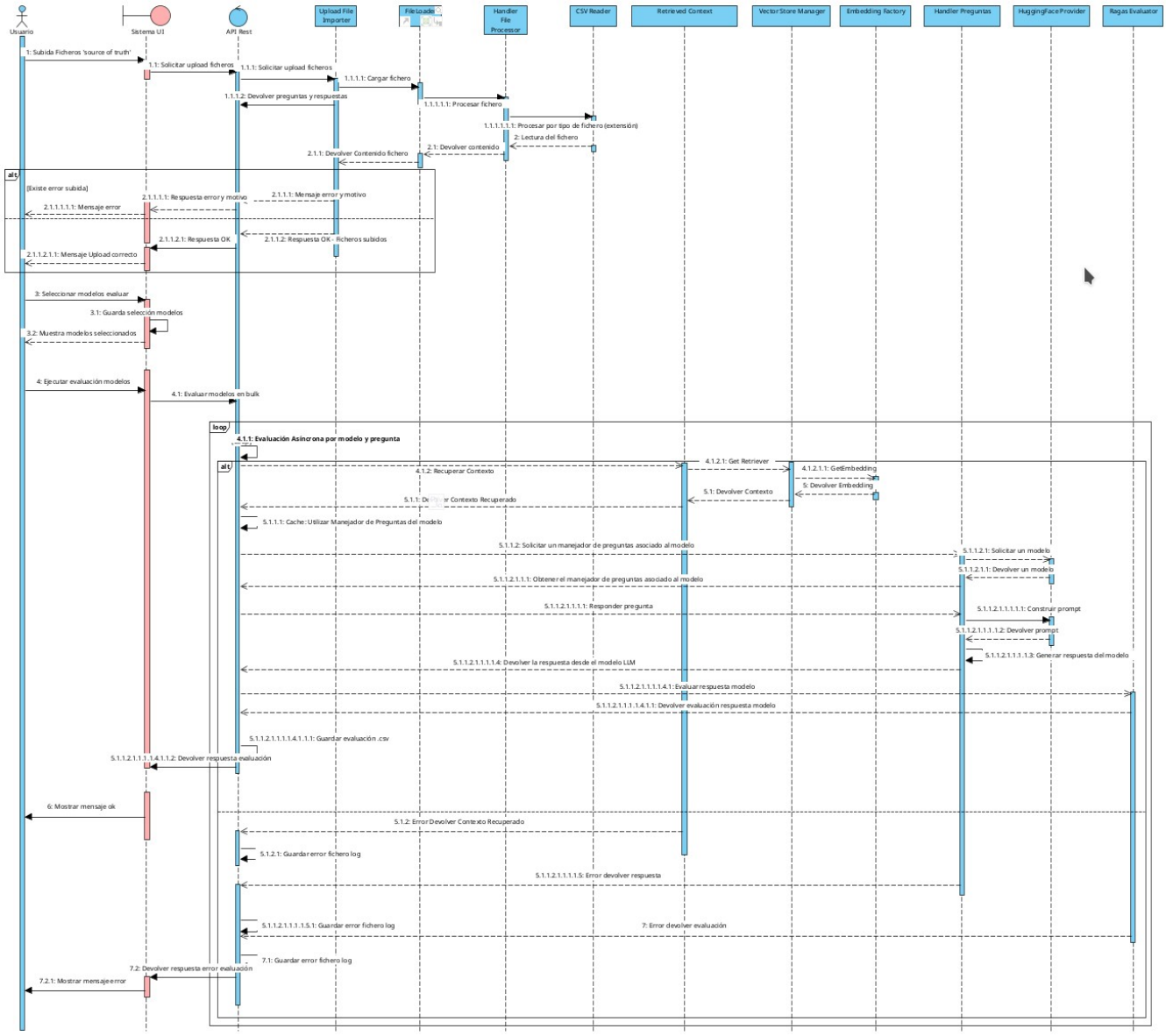


Figura 4.14: Diagrama de secuencia: Evaluación masiva de modelos LLMs.

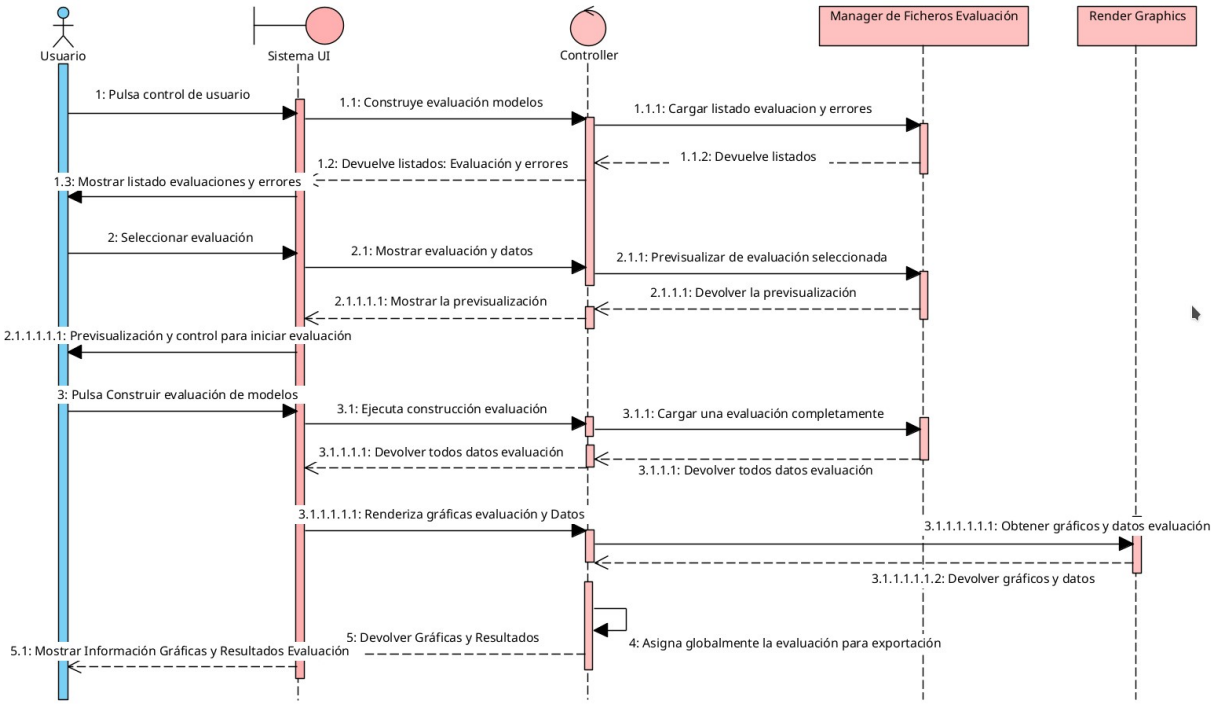


Figura 4.15: Diagrama de secuencia: Construcción de la evaluación masiva de modelos LLMs.

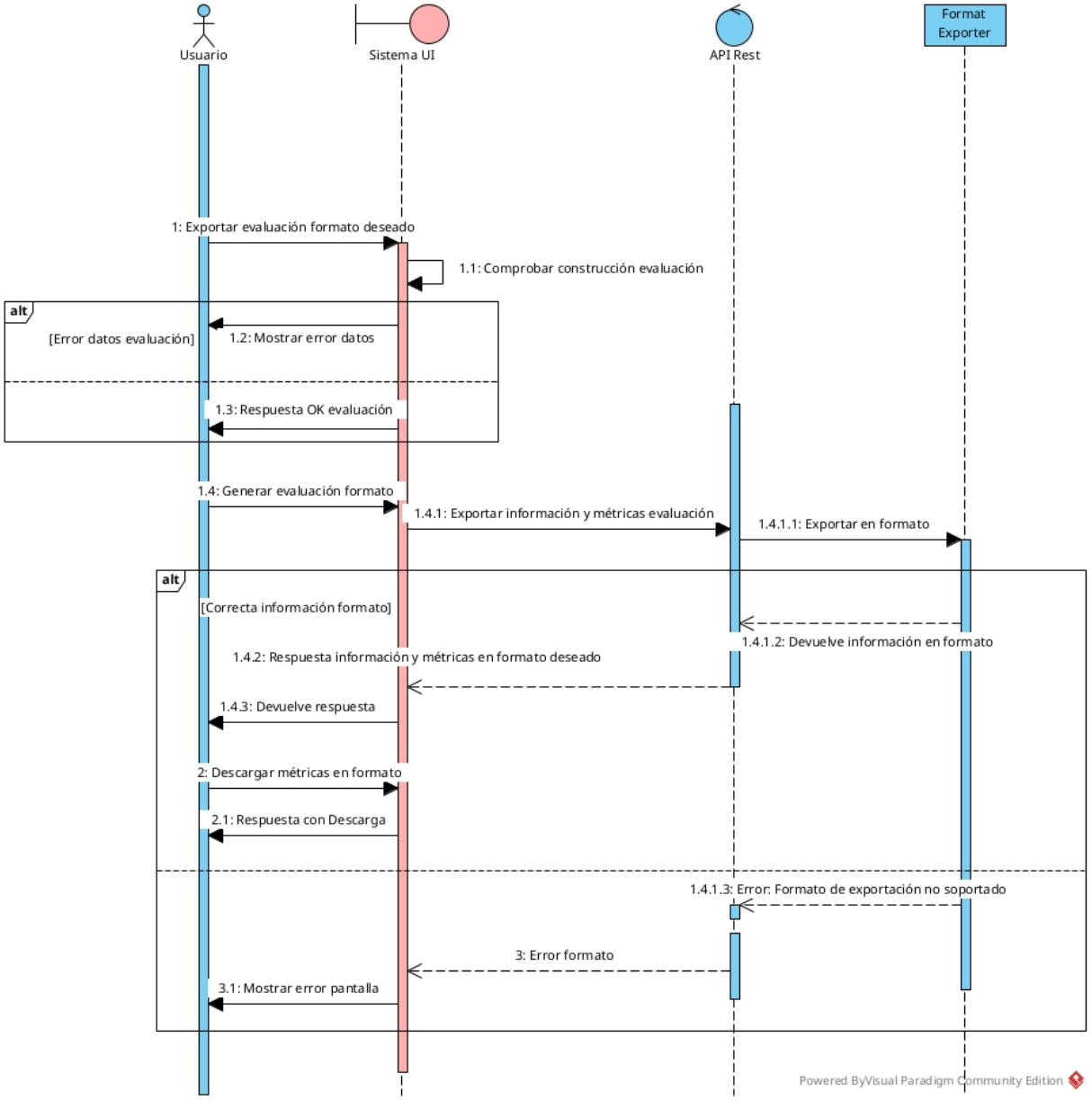


Figura 4.16: Diagrama de secuencia: Exportar Evaluación Modelos.

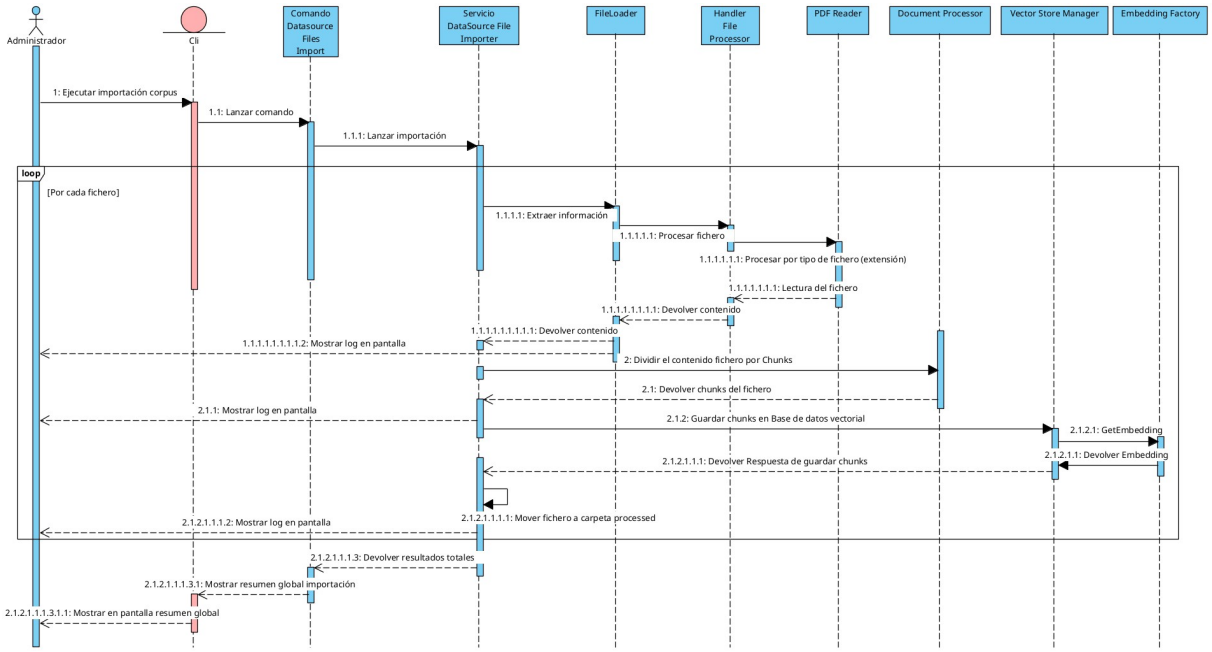


Figura 4.17: Diagrama de secuencia: Importar leyes del presupuesto.

4.3.3. Diagramas de Clase de Análisis

En el desarrollo de software, los diagramas de clases de análisis desempeñan un papel fundamental al ofrecer una representación estructurada de los elementos clave que intervienen en el sistema. Estos diagramas permiten identificar, organizar y visualizar las relaciones entre las entidades del dominio, proporcionando un marco conceptual sólido que facilita la transición hacia el diseño e implementación del sistema. Es por ello que este tipo de diagramas se enfocan en capturar la esencia del problema y estructurar una solución viable desde una perspectiva conceptual.

Cada elemento dentro del diagrama se clasifica en tres tipos principales de clases:

- **Clases límite**, que gestionan la interacción entre el sistema y sus usuarios, actuando como intermediarios en el procesamiento de datos.
- **Clases control**, responsables de coordinar el flujo de ejecución dentro del sistema y garantizar la coherencia de los procesos.
- **Clases entidad**, que representan los datos y comportamientos del dominio de aplicación, modelando los elementos persistentes y su relación con otros componentes.

Por último indicar que estos diagramas permiten ajustar y refinar la arquitectura del sistema antes de proceder a su desarrollo, asegurando una solución alineada con las necesidades del proyecto. Con esto, en la (Figura 4.18) observamos el diagrama resumen creado, donde las páginas correspondientes a la interfaz de usuario, partes del layout de dicha interfaz y para el actor administrador, la consola del sistemas son clases límite; las clases control se corresponden a los componentes que realizarán las tareas de recuperación de respuesta, exportar evaluación, lanzar evaluación y construir la evaluación. También esta incluido un nivel más especificación como es el recuperar contexto, analizar respuesta, devolver el modelo pipeline. La entidades son los contextos recuperados de la base de datos de conocimiento.

En el caso del administrador, las clase límite es la propia consola y las clase control es la que se encarga de importar los ficheros de base de datos de conocimiento. La clase entidades son los contextos guardados de la base de datos de conocimiento.

En la (Figura 4.18) se presenta un diagrama resumen que refleja la organización estructural del sistema. En él, se distinguen las clases límite, que abarcan los elementos relacionados con la interfaz de usuario, incluyendo las páginas del sistema, los componentes del diseño visual y, en el caso del administrador, la consola del sistema.

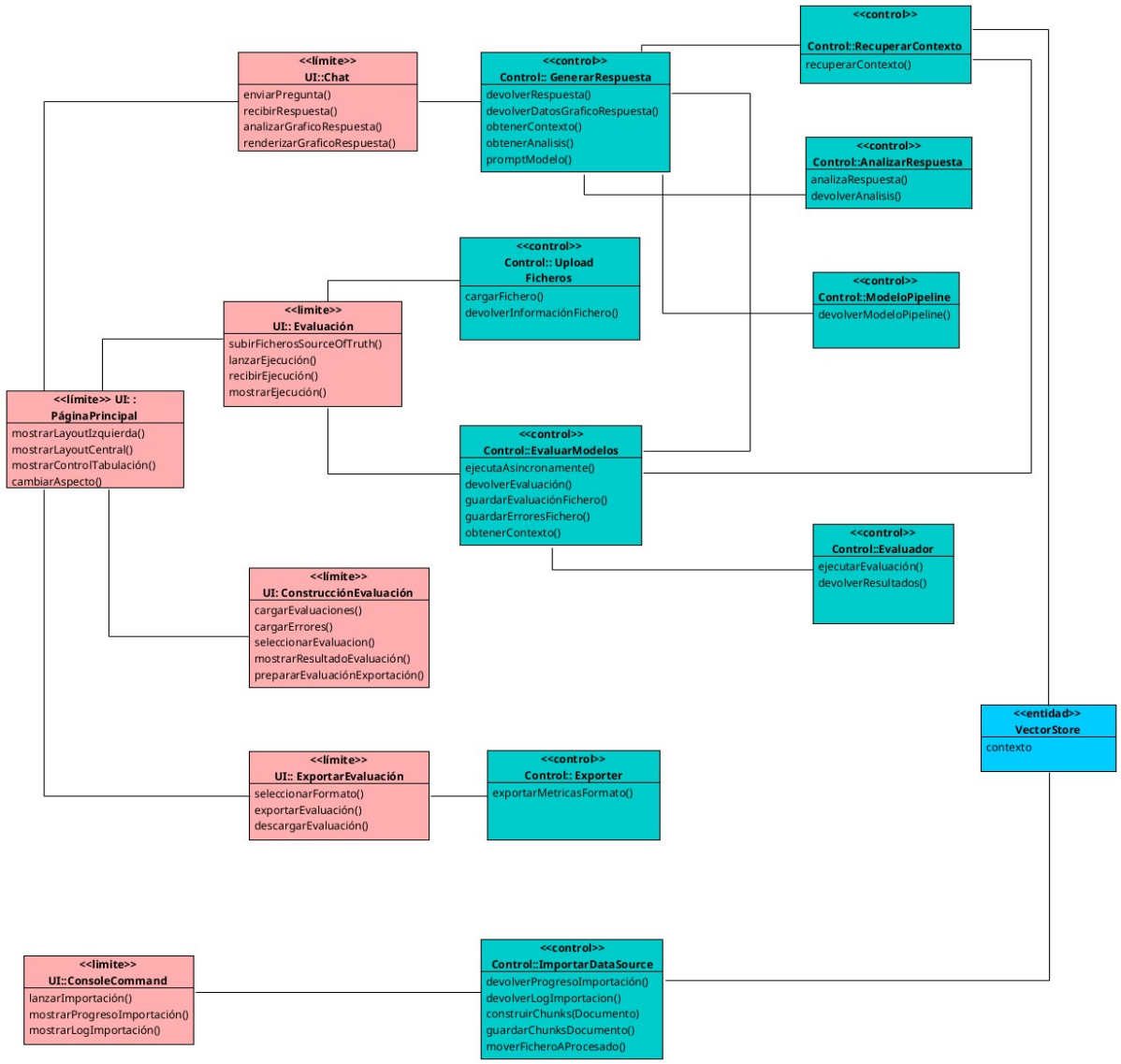


Figura 4.18: Diagrama de clases de Análisis.

Por otro lado, las clases control agrupan los componentes encargados de la ejecución de las principales funcionalidades, como la recuperación de respuestas, la exportación y evaluación de modelos, y la generación del análisis de evaluación. Además, se incluye un nivel más detallado dentro de estas clases, donde se definen procesos específicos como la recuperación del contexto, el análisis de la respuesta y la gestión del modelo de procesamiento (pipeline). En el caso del administrador, la clase control corresponde al proceso de importación de nuevos ficheros dentro de la base de conocimiento

Finalmente, las clases entidad representan los datos almacenados en la base de conocimiento del sistema, conteniendo la información recuperada que sirve como contexto para las consultas realizadas por los usuarios. Para el caso del administrador, salva los documentos divididos en 'contexto'.

Capítulo 5

DISEÑO

El diseño del sistema RAG para la consulta de leyes presupuestarias y su posterior evaluación se ha estructurado con el objetivo de garantizar eficiencia, precisión y accesibilidad en la recuperación y generación de información.

La arquitectura propuesta integra modelos de lenguaje avanzados junto con bases de datos vectoriales, asegurando que la información recuperada sea relevante y contextualizada. Además, se detallarán las decisiones de diseño adoptadas, enfocadas en la escalabilidad y en la correcta integración de las tecnologías empleadas, proporcionando así una base sólida para el desarrollo e implementación del sistema.

5.1. Introducción

El diseño del sistema RAG para la consulta de leyes presupuestarias es una fase fundamental en su desarrollo, ya que permite transformar la arquitectura conceptual en una estructura funcional y operativa. En esta etapa, se definen los componentes clave del sistema, asegurando que cada uno cumpla con los requisitos establecidos y garantizando una integración eficiente entre la interfaz de usuario, el motor de recuperación de información/modelo de lenguaje y la evaluación de dichos modelos.

Es por ello, la importancia de dos aspectos esenciales: el **diseño de salida** y el **diseño de datos.** Éstos son esenciales en la construcción del sistema RAG para la consulta de leyes presupuestarias y de evaluación de modelos, ya que garantizan tanto la correcta presentación de la información al usuario como la eficiencia en la gestión y almacenamiento de los datos procesados.

- **Diseño de Salida:** Se centra en la manera en que el sistema presenta la información recuperada y generada, asegurando que los usuarios puedan interpretar y utilizar los datos de forma clara y efectiva. Se debe priorizar una presentación intuitiva, estructurada y visualmente comprensible, facilitando la búsqueda de información en lenguaje natural, la visualización de respuestas estructuradas y la evaluación de modelos. Además, cuando las respuestas incluyan comparaciones de datos presupuestarios, el sistema debe ser capaz de generar representaciones gráficas como gráficos de barras o tablas dinámicas, mejorando así la interpretación de los resultados. Otro factor relevante en el diseño de salida es la posibilidad de exportar los datos obtenidos en formatos estándar como CSV o JSON, permitiendo su integración en otros sistemas de análisis financiero o documentación oficial. Este enfoque proporciona versatilidad al sistema, permitiendo a los usuarios manipular y reutilizar la información de acuerdo con sus necesidades.
- **Diseño de Datos:** El diseño de datos, por otro lado, se enfoca en la estructura, almacenamiento, recuperación eficiente de la información dentro del sistema. Dado que el RAG opera sobre un conjunto de documentos presupuestarios extensos, es fundamental contar con una base de conocimiento optimizada para consultas semánticas, permitiendo recuperar información con precisión y rapidez. Además, el diseño de datos debe considerar la escalabilidad y eficiencia del sistema, permitiendo la incorporación de nuevas leyes presupuestarias sin comprometer el rendimiento.

En este capítulo, se abordarán en detalle las decisiones adoptadas en el diseño de salida y el diseño de datos, explicando cómo estas estrategias contribuyen a la usabilidad del sistema, la precisión en la recuperación de información y la capacidad de adaptación a futuros volúmenes de datos.

5.2. Arquitectura del sistema

El sistema desarrollado se basa en una arquitectura RAG (Retrieval-Augmented Generation) para optimizar la recuperación y generación de información a partir de una base de conocimiento compuesta por documentos de leyes del presupuestos. Mediante este enfoque, se combinan una extracción semántica de acuerdo a la consulta y la generación de texto con prompting a modelos de lenguaje avanzados para extraer respuestas precisas y contextualizadas. Su diseño está orientado a automatizar el acceso y la comprensión de información compleja, proporcionando respuestas

estructuradas de manera eficiente y mejorando la interacción del usuario con los datos oficiales y científicos. También hay que destacar que la arquitectura facilita la evaluación dinámica de modelos del lenguaje y su posterior evaluación. Por último, destacar el proceso de inserción de la base de datos de conocimiento, fundamental para poder recuperar la información. En la figura Figura 5.1 se puede observar como los compo-

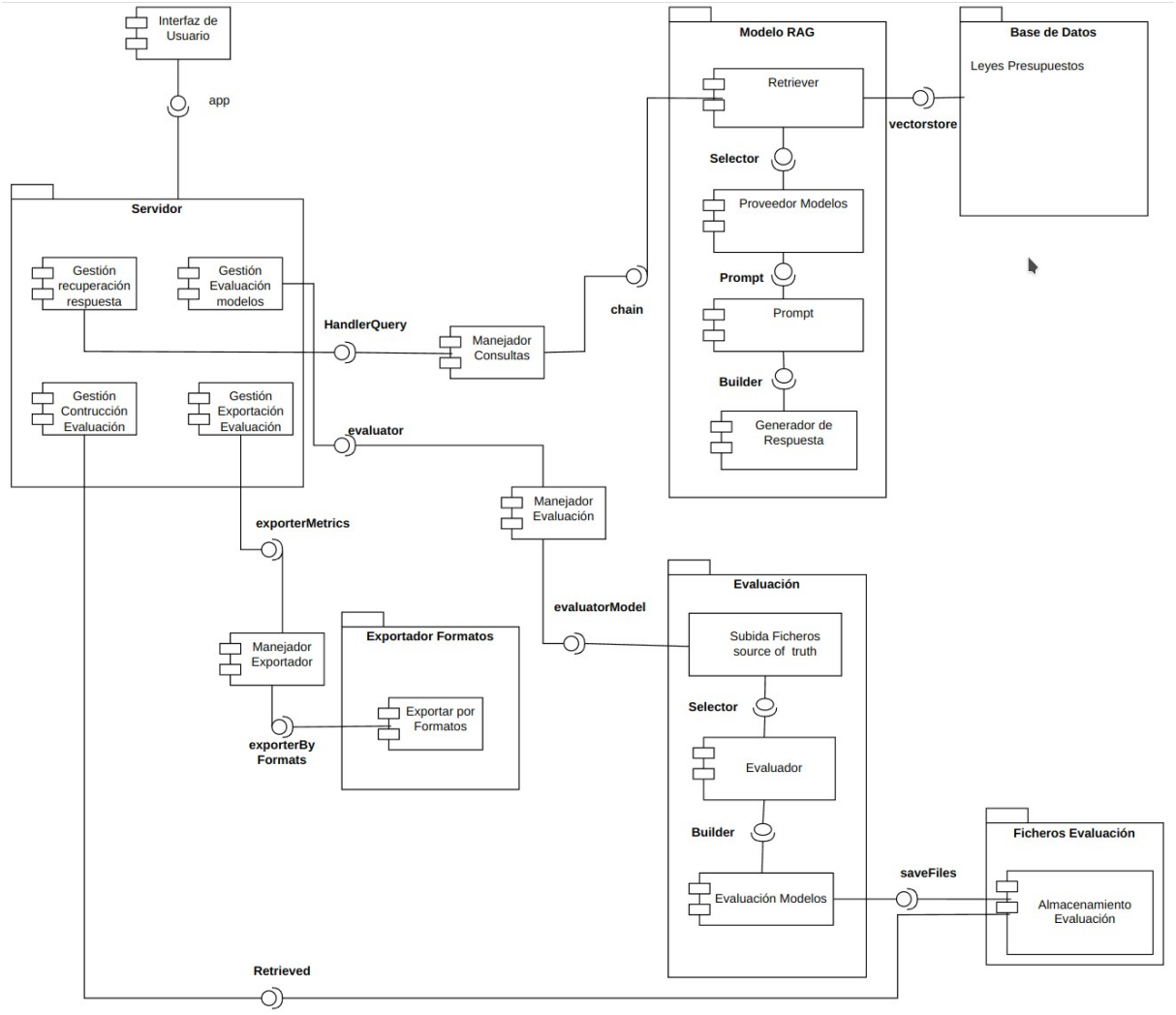


Figura 5.1: Diagrama de componentes del sistema - Usuario.

nentes se relacionan entre sí y como se conectan. El flujo de trabajo empieza cuando el usuario realiza distintas acciones como recuperar una respuesta a partir de una pregunta, ejecutar la evaluación de modelos, construir la evaluación de modelos y exportar la información de dicha evaluación. Se observa el servidor, el modelo RAG, la parte de generación de evaluación, construcción de la misma, exportación y la base de datos de leyes presupuestarias. Este diseño facilita la recuperación de información y la evaluación de modelos de lenguaje. En la figura Figura 5.2 muestra el detalle del proceso de importación y almacenamiento de documentos de leyes de presupuestos en la base de conocimiento. Incluye la interfaz de administración (consola), el gestor de inserción de leyes, el módulo de importación de archivos y la base de datos vecto-

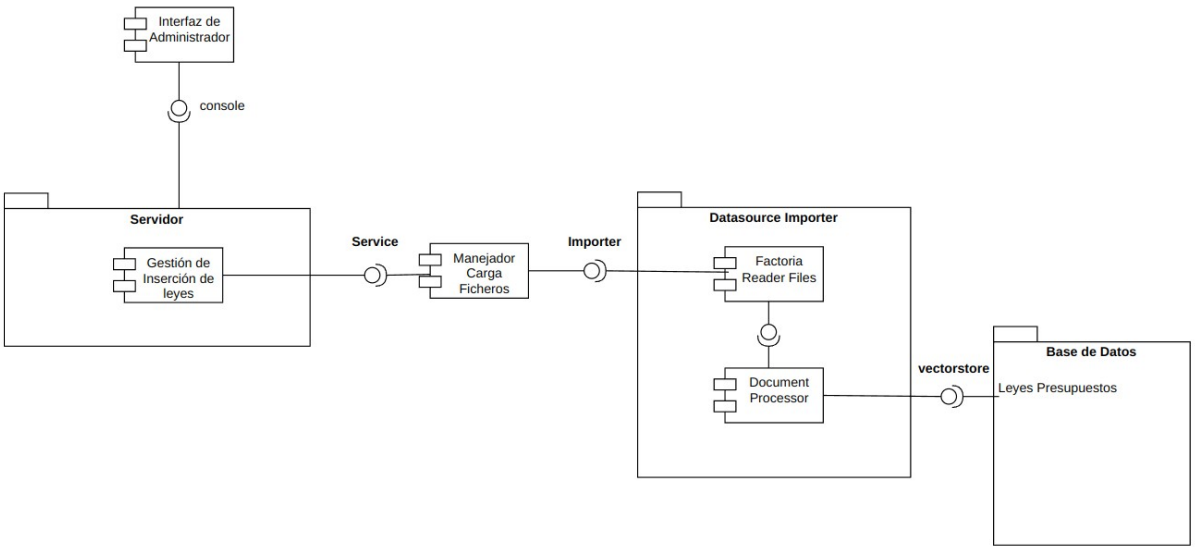


Figura 5.2: Diagrama de componentes del sistema - Administrador.

rial. Este flujo permite la carga eficiente de nuevas normativas presupuestarias en el sistema.

5.2.1. Diagrama de despliegue

Otra forma de mostrar el diseño de un sistema, es a través de los diagramas de despliegue. Estos representan la distribución física de los componentes de software en los distintos nodos de hardware, permitiendo visualizar la infraestructura en la que opera el sistema. Son esenciales en la ingeniería de software para definir la arquitectura de implementación y garantizar una correcta integración entre los elementos del sistema. Su uso facilita la planificación y optimización del rendimiento en entornos distribuidos. En la figura Figura 5.3 representa la ejecución del sistema RAG, donde un usuario interactúa a través de un navegador web con un servidor central que gestiona la recuperación de información, la evaluación de modelos y la exportación de datos, conectándose con el modelo RAG y la base de conocimiento. En la figura Figura 5.4 el diagrama ilustra el proceso de importación de leyes presupuestarias, donde un servidor y un administrador administra la inserción de documentos mediante un manejador de archivos y un procesador de documentos, almacenándolos en una base de conocimiento.

En definitiva, estos diagramas proporcionan una visión clara de la arquitectura y el flujo de datos en el sistema.

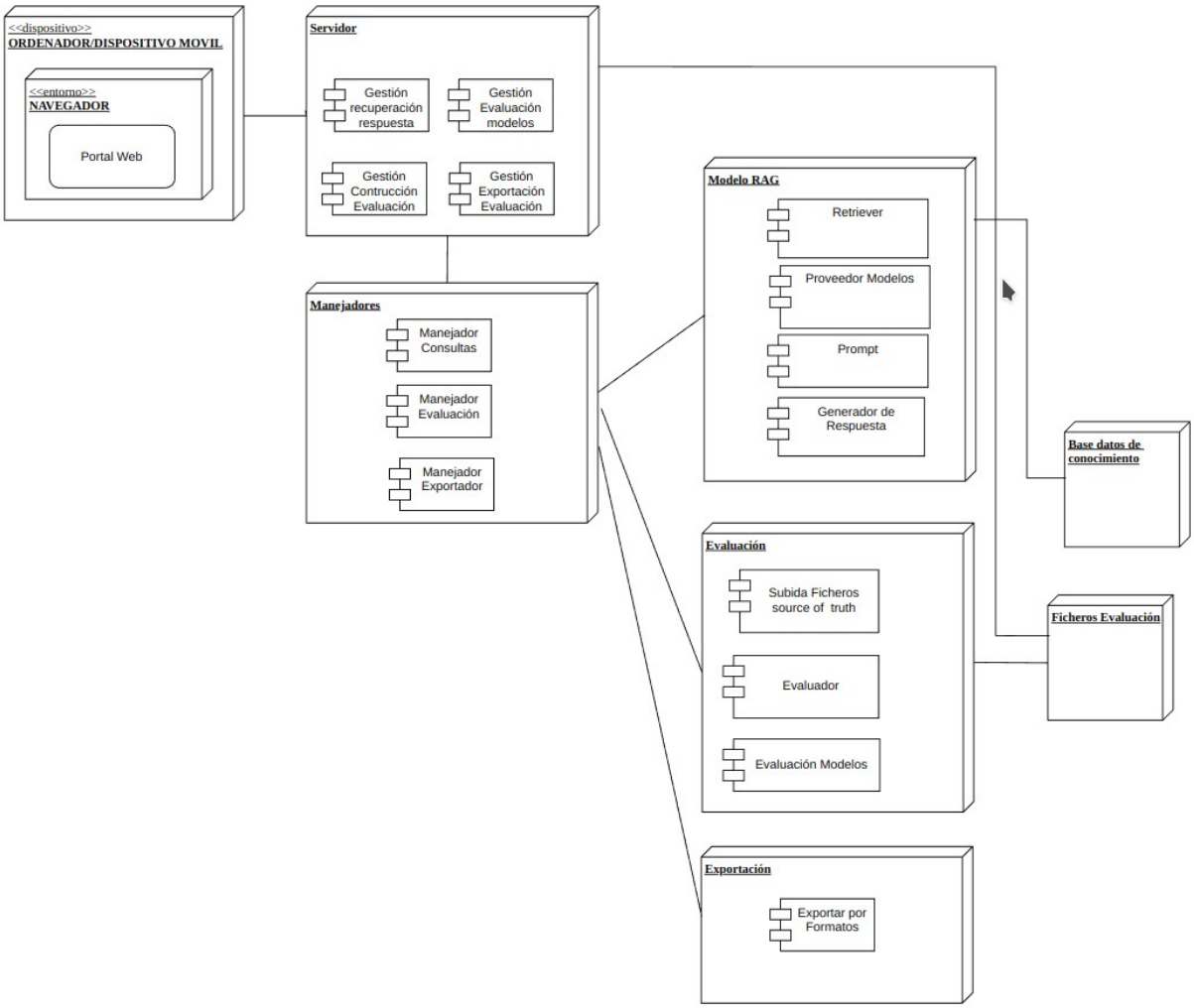


Figura 5.3: Diagrama de despligue del sistema - Usuario.

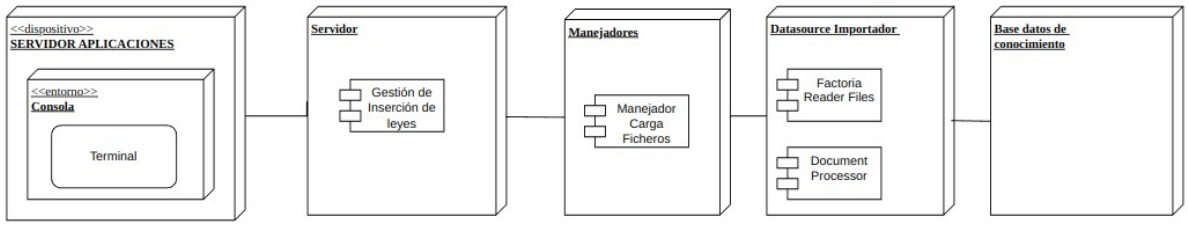


Figura 5.4: Diagrama de despligue del sistema - Administrador.

5.2.2. Diseño de datos

Para abordar el Diseño de Datos, es fundamental definir cómo se estructuran, almacenan y gestionan los datos en el sistema. Para ello, se han procesado todas las leyes de presupuestos y documentos relacionados con ellas de los últimos 21 años (Desde 2004). Es esencial que, según el método de almacenamiento elegido, permita recuperar la información de manera rápida y precisa.

La técnica de almacenamiento utilizada es la llamada **Representación vectorial de palabras** o también denominada como **word embedding** [29]. Esta técnica es utilizada para transformar el contenido textual de los documentos de leyes del presupuesto en representaciones vectoriales de menor dimensión (chunks). Este enfoque permite modelar la semántica del lenguaje natural, asignando a cada palabra o conjunto de palabras un vector en un espacio continuo, lo que facilita la búsqueda y recuperación de información relevante dentro de la base de datos. A diferencia de los métodos tradicionales basados en coincidencias exactas de palabras clave, los embeddings capturan relaciones semánticas, mejorando la precisión de las respuestas generadas por el sistema.

Para generar estos embeddings, se utilizan modelos de aprendizaje automático entrenados en grandes volúmenes de datos textuales. Algoritmos como **word2vec**, desarrollado por Google en 2013, o técnicas como **GloVe y FastText** [30], permiten construir representaciones vectoriales eficientes mediante el análisis del contexto en el que aparecen las palabras. Estas técnicas permiten que términos con significados similares tengan representaciones cercanas en el espacio vectorial, facilitando la recuperación de documentos presupuestarios incluso cuando se utilizan términos distintos pero equivalentes en una consulta.

La integración de word embeddings en el sistema ofrece múltiples beneficios. En primer lugar, optimiza la velocidad y eficiencia de las consultas, ya que la búsqueda se realiza en un espacio numérico en lugar de texto sin procesar. Además, reduce el almacenamiento necesario, ya que los vectores suelen ocupar menos espacio que los documentos originales. Finalmente, esta estructura es altamente escalable, permitiendo la incorporación progresiva de nuevas leyes y presupuestos sin comprometer el rendimiento del sistema.

5.2.3. Modelo RAG, Evaluación Modelos, Exportación evaluación e inserción de base del conocimiento

Es necesario dentro del diseño, poner especial importancia e hincapié, en el detalle de dicho diseño de los componentes del sistema como son la recuperación de la respuesta (dentro de RAG),la evaluación y construcción del análisis de los Modelos, la exportación de la Evaluación, para utilizarlo en terceros, y la inserción de datos de la base de conocimiento dentro de la base vectorial capaz de soportar el diseño de datos anteriormente explicado en el punto anterior.

Como uno de los objetivos de este trabajo, es lograr la escalabilidad, eficiencia y

fácil mantenibilidad y legibilidad del código, se ha elegido una programación basada en clases e interfaces. En consecuencia de lo anterior, se han seguido técnicas de **Patrones de diseño** [31]. Esto posibilita que cualquier necesidad nueva acontecida en el tiempo, pueda adaptarse y evolucionar de acuerdo a nuevos requisitos de este proyecto.

A continuación se van a mostrar todos los componentes y su diseño de clases indicando las características más destacadas en relación a esta perspectiva. En la figura

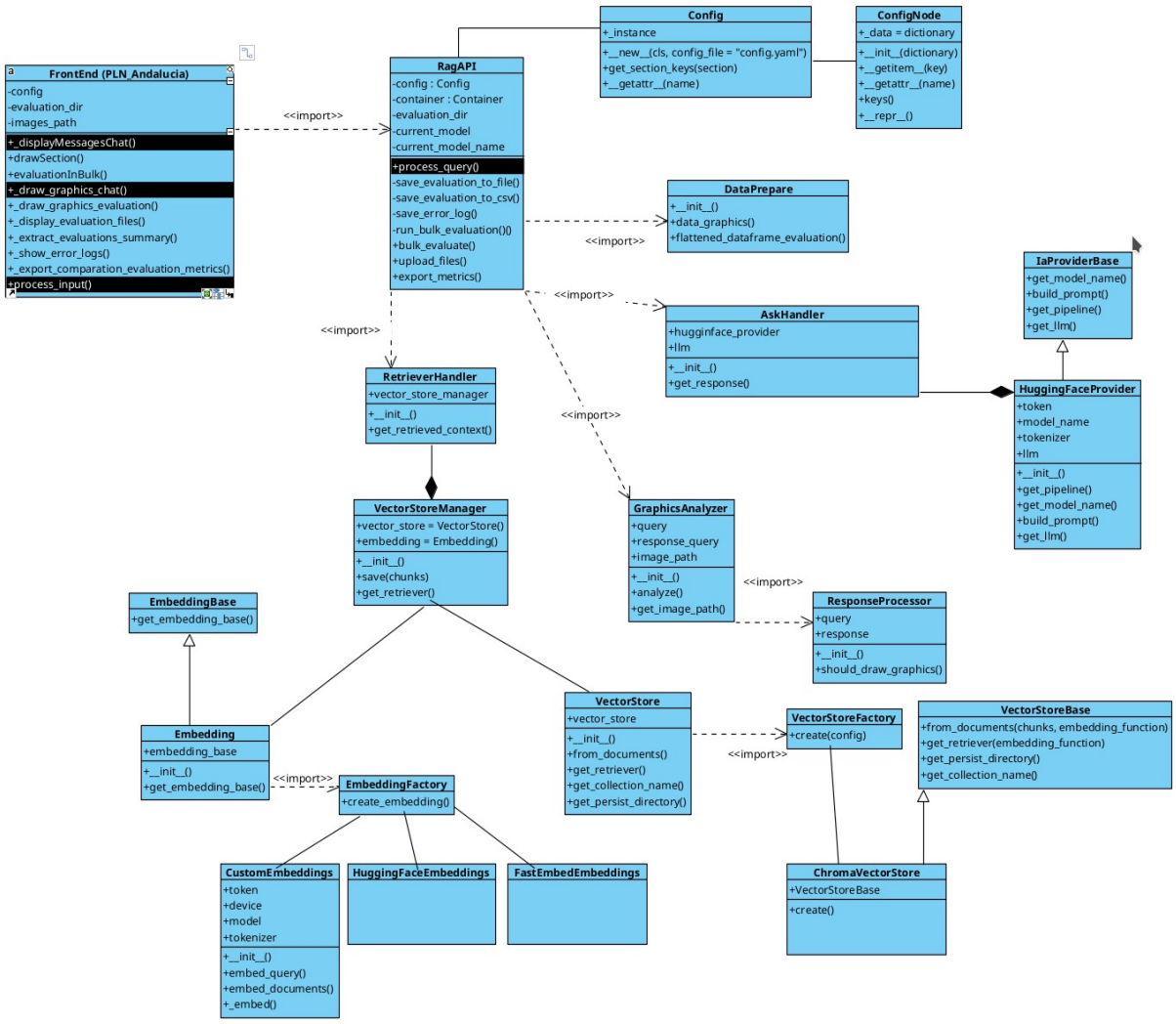


Figura 5.5: Diagrama de Clases: Respuesta RAG - Usuario.

Figura 5.5 aparece el diagrama en el que hay que destacar la flexibilidad del código añadiendo el patrón **Factory Method** en la creación del Embedding (Clase Embedding). Con ello, se asegura la configuración dinámica, desde el fichero de configuración .yaml del sistema, indicando qué tipo de embedding se utilizará en el sistema tanto para iniciar el sistema como para importar la base de datos de conocimiento (clase EmbeddingFactory). Es decir, el usuario podría en un simple cambio de configuración, elegir cualquier tipo de embedding de HuggingFace, utilizar una customizado desde

cualquier modelo, o utilizar un modelo desde Fastembedding. De la misma manera, se ha diseñado otro patrón de este tipo para la utilización de la base de datos vectorial. En concreto es la clase VectorStoreFactory que está diseñado una encapsulación o fachada de Chroma en la clase ChromaVectorStore. De igual manera está también parametrizado en el fichero de configuración .yaml. Por lo que fácilmente se podría utilizar otra base de datos vectorial, como FAISS, sin apenas esfuerzo y sin cambiar implementación existente. Destacar que en ambos ejemplos se está utilizando una interface adaptada a cada caso(VectorStoreBase y EmbeddingBase). Por otro lado hay que destacar el gran nivel de granularidad y de delegación de responsabilidades plasmada en la gran variedad de clases (AsKHandler, RetrievedHandler, VectorStoreManager, GraphicsAnalyzer, el recubrimento para los modelos de huggingface llamada la clase HuggingFaceProvider etc.) En la figura Figura 5.6 muestra los mismos patrones

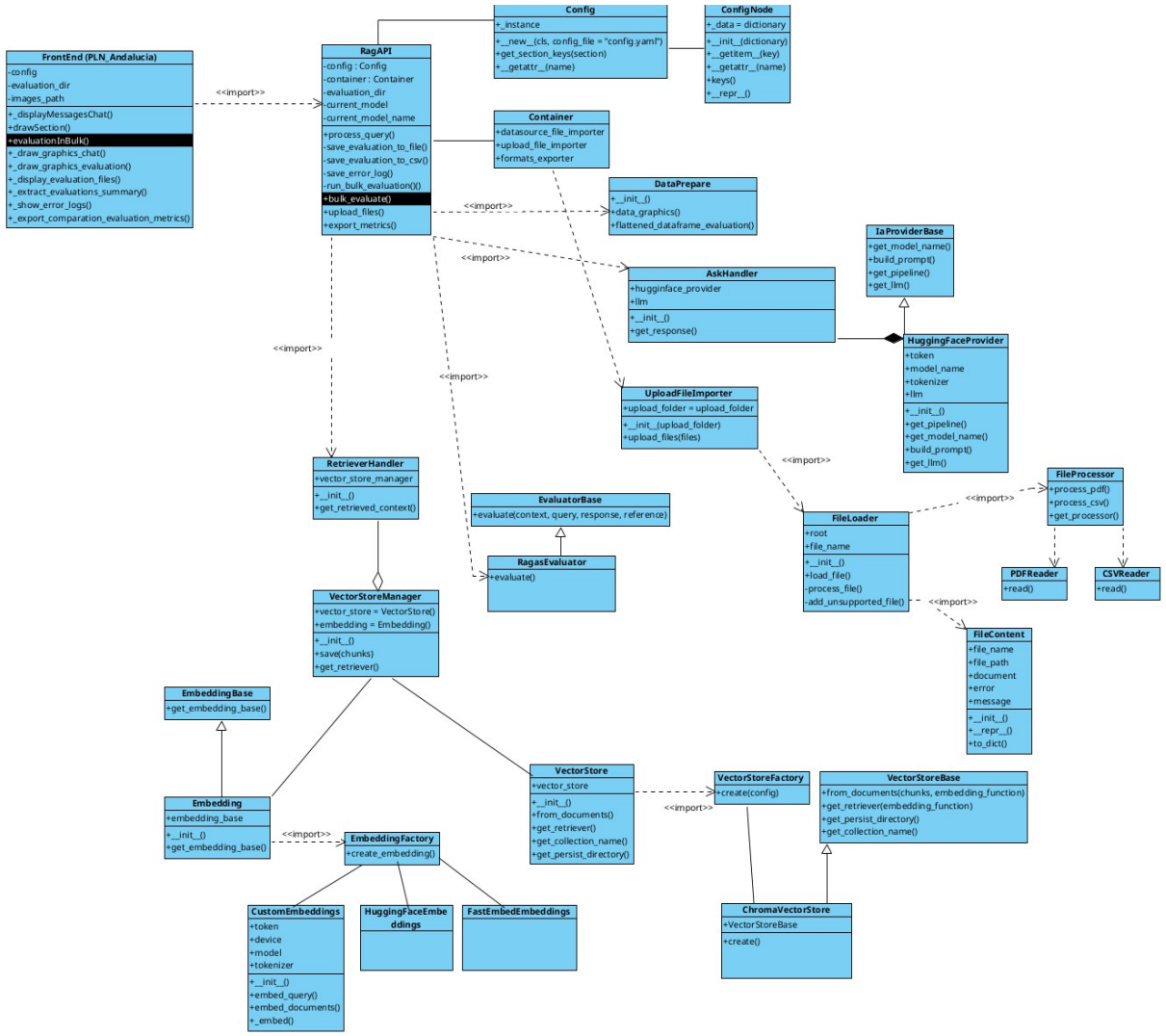


Figura 5.6: Diagrama de Clases: Ejecución de la evaluación de Modelos - Usuario.

Factory Method en las mismas clases pero añade un nuevo patrón llamado Patrón Fachada (Facade Pattern) en la que dinámicamente analizando la extensión del fichero

elige que Reader tiene que utilizar(PDFReader o CSVReader). Esta clase es la clase FileProcessor. Con este diseño se podría añadir cualquier otro tipo de reader de una manera fácil y sencilla sin comprometer el diseño del sistema. Destacar también, igual que en Figura 5.5, el gran nivel de granularidad y de delegación de responsabilidades en la gran cantidad de clases y el uso de interfaces para poder asegurar un diseño unificado y entre otros aspectos, añadir nuevos requerimientos sin modificar el codigo existente **Principio de Abierto/Cerrado (SOLID)**. Véase por ejemplo en la clase RagasEvaluator con su interfaz EvaluatorBase. Por otro lado, mencionar a efectos académicos utilizado en este proyecto, el utilizar un contenedor de servicios para utilizar el servicio UploadFileImporter para importar ficheros .csv. Con ello se pretende con este servicio y cualquier servicio del contenedor de dependencias, poder gestionar eficientemente las dependencias, mejorar la reutilización del código y reducir el acoplamiento del sistema. En definitiva, afecta a la mantenibilidad del código.

En la figura Figura 5.7 todo el procesamiento lo hace el frontend a partir de los ficheros de evaluación generando en el proceso de Ejecución de la evaluación (Figura 5.6).

En la figura Figura 5.8 aparece de nuevo el uso del patrón Strategy y del contenedor de servicios para utilizar la clase FormatsExporter. Con ella, dependiendo de la selección desde la interfaz de usuario del formato de la exportación, extrae la información para su exportación y posterior descarga.

En la figura Figura 5.9 muestra como el usuario administrador desde la consola del sistema donde está hospedado el sistema, ejecuta un comando para importar estas leyes. Cabe destacar que se ha utilizado, a efectos académicos, una 'clase' cli.py (utilizando la librería click) puede llamar a cualquier comando creado, en este caso 'datasource_import_command'. A su vez llama al contenededor de servicios para utilizar el servicio DataSourceFileImporter. Con ello se reutiliza toda la parte del VectorStore, la parte de las clases de Embedding y las clases de la lectura de ficheros FileLoader utilizado en el proceso de ejecución de la evaluación de los modelos (Figura 5.6).

Por último indicar, que el **modelo RAG (Retrieval-Augmented Generation)** ha revolucionado la extracción de información en el Procesamiento del Lenguaje Natural (PLN) al combinar recuperación y generación de texto en un mismo proceso. A diferencia de los enfoques tradicionales, que dependían de reglas estáticas o algoritmos de clasificación, RAG recupera información relevante de una base de conocimiento antes de generar una respuesta, garantizando mayor coherencia y precisión. En el contexto de la extracción de definiciones, este modelo permite consolidar fragmentos dispersos en una única definición estructurada, minimizando errores y evitando respuestas fa-

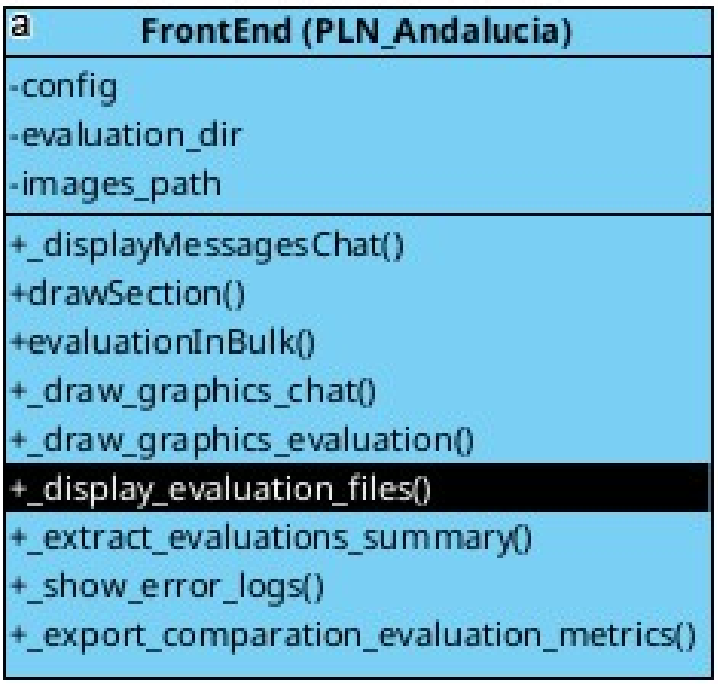


Figura 5.7: Diagrama de Clases: Construir la evaluación de Modelos - Usuario.

bricadas. Su capacidad para contextualizar la información mejora significativamente la fidelidad y exactitud de las respuestas generadas por nuestros modelos del lenguaje, mejorando la calidad de la información presentada y facilitando su uso en tareas de análisis y toma de decisiones.

5.2.4. Diseño de la salida

El diseño de salida del sistema está orientado a ofrecer una presentación estructurada y comprensible de la información recuperada y generada, tanto en las respuestas obtenidas mediante RAG como en la evaluación de modelos. Para garantizar una experiencia de usuario óptima, el sistema incorporará una interfaz web moderna y adap-

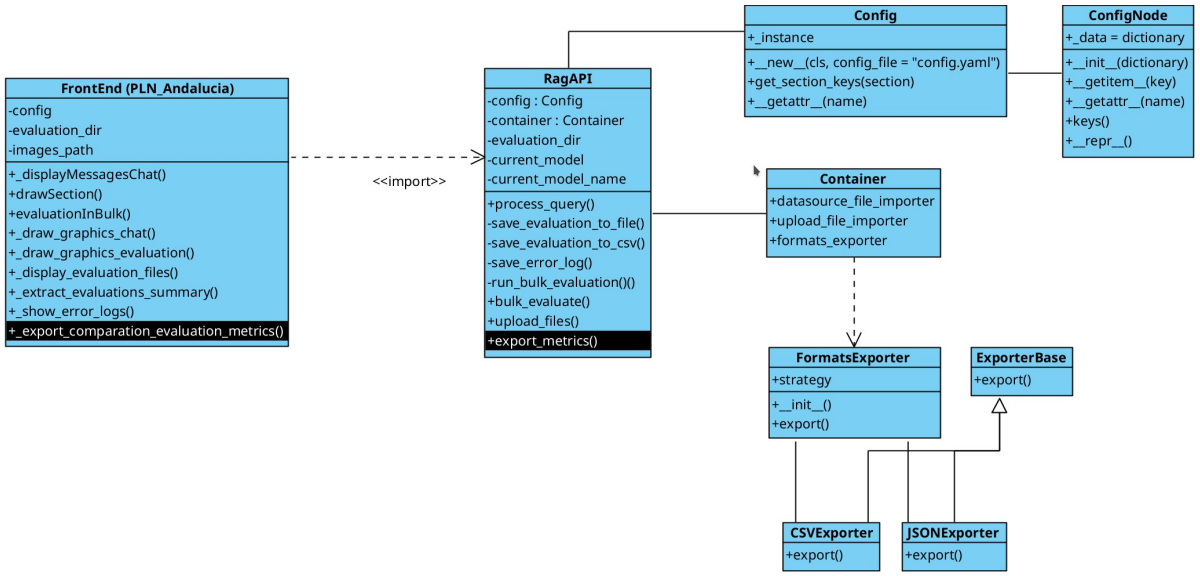


Figura 5.8: Diagrama de Clases: Exportación de las métricas de la evaluación de Modelos - Usuario.

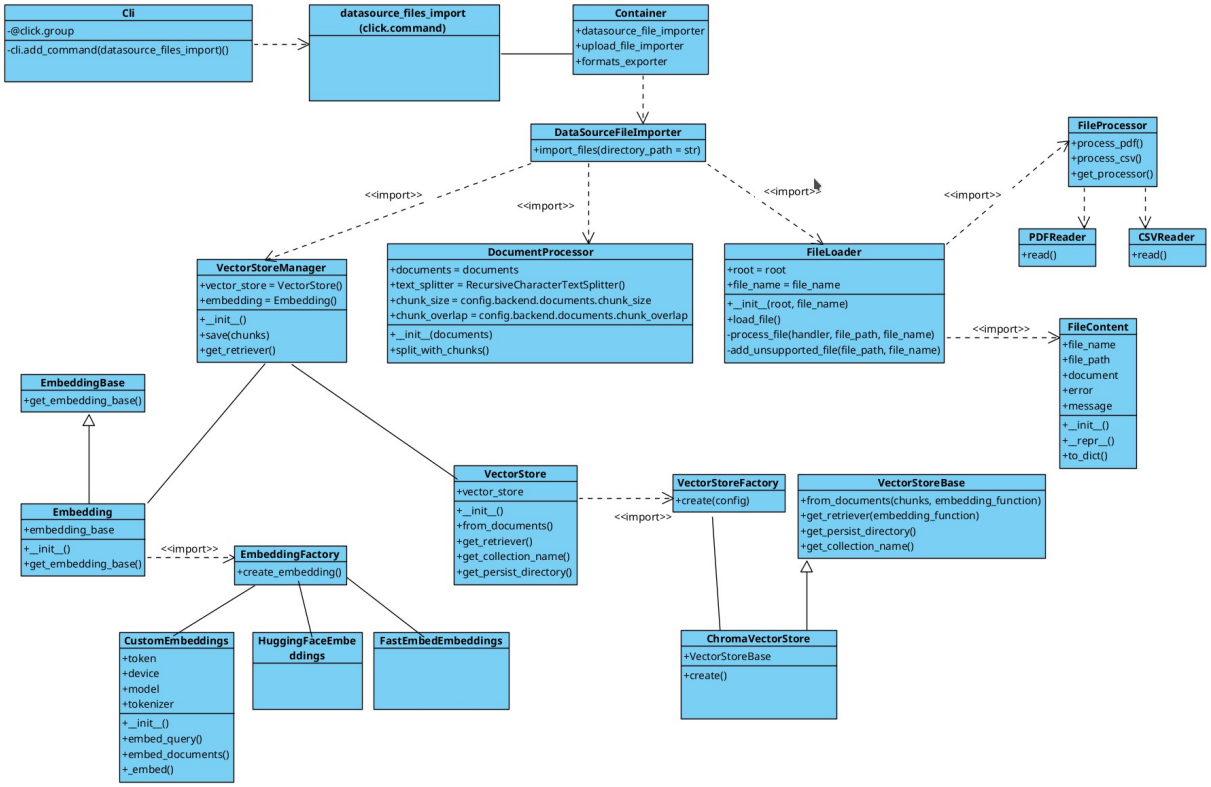


Figura 5.9: Diagrama de Clases: Importación de leyes del presupuesto - Administrador.

table, enfocada en una navegación intuitiva, facilitando la interacción con las distintas funcionalidades de manera sencilla. Desde la página principal, los usuarios podrán acceder a todas las herramientas disponibles, asegurando una navegación fluida y eficiente. La organización de los distintos componentes y su interconexión se reflejan en la (Figura 5.10), proporcionando una visión clara de la arquitectura de la interfaz.

Además, la página de inicio actuará como el nodo central del sistema, permitiendo a los usuarios volver rápidamente a esta sección en cualquier momento. Además, se garantiza la exportación de datos en formatos estándar (CSV, JSON), facilitando su integración con otras herramientas de terceros para su análisis.

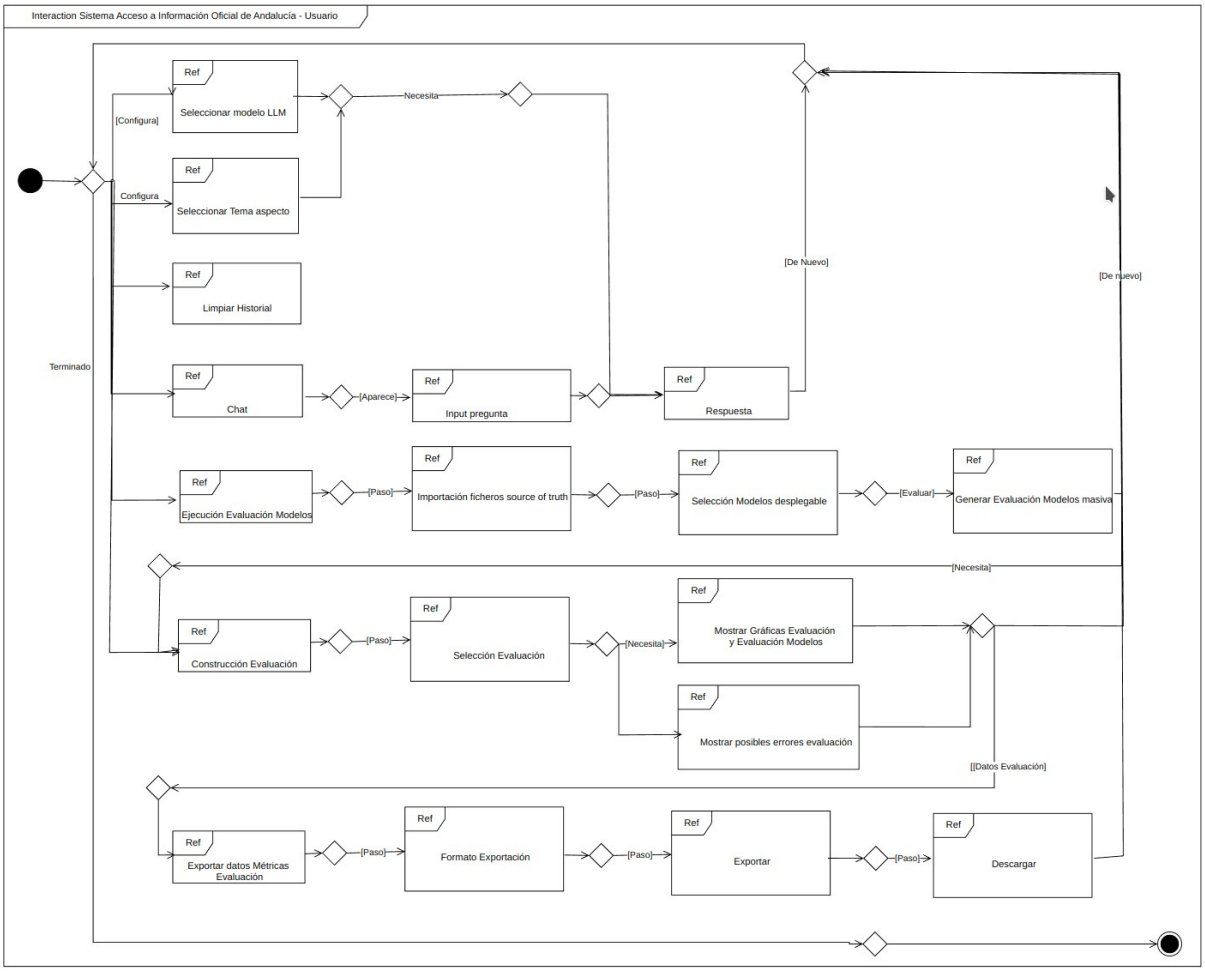


Figura 5.10: Diagrama de interacción de la interfaz de usuario.

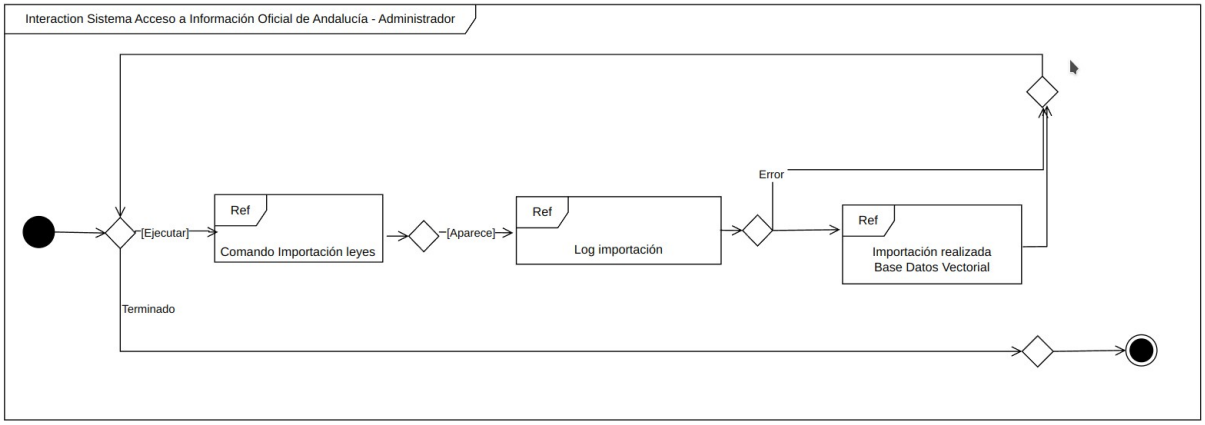


Figura 5.11: Diagrama de interacción de la interfaz de administrador.

El diseño de la interfaz principal del sistema ha sido estructurado con el objetivo de proporcionar una experiencia de usuario intuitiva y eficiente. Se presenta una disposición clara y organizada, en la que podemos destacar la parte superior, un encabezado destaca el propósito del sistema, acompañado de un logotipo representativo, asegurando identidad visual y reconocimiento por parte del usuario. Justo debajo, una barra de navegación horizontal (o 'tabs') que agrupa las principales funcionalidades, permitiendo un acceso rápido a las opciones clave sin sobrecargar la pantalla ni afectar la usabilidad. En el panel lateral izquierdo, se han dispuesto controles interactivos que

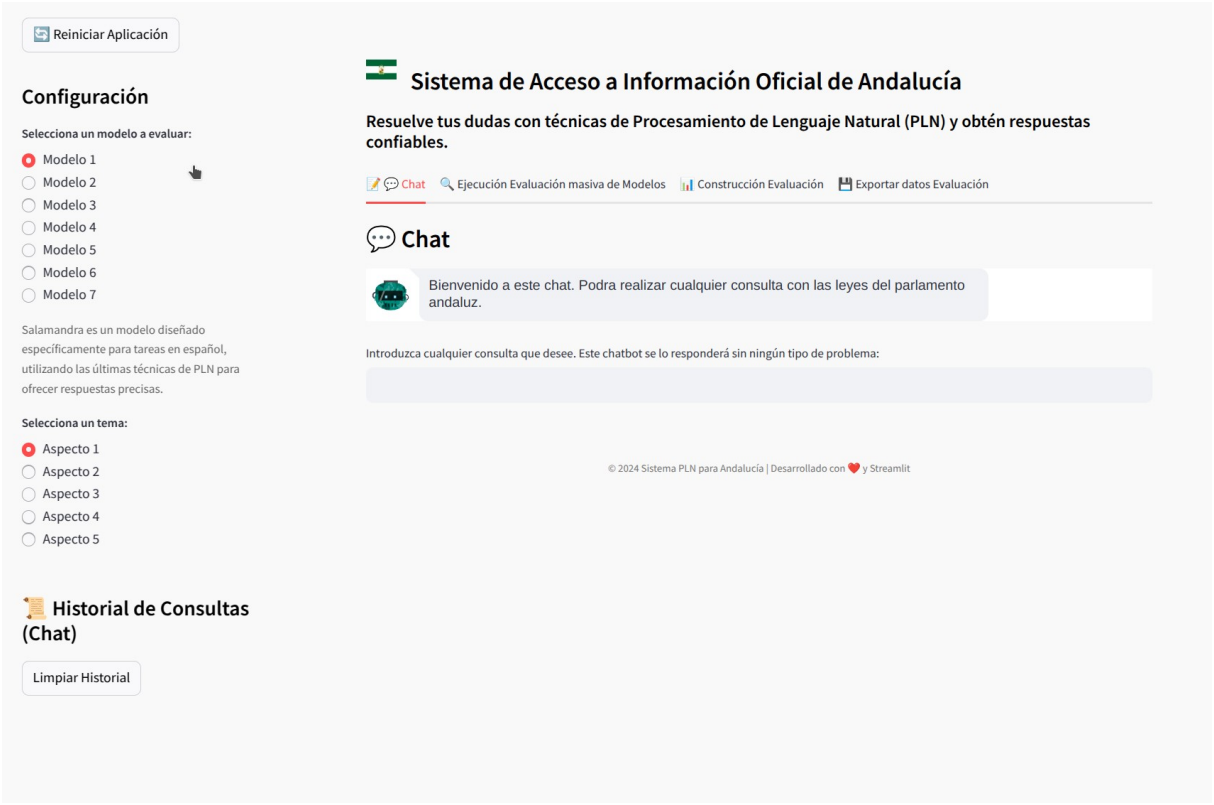


Figura 5.12: Mockup de la Página principal de la interfaz de usuario.

permiten la configuración personalizada del sistema. Entre estos, se incluye la selección de modelos de lenguaje disponibles para evaluación, así como la posibilidad de personalizar la apariencia visual mediante distintos temas. Además, se ofrece una sección de historial de consultas, lo que facilita el seguimiento de interacciones previas con el sistema. Esta distribución garantiza que los usuarios puedan gestionar sus preferencias de manera accesible sin interferir con el área principal de interacción.

El núcleo funcional de la página se encuentra en la sección central, donde se despliega el entorno de chat en la primera opción de la barra de navegación horizontal, diseñado para recibir y procesar consultas en lenguaje natural. La caja de texto permite la introducción de preguntas, mientras que el chatbot responde de forma estructurada, asegurando claridad y comprensión en la información proporcionada. Adicionalmente,

el sistema cuenta con un mecanismo de gestión de evaluaciones de modelos, accesible desde la barra de navegación superior en la siguiente opción, lo que refuerza la capacidad del sistema para analizar y comparar diferentes modelos de procesamiento del lenguaje natural aplicados a la temática del sistema.

Además de las funcionalidades principales, la interfaz incluye dos opciones adicionales accesibles desde la barra de navegación horizontal. La primera permite la construcción y visualización de los resultados obtenidos en la evaluación de modelos en cualquier momento y de la evaluación que desee, proporcionando representaciones gráficas que facilitan el análisis comparativo. La segunda opción habilita la exportación de los datos de la última evaluación en distintos formatos, asegurando compatibilidad con herramientas externas de terceros y ampliando las posibilidades de integración con otros sistemas de procesamiento y análisis de información.

Gracias a esta arquitectura de interfaz, se garantiza una navegación fluida, accesibilidad mejorada y una presentación clara de los datos esenciales, permitiendo a los usuarios gestionar y comprender la información de manera eficiente.

Se ha primado en generar una primera versión o iteración del sistema (MVP) en la que sea el administrador, como dueño de la información y que posee el mecanismo de seguridad e integridad de los datos, la persona que haga el proceso de importación de leyes del presupuesto.

Además para ello, se ha optado en este caso por la ejecución de un comando o script que, si los requerimientos cambiaran, pueda ser lanzando con un cron cada cierto tiempo de frecuencia para olvidarse de lanzarlo siempre que se necesite manualmente. Solo sería necesario incluir las leyes o las nuevas que se deseen, dentro de una carpeta definida por el administrador. Con ello también se deja abierta la puerta para una última iteración del sistema, la aplicación de un futuro proceso de captura de la base de datos del conocimiento con técnicas como Scrapping o WebScrapping, que podría hacerse de forma automática. En definitiva, este es el primer paso en el diseño dado para que el sistema de importación del corpus de documentos pueda construirse del todo automático.

5.3. Tecnologías Utilizadas

El desarrollo del presente proyecto ha requerido la integración de diversas tecnologías que han facilitado la implementación de un sistema de **Recuperación Aumentada por Generación (RAG)** basado en las leyes presupuestos de la Junta de Andalucía

y su posterior **evaluación con modelos del lenguaje (LLM)**. La selección de estas herramientas se ha realizado en función de su capacidad para procesar información de **manera eficiente**, **gestionar datos vectoriales** y proporcionar una **interfaz intuitiva** para la interacción con el usuario y con un administrador del sistema. Véase el siguiente resumen:

Tipo de Tecnología	**Tecnologías**
Lenguaje de Programación	Python
Desarrollo Web e Interfaz de Usuario	Streamlit, Flask
Procesamiento del Lenguaje Natural (PLN)	LangChain, Hugging Face
Bases de Datos y Almacenamiento Vectorial	ChromaDB
Optimización del Rendimiento y Ejecución Asíncrona	Threading, Bitsandbytes
Inyección de Dependencias y Gestión Modular	Dependency Injector
Automatización y Ejecución desde la Línea de Comandos	Click
Análisis, Evaluación y Visualización de Datos	Ragas, Seaborn, Altair, Matplotlib
Manejo y Procesamiento de Documentos	PyPDFLoader

Tabla. 5.1: Tecnologías utilizadas en el proyecto agrupadas por categoría

A continuación, se describe en profundidad cada una de las herramientas utilizadas, su historia, evolución y **ventajas dentro del proyecto.**

5.3.1. Python

Python es un lenguaje de programación interpretado de alto nivel, diseñado por Guido van Rossum en 1991 en el *Centro de Matemáticas y Computación* (CWI) en los Países Bajos [32]. Van Rossum creó Python como una alternativa a ABC, con una sintaxis más limpia y accesible, influenciada por C, Perl y Lisp. Su primera versión pública, Python 1.0, se lanzó en 1994, incluyendo características como clases, manejo de excepciones y módulos.

Durante su evolución, Python ha pasado por diversas versiones significativas. Python 2.0 (lanzado en 2000) introdujo mejoras en la gestión de memoria y la recolección de basura. Sin embargo, no fue hasta Python 3.0 (lanzado en 2008) que se realizaron cambios fundamentales en la sintaxis y compatibilidad, eliminando redundancias [33]. Actualmente, Python es mantenido por la *Python Software Foundation* y es uno de los lenguajes más utilizados en ciencia de datos e inteligencia artificial [34].

Python ha sido ampliamente adoptado en múltiples disciplinas, desde el análisis de datos hasta el desarrollo de inteligencia artificial. Su compatibilidad con librerías como

NumPy, Pandas y TensorFlow ha impulsado su crecimiento en el ámbito del *machine learning* y PLN [35]. Además, su sintaxis intuitiva lo hace ideal para desarrolladores y académicos.

En este proyecto, Python ha sido elegido por su **versatilidad y compatibilidad** con herramientas avanzadas como **LangChain, Hugging Face, ChromaDB y Streamlit**. Su ecosistema permite la integración fluida de modelos de PLN, almacenamiento de datos vectoriales y visualización de resultados, asegurando un flujo de trabajo eficiente y escalable.

5.3.2. Streamlit

Streamlit es un *framework* de código abierto para el desarrollo de aplicaciones web en Python, creado en 2019 por Adrien Treuille, Thiago Teixeira y Amanda Kelly. La idea nació cuando Treuille, exprofesor de la Universidad Carnegie Mellon, observó la necesidad de un entorno simple y directo para la creación de interfaces web en ciencia de datos sin conocimientos avanzados en *frontend*.

Desde su lanzamiento, Streamlit ha experimentado un crecimiento notable en la comunidad de ciencia de datos, agregando compatibilidad con bibliotecas como *Pandas, Matplotlib y Altair* [36]. En 2022, la empresa *Snowflake* adquirió Streamlit, lo que impulsó su uso en entornos empresariales y su optimización para *big data* [37].

Uno de los aspectos más destacados de Streamlit es su capacidad para transformar scripts de Python en aplicaciones interactivas con solo unas pocas líneas de código. Su arquitectura permite la actualización automática de componentes y una integración sin esfuerzo con modelos de inteligencia artificial.

Es por ello para este proyecto, Streamlit es utilizado para **desarrollar una interfaz web interactiva** donde los usuarios pueden consultar y visualizar los datos procesados. Su **simplicidad y flexibilidad en el código** permiten mostrar resultados de forma intuitiva, optimizando la experiencia de usuario y facilitando la interpretación de los resultados generados por el sistema de recuperación de información.

5.3.3. Flask

Flask es un *microframework* ligero para el desarrollo de aplicaciones web en Python, creado en 2010 por Armin Ronacher. Flask nació como un experimento en el proyec-

to *Pocoo*, pero rápidamente ganó popularidad debido a su sencillez y flexibilidad. Se diseñó para ser minimalista, permitiendo a los desarrolladores elegir sus propias herramientas sin restricciones de arquitectura.

Desde su lanzamiento, Flask ha evolucionado con la incorporación de extensiones para manejo de bases de datos, autenticación y creación de APIs RESTful [38]. Su comunidad de desarrollo ha crecido significativamente, siendo utilizado en proyectos de alto rendimiento como Reddit y LinkedIn.

Flask es conocido por su enfoque modular y su compatibilidad con diversas bibliotecas. A diferencia de frameworks más pesados como Django, Flask ofrece una mayor flexibilidad y control sobre la estructura de las aplicaciones.

En este proyecto, Flask ha sido seleccionado como herramienta para codificar el **backend** y con ello **desacoplar la parte del cliente-servidor e independizar ambos** (por un lado la parte del frontend con Streamlit y por otro lado la parte del backend bajo esta tecnología). Su ligereza y facilidad de integración con otras herramientas aseguran un rendimiento óptimo en el procesamiento de peticiones. Además, hace que el sistema sea **robusto y escalable** cumpliendo unos de los requerimientos No funcionales ya citados en este proyecto.

5.3.4. LangChain

LangChain es un *framework* especializado en la integración de modelos de lenguaje con bases de datos y sistemas de recuperación de información. Fue desarrollado por *Harrison Chase* en 2022 con el objetivo de facilitar el desarrollo de aplicaciones basadas en inteligencia artificial generativa [39]. Desde su lanzamiento, LangChain ha experimentado un crecimiento acelerado, siendo adoptado por la comunidad de inteligencia artificial para construir soluciones de cualquier tipo y en particular RAG (*Retrieval-Augmented Generation*). Su evolución ha incorporado compatibilidad con modelos como GPT y sistemas de almacenamiento vectorial [**?**].

LangChain es esencial en la automatización de flujos conversacionales y análisis de documentos. Su enfoque modular permite mejorar la precisión de respuestas en sistemas de recuperación de información.

Para la realización de este proyecto, LangChain es fundamental y esencial ya que permite la **conexión eficiente entre modelos de lenguaje y bases de datos vectoriales**, optimizando la recuperación de información y asegurando resultados precisos

en consultas textuales.

5.3.5. Hugging Face

Hugging Face fue fundado en 2016 por Clément Delangue, Julien Chaumond y Thomas Wolf con la idea inicial de crear un chatbot conversacional [40]. Sin embargo, la empresa cambió su enfoque hacia el desarrollo de modelos de lenguaje y procesamiento del lenguaje natural (PLN), convirtiéndose en una referencia en la industria.

El impacto de Hugging Face se consolidó con la introducción de la biblioteca `Transformers` en 2019, facilitando el acceso a modelos de inteligencia artificial como BERT y GPT [41]. Actualmente, su plataforma permite el uso e implementación de modelos de lenguaje avanzados con facilidad.

Su mayor fortaleza es su ecosistema y su función de repositorio, que incluye miles de modelos y conjuntos de datos listos para su uso. Hugging Face ha popularizado el acceso a la inteligencia artificial, permitiendo que investigadores y empresas desarrollen soluciones PLN sin necesidad de entrenar modelos desde cero.

Para este proyecto, Hugging Face **proporciona los modelos de lenguaje que permiten la comprensión y generación de texto** en el sistema de recuperación de información. Su **integración con LangChain y ChromaDB** optimiza la precisión de las respuestas generadas.

5.3.6. ChromaDB

ChromaDB es una base de datos vectorial diseñada para almacenar y recuperar información semántica de manera eficiente. Fue lanzada en 2021 por *Anton Troynikov y Jeff Huber* con el objetivo de mejorar la búsqueda en entornos basados en inteligencia artificial [42].

A lo largo de su desarrollo, ChromaDB ha incorporado optimizaciones en la indexación de datos y la consulta de información vectorial, convirtiéndose en una alternativa eficiente frente a bases de datos tradicionales.

Su arquitectura está diseñada para manejar grandes volúmenes de datos de texto, lo que la hace ideal para aplicaciones de PLN y búsqueda semántica.

En este proyecto, ChromaDB es utilizada como base de datos vectorial, permitien-

do almacenar representaciones numéricas de documentos y **mejorar la recuperación de información basada en embeddings**.

5.3.7. Threading

El módulo `threading` de Python fue introducido en la versión 1.5.2 de 1999 como una solución para ejecutar múltiples tareas de forma concurrente dentro de un solo proceso [43]. Su objetivo inicial era mejorar la eficiencia de programas que requerían la ejecución de múltiples tareas sin bloquear el sistema.

Con el paso del tiempo, se han añadido mejoras en la gestión de subprocesos, permitiendo una mejor administración de recursos en sistemas con múltiples núcleos. Aunque Python tiene limitaciones con el *Global Interpreter Lock* (GIL), `threading` sigue siendo útil para tareas I/O intensivas.

En este proyecto, `threading` se emplea para **ejecutar tareas en paralelo,** permitiendo que la generación de la **evaluación de modelos del lenguaje se realice en segundo plano.** Esto garantiza que la interfaz permanezca operativa, permitiendo al usuario continuar interactuando con el sistema sin interrupciones. De esta manera, se **evita la percepción de bloqueo mientras se lleva a cabo el proceso de evaluación,** mejorando la experiencia de uso y la fluidez del sistema.

5.3.8. Bitsandbytes

`bitsandbytes` es una biblioteca desarrollada por Tim Dettmers en 2022, diseñada para mejorar la eficiencia del entrenamiento de modelos de aprendizaje profundo mediante la reducción del uso de memoria [44]. Su enfoque en la cuantización de modelos ha permitido mejorar el rendimiento en hardware limitado.

Desde su creación, `bitsandbytes` ha sido adoptado en frameworks como Hugging Face y PyTorch, permitiendo la ejecución de modelos de IA con menor consumo de recursos. Su uso ha sido clave en la democratización del acceso a modelos de lenguaje avanzados.

La mayor ventaja de esta herramienta es la reducción del tamaño de los modelos sin afectar significativamente su rendimiento, facilitando la ejecución de modelos en entornos con recursos limitados.

En este proyecto, `bitsandbytes` se emplea para **optimizar la carga y procesamiento de modelos de lenguaje, reduciendo el uso de memoria y acelerando las operaciones.**

5.3.9. Dependency Injector

El patrón de **inyección de dependencias** ha sido ampliamente utilizado en el desarrollo de software para mejorar la modularidad y escalabilidad de los sistemas. Dependency Injector fue introducido por Roman Mogylatov en 2016 como una solución para la gestión eficiente de dependencias en Python [45].

Con el tiempo, esta biblioteca ha mejorado su soporte para arquitecturas de microservicios y ha sido adoptada en proyectos de gran escala. Su implementación ha facilitado la separación de componentes y ha permitido reducir el acoplamiento entre módulos.

La principal ventaja de Dependency Injector es su capacidad para mejorar la mantenibilidad del código y facilitar la reutilización de componentes sin dependencias rígidas.

En este proyecto, se ha utilizado para **estructurar la aplicación de manera modular**, asegurando una mejor organización del código y facilitando la **escalabilidad del sistema**. En concreto ha sido aplicado en la creación de los servicios de importación de ficheros de las leyes del presupuesto, en la exportación de las evaluaciones de modelos por formatos y en el servicio de subida de ficheros source of truth como paso previo a la evaluación masiva.

5.3.10. Click

`Click` (*Command Line Interface Creation Kit*) es una biblioteca de Python desarrollada por Armin Ronacher en 2014, con el objetivo de facilitar la creación de interfaces de línea de comandos (**CLI**) [46]. Su diseño modular permite construir herramientas de automatización de forma sencilla y eficiente.

Desde su lanzamiento, Click ha sido adoptado en múltiples proyectos debido a su simplicidad y compatibilidad con otros frameworks como Flask y Django. Ha evolucionado con mejoras en la gestión de argumentos y opciones interactivas.

La ventaja principal de Click es su facilidad de uso y flexibilidad para crear herramientas CLI sin necesidad de estructuras complejas.

Para este proyecto, Click ha sido utilizado para la **automatización del proceso en consola como la importación de la base de datos de conocimiento** a *modo académico* de similares características al gestor de comandos implementado por el framework Symfony de PHP [47] [48].

5.3.11. Ragas

En el desarrollo de este sistema, el análisis y la evaluación de datos juegan un papel crucial para garantizar la precisión y utilidad de las respuestas generadas.

Las métricas tradicionales, como *BLEU* y *ROUGE*, ampliamente utilizadas en el procesamiento del lenguaje natural, no suelen captar los requisitos de las aplicaciones RAG. Veámoslo en detalle brevemente en este tabla:

Métrica Tradicional	**Descripción**	**Limitaciones en metodologías RAG**
BLEU [49]	Métrica basada en precisión que evalúa la superposición de n-gramas(secuencias de palabras) entre la respuesta generada y una respuesta de referencia.	Realiza una comparación superficial del texto sin evaluar la corrección, relevancia o coherencia con la información recuperada, ignorando el contexto y la veracidad y coherencia de una respuesta en relación con las fuentes de información utilizadas.
ROUGE [50]	Métrica que mide la superposición de n-gramas, secuencias de palabras y pares de palabras entre la respuesta generada y los textos de referencia.	Prioriza el 'recuerdo' sin evaluar la precisión o relevancia de la información recuperada, ignorando el contexto y la evidencia proporcionada por los documentos.

Tabla. 5.2: Comparativa Modelos Evaluación

Aquí es donde entra en juego esta metodología de evaluación **RAGAS (Retrieval-Augmented Generation Assessment Suite).** Es por ello que está diseñada para *permitir analizar la calidad de las respuestas generadas en un sistema RAG, midiendo relevancia, fidelidad y coherencia con respecto al contexto recuperado.* Además, evalúa la precisión y cobertura del contexto recuperado en relación con la consulta.

Para ello, las respuestas generadas se comparan con un conjunto de referencia (source of truth), compuesto por pares de preguntas y respuestas. Este enfoque permite en el proyecto facilitar la evaluación objetiva de los modelos LLM, permitiendo seleccionar el más preciso y relevante.

5.3.12. Seaborn, Altair y Matplotlib

Las bibliotecas `Seaborn`, `Altair` y `Matplotlib` han sido ampliamente utilizadas en el ámbito del análisis de datos para la generación de gráficos visualmente atractivos y comprensibles. `Seaborn`, creada en 2012 por Michael Waskom, se basa en Matplotlib y facilita la generación de gráficos estadísticos con una sintaxis simplificada. `Altair`, por otro lado, destaca por su enfoque declarativo, permitiendo la creación de gráficos interactivos sin necesidad de código complejo. Finalmente, `Matplotlib`, desarrollada por John D. Hunter en 2003, es la biblioteca más flexible y potente para la generación de gráficos en Python, proporcionando opciones avanzadas de personalización.

Además de estas herramientas de visualización, `NumPy` es una biblioteca esencial en la computación numérica en Python. Creada en 2006 por Travis Oliphant, ofrece estructuras de datos optimizadas para manipular matrices y realizar cálculos de alto rendimiento. Su integración con otras bibliotecas como `Pandas`, `SciPy` y `TensorFlow` ha permitido su consolidación en disciplinas como inteligencia artificial, análisis de datos y modelado científico.

En este proyecto, se han empleado estas bibliotecas para **facilitar la representación de datos y optimizar el procesamiento numérico.** `Seaborn` se ha utilizado para visualizar los resultados de la evaluación de modelos, `Altair` para la generación de gráficos dinámicos en la interfaz de usuario, `Matplotlib` para la creación de gráficos estáticos y `NumPy` para la manipulación eficiente de datos y cálculos vectoriales dentro del sistema.

5.3.13. PyPDFLoader

`PyPDFLoader` es una herramienta especializada en la extracción de texto desde documentos PDF, desarrollada como parte del ecosistema de bibliotecas de **LangChain** en 2023 [51]. Su propósito es facilitar la conversión de archivos PDF en datos estructurados para su posterior procesamiento en modelos de lenguaje.

Desde su lanzamiento, PyPDFLoader ha evolucionado con mejoras en la extracción de texto, permitiendo la segmentación de documentos en fragmentos estructurados que pueden ser utilizados para búsqueda semántica y análisis de contenido [52]. Su integración con otras herramientas de PLN ha incrementado su adopción en el procesamiento de documentos legales y científicos.

Su principal ventaja es la capacidad de procesar documentos en múltiples formatos

y adaptarse a distintos tipos de estructuras textuales, permitiendo una recuperación precisa de información relevante [53].

Para este proyecto, PyPDFLoader es utilizado para **extraer información de documentos oficiales en formato PDF en el proceso de importación de la base de conocimiento, permitiendo su conversión a embeddings para la posterior consulta y recuperación de datos mediante el modelo de RAG.**

5.3.14. Conclusión

El uso de estas tecnologías ha permitido la construcción de un sistema altamente eficiente y escalable, capaz de gestionar grandes volúmenes de datos textuales y realizar búsquedas semánticas avanzadas. La combinación de herramientas de desarrollo web, procesamiento del lenguaje natural y bases de datos vectoriales ha proporcionado una solución robusta para la recuperación de información dentro del dominio de las leyes de presupuestos de la Junta de Andalucía.

Capítulo 6

IMPLEMENTACIÓN

La implementación del sistema representa la materialización de la arquitectura y el diseño previamente establecidos, integrando cada componente en un entorno funcional y eficiente. En esta fase, se detallan los procesos técnicos llevados a cabo para construir un sistema robusto basado en la metodología Retrieval-Augmented Generation (RAG) y un sistema de evaluación de modelos referido a esta temática, optimizando la recuperación y generación de información relevante. A lo largo de este capítulo, se explicará la transformación de los documentos de leyes de presupuestos en una base de conocimiento estructurada, su inserción en una base de datos vectorial, la posterior integración con modelos de lenguaje avanzados para garantizar respuestas precisas y contextualizadas y la posterior evaluación de estos modelos en este contexto.

En primer lugar, se abordará el procesamiento e importación de datos, una fase crítica en la que los documentos son transformados en representaciones vectoriales optimizadas para una recuperación eficiente. Este proceso facilita la consulta semántica y mejora la precisión del sistema. Posteriormente, se presentará la implementación del sistema en forma de chatbot con el asistente generando respuestas a partir de una pregunta formulada por el usuario. En esta parte se describirá la selección y ajuste del modelo de lenguaje más adecuado y todo su funcionamiento relacionado. También se presentará la implementación en cómo se genera el sistema de evalución en segundo plano y cómo poder recuperarlo para poder construir o visualizar el contenido de cada evaluación en cualquier momento. Por último, se pondrá de manifiesto la exportación a varios formatos de ficheros de la evaluación construida para integrar este sistema con herramientas de terceros.

Destacar que se ha seguido **buenas prácticas en programación** mejorando la

legibilidad, mantenibilidad y eficiencia del código, facilitando su escalabilidad y colaboración en equipos. También se han usado **patrones de diseño** que ayudan a estructurar soluciones reutilizables como Fachada (Facade) que simplifica el acceso a sistemas complejos proporcionando una interfaz única, mientras que Factory centraliza la creación de objetos, promoviendo la flexibilidad y el principio de inversión de dependencias.

Finalmente, se expondrá la construcción e integración del sistema cliente-servidor, describiendo el desarrollo de una API REST que actúa como intermediaria entre el modelo de lenguaje y la interfaz de usuario. Por último, se explicará la implementación de la interfaz web, diseñada para ofrecer una experiencia intuitiva y accesible, permitiendo a los usuarios interactuar de manera fluida con el sistema. Con esta estructura modular, se asegura un sistema escalable y adaptable a futuras mejoras, optimizando el acceso y análisis de la información oficial de la Junta de Andalucía.

6.1. Inserción y procesamiento de datos

Como ya se especificó en el diseño, la base de conocimiento consta de todos los documentos de los últimos 20 años de las leyes del presupuesto de Andalucía.

Para ello, se ha diseñado la implementación de un sistema de automatización para la inserción de datos por parte del administrador, utilizando una interfaz basada en línea de comandos. Este enfoque permite gestionar el proceso de manera eficiente desde la consola. La tecnología empleada para este propósito es **Click**, cuya explicación detallada se encuentra en la Sección 5.3.10 Click.

Estos son los pasos de ejecución:

1. **Inicialización del Proceso de Importación:** Se utiliza el script `cli.py` que inicializa la ejecución del comando en consola importando la clase `datasource_files_import` en la ruta especificada en el comando como primer parámetro del comando.

2. **Instancia del Contenedor de Dependencias:** Se crea una instancia de Container, que gestiona las dependencias del sistema y obtiene una instancia del importador de archivos `datasource_file_importer`:

   `datasource_file_importer = container.datasource_file_importer()`. Es a partir de este servicio donde se ejecuta el proceso en sí de importación.

Algoritmo 1: Definición y Ejecución de la CLI en Python

a) Importar la librería `click`.

b) Importar el módulo de comandos `datasource_import_command`.

c) Definir el grupo de comandos CLI con la función `cli()`.

d) Agregar el comando de importación `datasource_files_import` al grupo `cli`.

e) Si el script es ejecutado directamente (`__main__`):

1) Ejecutar `cli()`.

Algoritmo 2: Importación de Ficheros desde una Ruta Específica

Input: `file_path` (Ruta del directorio con los archivos)
Output: Resumen de la importación con éxito y errores
Instanciar el contenedor de dependencias;
Obtener instancia del importador de archivos;
Ejecutar la importación con: `result = datasource_file_importer.import_files(file_path)`
Extraer información de la importación:
Calcular e Imprimir tiempo transcurrido y resumen de la importación;
if *failed_imports no está vacío* **then**
    Imprimir mensaje de errores;
    foreach *archivo en* `failed_imports` **do**
        Imprimir nombre del archivo y mensaje de error;
    end
end
Imprimir mensaje de finalización;

3. **Carga y Procesamiento de Archivos:** El proceso de carga y procesamiento de archivos comienza con la clase `DataSourceFileImporter`. Esta se encarga de crear, en caso de no existir, una carpeta denominada `processed`, donde se almacenarán los archivos una vez procesados. Posteriormente, se instancia un objeto de la clase `VectorStoreManager`, el cual gestiona tanto la base de datos vectorial como los embeddings.

   A continuación, se recorre la carpeta origen del *datasource*, que ha sido pasada como parámetro al comando de ejecución, y se contabiliza el número de archivos a importar. Para proporcionar un seguimiento visual del proceso, se utiliza la librería `tqdm`, que muestra una barra de progreso en la consola. Durante este recorrido, si un archivo ya ha sido procesado y se encuentra en la carpeta `processed`, se omite su procesamiento. En caso contrario, se emplea el objeto `FileLoader` para determinar el formato del archivo y recuperar su contenido,

almacenándolo en la variable `file_content`.

Si la carga del archivo se ha realizado correctamente (`if not file_content.error`), se procede a su procesamiento. Para mejorar la búsqueda semántica, el archivo se divide en fragmentos (*chunks*) mediante la clase `DocumentProcessor`, de la siguiente manera: `processing_documents = DocumentProcessor(document)` `chunks = processing_documents.split_with_chunks()`

Dicha división se realiza utilizando la librería `RecursiveCharacterTextSplitter` de `langchain.text_splitter`. El tamaño de los fragmentos y el solapamiento entre ellos (`chunk_overlap`) se configuran en el archivo de configuración `config.yaml`. Finalmente, los fragmentos generados se almacenan en la base de datos vectorial mediante `vector_store`, la cual sigue un **patrón Factory** para gestionar las operaciones de almacenamiento. Una vez completado el procesamiento, el archivo original se mueve a la carpeta `processed` con el fin de evitar duplicaciones en futuras ejecuciones.

Algoritmo 3: Importación y Procesamiento de Archivos

Input: `directory_path` (Ruta del directorio con los archivos a importar)
Output: Resumen del proceso con número de archivos procesados, importaciones exitosas y errores
Inicializar carpeta `processed_directory` si no existe ;
Instanciar `VectorStoreManager` ;
Contar archivos en `directory_path` ;
Inicializar barra de progreso ;
foreach *archivo en* `directory_path` **do**
  if *archivo ya está en* `processed_directory` **then**
    Continuar con el siguiente archivo ;
  end
  Obtener ruta completa del archivo ;
  Instanciar `FileLoader` ;
  Cargar archivo con `load_file()` ;
  if *archivo cargado correctamente* **then**
    Obtener contenido del documento ;
    Instanciar `DocumentProcessor` ;
    Dividir documento en fragmentos (*chunks*) ;
    Guardar fragmentos en base de datos vectorial con `vector_store.save()` ;
    Incrementar contador de `successful_imports` ;
    Obtener ruta relativa del archivo ;
    Crear subcarpeta en `processed_directory` ;
    Mover archivo a subcarpeta correspondiente ;
  else
    Agregar archivo a `failed_imports` con mensaje de error ;
  end
  Actualizar barra de progreso ;
end
Devolver resultados: total de archivos procesados, exitosos y fallidos ;

Algoritmo 4: División de documentos en fragmentos manejables

Input: Lista de documentos
Output: Lista de fragmentos generados
Inicializar `text_splitter` con parámetros de configuración a partir de `RecursiveCharacterTextSplitter`;
Dividir documentos en fragmentos con solapamiento definido;
Retornar la lista de fragmentos;

Cabe resaltar el proceso de carga de documentos, el cual es gestionado por el manejador `FileProcessor`. Este manejador implementa el **patrón Fachada (Facade)**, permitiendo la selección dinámica del método de lectura según el tipo de archivo a procesar. Dependiendo de la extensión del fichero (`.csv` o `.pdf`), el sistema elige automáticamente el lector adecuado, delegando la extracción del contenido a la clase correspondiente, ya sea `PDFReader` para documentos en formato PDF o `CSVReader` para archivos CSV.

Algoritmo 5: Cargador y procesamiento de archivos

```
Input: Ruta del directorio y nombre del archivo
Output: Objeto con los datos procesados o error
Obtener ruta completa del archivo;
Identificar la extensión del archivo;
Buscar el procesador adecuado en la lista de handlers;
if existe un handler para el archivo then
    Procesar el archivo usando el handler;
else
    Registrar el archivo como no soportado;
end
Retornar objeto con el contenido o error;
```

Algoritmo 6: Procesamiento de archivos según su extensión

```
Input: Ruta del archivo y extensión
Output: Contenido del archivo procesado o error
switch extensión del archivo do
    case .pdf do
        Procesar archivo con PDFReader;
    end
    case .csv do
        Procesar archivo con CSVReader;
    end
    Retornar error por formato no soportado;
end
```

También a destacar la implementación de la base de datos vectorial para el almacenamiento de los *chunks* es un aspecto clave del sistema. Para gestionar este proceso, se ha desarrollado un manejador denominado `VectorStoreManager`, el cual centraliza las dependencias esenciales: la propia base de datos vectorial (`VectorStore`) y los embeddings (`Embedding`).

Estas clases actúan como una capa de abstracción que, en su interior, implementa otro patrón de diseño conocido como **Factory Pattern**. En particular, la clase `VectorStore` depende de `VectorStoreFactory`, la cual, a través de la invocación de `VectorStoreFactory.create(config.vector_store)`, selecciona

dinámicamente el tipo de base de datos vectorial a utilizar según la configuración establecida en el archivo `config.yaml`. En este caso, con el valor *chroma*, se instancia un objeto de la clase `ChromaVectorStore`, que a su vez encapsula la funcionalidad de la biblioteca `Chroma` de `LangChain` (`from langchain_chroma import Chroma`).

Dentro de esta estructura, la persistencia de los *chunks* se realiza mediante la invocación del método `from_documents`, que recibe como parámetros los fragmentos de texto y la función de embeddings correspondiente (`from_documents(chunks, embedding_function)`). Además, `VectorStore` sigue una interfaz bien definida llamada su clase `VectorStoreBase`, lo que permite la sustitución de la base de datos vectorial (por ejemplo, FAISS) simplemente implementando los métodos requeridos en una nueva clase adaptada a esa tecnología.

De manera similar, la clase `Embedding` depende de `EmbeddingFactory`, que a través de la función `VectorStoreFactory.create_embedding()` selecciona dinámicamente el tipo de embeddings a emplear según la configuración definida en `config.yaml`. Entre las opciones disponibles se encuentran *'huggingface'*, *'fastembed'* y *'CustomEmbeddings'*. Esta flexibilidad permite integrar distintos proveedores de embeddings o incluso definir una implementación personalizada. En el caso de requerir la incorporación de embeddings de OpenAI, bastaría con agregar una nueva clase de embeddings y configurar su `API_KEY` para que el sistema la reconozca automáticamente.

En este sistema, se ha seleccionado el modelo de embeddings **BAAI/bge-small-en-v1.5** de Hugging Face debido a su alto rendimiento en tareas de búsqueda semántica, optimizando la representación vectorial de textos en espacios de alta dimensionalidad. Este modelo ha sido diseñado específicamente para capturar relaciones semánticas profundas entre fragmentos de texto, lo que permite mejorar la precisión en la recuperación de información dentro de bases de datos vectoriales extensas.

Además, **BAAI/bge-small-en-v1.5** se distingue por su capacidad de mantener un equilibrio entre eficiencia computacional y calidad en la generación de embeddings, lo que resulta fundamental para aplicaciones que requieren una rápida indexación y consulta sobre grandes volúmenes de datos. Aunque su entrenamiento está mayoritariamente basado en corpus en inglés, su desempeño en contextos multilingües sigue siendo competitivo debido a la robustez de su arquitectura basada en `Transformers` [54].

La elección de este modelo responde a la necesidad de maximizar la fidelidad semántica en la recuperación de documentos, minimizando la distancia entre vectores de textos similares y facilitando el uso de técnicas de *nearest neighbor*

search (`k-NN`) optimizadas para bases de datos vectoriales como `ChromaDB`. Su integración en el sistema permite mejorar la calidad de la información recuperada sin comprometer la velocidad de procesamiento, ofreciendo así una solución eficiente y escalable para tareas de *Retrieval-Augmented Generation* (RAG). [55]

Algoritmo 7: Fábrica para la creación de diferentes tipos de almacenes vectoriales con VectorStoreFactory

Data: Configuración del almacén vectorial
Result: Instancia del almacén vectorial correspondiente
`Crear almacén vectorial(`*config*`)` Obtener tipo de almacén vectorial de la configuración;
if *store_type es 'chroma'* **then**
  return Nueva instancia de ChromaVectorStore con configuración de Chroma;
end
else
  return Error: "Tipo de base de datos vectorial desconocido
end

Algoritmo 8: Fábrica para la generación de funciones de embeddings con EmbeddingFactory

Data: Configuración del proveedor de embeddings
Result: Instancia del modelo de embeddings adecuado
`Crear función de embeddings()` Obtener configuración del backend para embeddings;
Obtener proveedor y modelo de embeddings;
switch *proveedor* **do**
  case *"huggingface"* **do**
    return Instancia de HuggingFaceEmbeddings con parámetros configurados;
  end
  case *"fastembed"* **do**
    return Instancia de FastEmbedEmbeddings con modelo configurado;
  end
  case *'custom_model'* **do**
    return Instancia de CustomEmbeddings con modelo y tokenizer;
  end
  return Error: "Proveedor de embeddings no soportado.
end

Algoritmo 9: Almacenamiento de documentos en ChromaDB

Input: Lista de documentos `chunks`, Función de embeddings `embedding_function`
Output: Booleano indicando si la operación fue exitosa
Comprobaciones de parámetros
Try Ejecutar:;
Chroma.from_documents(documents=chunks,
embedding=embedding_function,
persist_directory=self.persist_directory,
collection_name=self.collection_name,
collection_metadata='hnsw:space': 'cosine') ;
Catch Exception `error`;

6.2. Elementos de la metodología RAG

El siguiente paso es la generación de respuestas a partir de preguntas realizadas por el usuario. Estas son las fases a destacar:

6.2.1. Inicialización del sistema y renderización inicial

Desde el frontend (`pln_andalucia.py`), se define el contenido de la página principal de nuestra aplicación bajo el framework Streamlit (`page()`), allí se inicializa el estado de la sesión, configura la interfaz llevando una cierta lógica a partir del diseño del sistema (desde `_page_config()`, `_page_sidebar()`, `_page_footer()`) y establece los elementos de la aplicación de la página central (`_page_central_layout()`).

Algoritmo 10: Carga de la Página Principal

Data: Estado de la sesión, configuración de interfaz, elementos de la aplicación
Result: Inicialización de la aplicación y carga de la interfaz
begin
  if *El estado de la sesión está vacío* **then**
    Inicializar lista de mensajes;
    Agregar mensaje de bienvenida;
  Llamar a `_page_config()` para establecer configuración de la página;
  Llamar a `_page_sidebar()` para configurar la barra lateral;
  Llamar a `_page_central_layout()` para definir el diseño central;
  Llamar a `_page_footer()` para establecer el pie de página;

En la carga de la configuración, destacar que se ejecuta de forma **dinámica** la carga de estilos a partir de lo configurado desde el fichero de configuración (`config.yaml`) en el apartado `frontend` desde un fichero `.css` de estilos. Por ejemplo:

```
frontend:
  css_path: frontend/assets/css/style.css
  images_path: frontend/assets/
  themes:
    Claro: theme-light
    Oscuro: theme-dark
    Frío Minimalista: theme-cold
    Verde Profesional: theme-green
    Cálido Acogedor: theme-warm
  path_graphics: graphics
```

Algoritmo 11: Configuración de la Página _page_config()

Data: Configuración del sistema y estilos de la aplicación
Result: Página inicial con interfaz estructurada
begin
    Establecer título y diseño de la página;
    Cargar estilos desde archivo CSS;
    Verificar existencia del archivo y aplicar estilos;

Para la carga de la barra izquierda, cabe destaca que la carga de los modelos se produce de forma **dinámica** a partir de la configuración del fichero (`config.yaml`). En concreto a partir de la clave `models` siendo esta un ejemplo de estructura:

```
models:
  Salamandra 2b instruct:
    model: BSC-LT/salamandra-2b-instruct
    tokenizer: BSC-LT/salamandra-2b-instruct
    model_message: Salamandra es un modelo diseñado .....
```

Indicar que esta clave (`Salamandra 2b instruct`) se usará para mostrar el valor del modelo en la interfaz. Este sería la forma completa de cómo se implementaría:

Algoritmo 12: Configuración de la Barra Lateral _page_sidebar()

Data: Opciones del usuario: modelo de lenguaje y temas visuales
Result: Configuración de parámetros del usuario
begin
    Inicializar opciones del usuario;
    Mostrar modelos disponibles y capturar selección;
    Aplicar configuración del modelo seleccionado por defecto;
    Mostrar inicialización Historial de consultas;

El siguiente paso es mostrar la paǵina central donde cabe destacar la presencia del control `st.tabs` gestionado por streamlit, en el que aparecen las cuatro opciones de nuestro sistema: `_display_messages()` para mostrar el chatbot, `_evaluation_in_bulk()` para ejecutar la evaluación de modelos, `_display_evaluation_files()` para el renderizado de la evaluación y `_export_comparation_evaluation_metrics()` para la exportación de la información de evaluación.

Algoritmo 13: Configuración de la interfaz principal _page_central_layout()

Data: Ninguna entrada específica
Result: Interfaz principal con secciones de chat, evaluación y exportación de datos
Definir la imagen del logo y **Mostrar encabezado y subtítulo: Definir Secciones Principales: Tabs**
Tab 1: Chat
    Ejecutar función `_display_messages()` para mostrar el historial de mensajes
    Mostrar campo de entrada de texto con clave `'user_input'`
    Configurar evento de disparo al cambiar el input: ejecutar `process_input()`
Tab 2: Evaluación masiva de modelos
    Ejecutar función `_evaluation_in_bulk()` para gestionar la evaluación
Tab 3: Construcción de evaluación
    Ejecutar función `_display_evaluation_files()` para mostrar archivos de evaluación
Tab 4: Exportación de datos
    if *'metrics' en estado de sesión* **then**
        Ejecutar función `_export_comparation_evaluation_metrics(st.session_state["metrics"])`
    else
        No realizar ninguna acción

6.2.2. Generación de la pregunta

Una vez renderizado el sistema, y dentro de la opción del tab 1 `Chat`, el usuario accede a introducir la pregunta acerca de las leyes del presupuesto andaluz. Allí aparece un `st.text_input` manejado por streamlit para introducir la pregunta del usuario. Una vez que se pulsa Intro, se ejecuta la llamada a la función `process_input` en la que muestra un 'spinner' (`thinking_spinner`) mientras el proceso de búsqueda de la respuesta está realizándose.

El siguiente paso, es propiamente la llamada a la API rest de Flask por medio de la invocación al **método POST /query**:

```
1 requests.post(f'{config.frontend.base_api_url}:config.frontend.
    base_api_port}/query', json={'query': user_query, 'model_name':
    model_name}).json()
```

donde le pasamos el modelo que hemos seleccionado en el selector de la sección del sidebar ('model_name') y la consulta que el usuario ha introducido. Además captura del fichero de configuración .yaml cual es el endpoint de la API y su puerto. Así básicamente sería de esta manera:

Algoritmo 14: Procesamiento de Entrada del Usuario - process_input

Data: Consulta del usuario en el chat
Result: Respuesta generada por el modelo de lenguaje
begin
  Capturar la consulta ingresada;
  Mostrar spinner;
  Llamada API para enviar la consulta al modelo seleccionado;
  Recibir respuesta y calcular tiempo de procesamiento;
  Almacena la consulta y respuesta para mostrar en pantalla;
  Almacenar la consulta y respuesta en el historial;

6.2.3. Obtención de la respuesta del modelo

Siguiendo con el flujo de la aplicación, se produce la llamada anteriormente comentada a la API de Flask. Esta llamada, y todas las demás, están registradas en el fichero `rag_api.py`. En él, está registrado la asociación de la ruta `/query` con la implementación en el método `process_query`.

```
@app.route("/query", methods=["POST"])
def process_query():
```

6.2.3.1. Obtención del contexto

Esta implementación consta del acceso a la base de datos vectorial extrayendo todos los fragmentos relacionados semánticamente con la consulta del cliente `query`. Para ello se utiliza la clase manejadora del contexto `RetrieverHandler` en su llamada `get_retrieved_context` pasando como parámetro la pregunta del usuario (`query`).

Veamos en más detalle todo. `RetrieverHandler` actúa como el manejador principal del retriever dentro del sistema RAG (Retrieval-Augmented Generation) y tiene una dependencia directa con la clase `VectorStoreManager`, encargada de gestionar el acceso a la base de datos vectorial y proporcionar los métodos necesarios para la recuperación de información. La obtención de estos documentos se realiza mediante la invocación del método `self.vector_store_manager.get_retriever()`, el cual, a su vez, se apoya en la clase `Embedding` para generar la representación vectorial (*embedding*) de los datos.

El proceso continúa con la ejecución del método `get_retriever()` en la clase `VectorStore`, la cual está diseñada bajo el patrón de diseño **Factory** mediante la clase `VectorFactory`. Finalmente, este flujo conduce a `ChromaVectorStore`, donde el método `get_retriever()` implementa la lógica específica para obtener el *retriever* dentro del sistema Chroma, asegurando una recuperación eficiente y optimizada de la información relevante. Además está bajo una interface llamada `VectorStoreBase`. La parametrización del retriever tiene determinada como

```
vector_store = Chroma(
            persist_directory=self.persist_directory,
            embedding_function=embedding_function,
            collection_name=self.collection_name,
            collection_metadata={"hnsw:space": "cosine"}
)
.....
search_type="similarity_score_threshold",
search_kwargs={"k": 3, "score_threshold": 0.4}
```

El parámetro `search_type="similarity_score_threshold"` permite filtrar documentos basándose en un umbral de similitud, asegurando que solo aquellos con una pun-

tuación mínima sean considerados. Adicionalmente, `search_kwargs` establece que se recuperen como máximo tres documentos (`k=3`), siempre que su similitud supere el umbral de `0.4`, equilibrando precisión y relevancia en la búsqueda. Además se fuerza para **buscar por la distancia del coseno** en vez de la distancia euclídea.

A continuación se detalla con el código fuente:

Algoritmo 15: Algoritmo de recuperación de contexto en RetrieverHandler

Data: Consulta del usuario $query$
Result: Lista de documentos relevantes $retrieved\_context$
Inicio de la clase RetrieverHandler
Inicializar instancia de `VectorStoreManager`

Función `get_retrieved_context(query)`:
Obtener el **retriever** desde `VectorStoreManager`
Ejecutar la consulta con `retriever.invoke(query)`
Almacenar los contextos recuperados en `retrieved_context`
Imprimir retrieved_context"para depuración
Retornar `retrieved_context`

Algoritmo 16: Obtención de un recuperador de información basado en embeddings en VectorStoreManager

Input: Ninguna
Output: Instancia del recuperador de información basado en embeddings
Obtener `embedding_function` ← `self.embedding.get_embedding_base();`
Retornar `self.vector_store.get_retriever(embedding_function);`

Algoritmo 17: Obtención de un recuperador de información basado en embeddings VectorStore

Input: Función de embeddings `embedding_function`
Output: Instancia del recuperador de información basado en embeddings
Retornar `self.vector_store.get_retriever(embedding_function);`

6.2.3.2. Obtención respuesta

El proceso de generación de respuestas en el sistema RAG inicia con la llamada a la API, donde se verifica si el modelo seleccionado ya está cargado en memoria. Esto permite **evitar cargas repetitivas innecesarias**, actuando como un mecanismo de caché en código. Si el modelo no está en memoria, se inicializa mediante la clase manejadora de consultas `AskHandler`, pasando como parámetro el nombre del modelo.

Algoritmo 18: Obtener un recuperador de información en ChromaDB ChromaVectorStore

Input: Función de embeddings `embedding_function`
Output: Instancia del recuperador de información basado en ChromaDB
Try Inicializando ChromaDB para recuperación de información...;
`vector_store ← Chroma(self.persist_directory, embedding_function, self.collection_name);`
return
`vector_store.as_retriever(search_type="similarity_score_threshold", search_kwargs={"k": 3, "score_threshold": 0.4})`
Catch Exception `error`

A continuación, `AskHandler` invoca el método `get_response(query, retriever_contexts)`, utilizando el contexto proporcionado por `RetrieverHandler` y la consulta del usuario. Dado que el sistema emplea modelos de Hugging Face configurados desde un fichero `.yaml`, la gestión del modelo se delega a la clase `HuggingFaceProvider`, la cual opera bajo la interfaz `IaProviderBase`. Su función principal es la construcción dinámica de prompts, la configuración de hiperparámetros y la generación de respuestas optimizadas.

El flujo de procesamiento sigue los siguientes pasos:

1. **Normalización del modelo:** Se convierte el nombre del modelo a minúsculas para evitar inconsistencias.
2. **Generación del prompt:** Se construye dinámicamente en función del contexto recuperado (`self.hugginface_provider.build_prompt(retrieved_contexts, query)`).
3. **Configuración de parámetros del modelo:**
   - Temperatura, calculada mediante el método privado `_get_optimal_temperature()`.
   - Número máximo de tokens permitidos en la respuesta.
   - Método de muestreo aplicado a la generación de texto.
   - Inclusión de un mensaje de sistema, determinado por `_requires_system_prompt()`.
4. **Ejecución del modelo:** Se genera la respuesta utilizando el modelo seleccionado.
5. **Limpieza de la salida:** Se eliminan etiquetas innecesarias y formatos no deseados.
6. **Retorno de la respuesta:** Se estructura la salida en un formato optimizado.

En nuestro caso, para los modelos "salamandra", "mistral.[el] mensaje al modelo hay que pasarle el role `system` con un texto específico (`Eres un asistente útil y experto en legislación.`). Para el resto, simplemente se le pasa el role `user` siendo la base del prompt:

```
prompt = f"""
Eres un asistente experto en legislación y de leyes de presupuestos públicos
de la Junta de Andalucía.
Tu tarea es responder preguntas de manera precisa y fundamentada utilizando el contexto
proporcionado.

Contexto:
{context_text}

Instrucciones para Responder:
- Usa únicamente el contexto para responder. No inventes datos ni la respuesta.
- Si el contexto no contiene la respuesta, di: "No se encontró información suficiente en
el contexto proporcionado para responder a esta pregunta con certeza."
- Responde de manera concisa y objetiva.
- Si la pregunta requiere una cifra presupuestaria, proporciona el número exacto con
su unidad monetaria en euros.
- Si se menciona una ley o decreto o un artículo, especifica su nombre completo, fecha de
publicación y el artículo.
- Estructura la respuesta en un lenguaje formal y claro.
- Evite usar comillas dobles en cualquier parte de la respuesta.
- Si la pregunta quiere comparar cifras presupuestarias por años, muestra la respuesta
que contenga cada año y su valor correspondiente de forma clara.
- Siempre diga "¡Gracias por preguntar!" al final de la respuesta.

"""
query =  f"""
### Pregunta:
{user_query}

Genera una respuesta fundamentada basada en el contexto proporcionado y las instrucciones
anteriores para responder.
"""
```

En definitiva, esta arquitectura modular y configurable permite una integración flexible y eficiente de distintos modelos de lenguaje (carga dinámica de modelos) dentro del sistema RAG, asegurando un rendimiento óptimo en la recuperación y generación de

información. Veamos un resumen de todo ello:

Algoritmo 19: Procesar Consulta al Modelo RAG - POST: /query

Data: Consulta del usuario (`query`) y modelo seleccionado (`model_name`)
Result: Respuesta del modelo con contexto y gráficos
begin
  Leer `query` y `model_name` desde la solicitud JSON;
  if *query o model_name es vacío* **then**
    Retornar error "Query y model_name son requeridos";
  else
    Crear instancia de `RetrieverHandler` para recuperar contexto;
    Obtener contextos recuperados mediante `retriever.get_retrieved_context(query);`
    if *El modelo cargado es diferente al solicitado* **then**
      Cargar nuevo modelo con `AskHandler(model_name);`
      Actualizar `current_model_name;`
    else
      Usar modelo en memoria;
    Obtener respuesta del modelo con contexto mediante `get_response(query, retriever_contexts);`
    Crear instancia de `GraphicsAnalyzer` y analizar gráficos;
    if *Gráficos generados* **then**
      Preparar datos con `DataPrepare.data_graphics(response_query);`
    else
    Retornar respuesta en formato JSON;

Algoritmo 20: Generación de Respuestas en `AskHandler`

Input: Modelo de lenguaje `model_name`, Consulta del usuario `query`, Contextos recuperados `retrieved_contexts`

Output: Diccionario con la respuesta generada bajo la clave `response_query` ;

Normalizar `model_name` (convertir a minúsculas) ;

Instanciar `HuggingFaceProvider` con el modelo ;

Obtener instancia del modelo `llm` desde el proveedor `HuggingFaceProvider` ;

Generar `prompt` y desde `build_prompt()` ;

Determinar si se requiere mensajes del sistema para **sistema de prompts** con `_requires_system_prompt()` ;

Obtener temperatura óptima del modelo con `_get_optimal_temperature()` ;

Configurar `top_p` y `max_tokens` según el modelo ;

Inicializar lista `messages` ;

if *Se requiere* ***sistema de prompts*** **then**

  Agregar mensaje del sistema ;

end

Agregar mensaje del usuario con `prompt` ;

  Definir parámetros de generación: `do_sample = True`, `temperature`, `top_p`, `max_new_tokens` ;

Invocar modelo con `self.llm(messages, **generate_kwargs)` ;

Extraer texto limpio con `_extract_and_clean_text(output)` ;

Excepción `e` Imprimir mensaje de error y lanzar excepción ;

return `response_query` con la respuesta generada ;

A continuación, se detalla la parametrización del código para definir los valores clave en la ejecución del modelo:

Definición de los valores de los parámetros `top_p` y `max_tokens` en función del nombre del modelo:

```
top_p = 0.85 if "deepseek" in self.model_name else 0.9
max_tokens = 512 if "2b" in self.model_name else 1024
```

Indicación indica a qué modelos es necesario pasarles el parámetro `system` en el mensaje:

```
use_system_prompt = self._requires_system_prompt()
```

```
def _requires_system_prompt(self):
        """
        Determina si el modelo necesita un mensaje de sistema.

        Returns:
            bool: True si el modelo requiere un mensaje de sistema, False en caso cont
        """
        mistral_models = ["salamandra", "mistral", "ministral"]
        return any(name in self.model_name for name in mistral_models)
```

Valor de la Temperatura por Modelo

El siguiente algoritmo define los valores de temperatura para diferentes modelos de lenguaje:

```
if "deepseek" in self.model_name:
    return 0.1
elif "llama" in self.model_name:
    return 0.2
elif "salamandra" in self.model_name:
    return 0.1
elif "mistral" in self.model_name or "ministral" in self.model_name:
    return 0.2
else:
    return 0.2  # Valor por defecto
```

Estos valores permiten ajustar la variabilidad de las respuestas generadas por cada modelo, asegurando un comportamiento adecuado en función de su arquitectura.

Indicar también que la clase `HuggingFaceProvider` actúa como un gestor de modelos de lenguaje dentro del sistema RAG, integrando modelos de **Hugging Face** y optimizando su ejecución mediante técnicas de cuantización.

- **Inicialización del modelo:** Se obtiene el nombre del modelo desde la configuración y se autentica mediante un token de acceso a Hugging Face.
- **Optimización del rendimiento:** Antes de cargar el modelo, se libera la memoria de la GPU (`torch.cuda.empty_cache()`) y se ejecuta una recolección de basura (`gc.collect()`).

- **Cuantización en 4 bits:** Se configura el modelo para utilizar `BitsAndBytesConfig`, permitiendo una ejecución eficiente con menor consumo de memoria.

- **Carga del modelo:** Se instancia un modelo preentrenado de `AutoModelForCausalLM` con los parámetros configurados, garantizando compatibilidad con el pipeline de generación de texto.

6.2.3.3. Análisis y renderizado final de la respuesta

Una vez conseguida la respuesta, se analiza dicha respuesta *buscando pares de años y valores* para construir una gráfica comparativa de barras que aparecerá justamente después de la respuesta si cumple esas condiciones, utilizando la clase `GraphicsAnalyzer` pasando por parámetro al constructor la pregunta y la respuesta del modelo y llamando a su método analyze (`analyzer.analyze()`). Si posee esa estructura, devolverá los datos en `response['graphics_data']` para su posterior renderizado junto con el resto de la respuesta del modelo (`return jsonify(response)`).

Una vez que tenemos la respuesta gracias a la devolución de la información desde el endpoint de la API, ya en el **frontend** se procede a la visualización en pantalla de los mensaje en el chatbot. Para ello desde el frontend, que ya posee la respuesta, itera todos los mensajes que se han guardado en la variable de sesión de streamlit `st.session_state['messages']` para ir uno a uno redendizándolos desde la función (`_display_messages()`) y mostrando la información ya bién sea texto o si tras el analizador de la respuesta considera que hay que mostrar la información del gráfico de barras de los pares de valores (año-valor).

Si es así, se renderiza justo despues de la respuesta en el mismo chat gracias a la funcion `_draw_graphics_chat` pasandole como parámetro los datos de valores. Dentro de esta función utiliza la clase del frontend `GraphicsPlotter` para poder mostrar los datos en forma de tabla, un gráfico de barras y de barras/lineas en el mismo chatbot.

Algoritmo 21: Visualización de Mensajes en el Chat

```
Data: Historial de mensajes del usuario y asistente
Result: Mensajes mostrados en la interfaz del chat
begin
    Mostrar cada mensaje en la conversación;
    Si el mensaje contiene gráficos, renderizarlos;
    Mostrar tiempo de procesamiento si está disponible;
```

6.2.4. Adicción al historial de consultas y renderizado

Dentro del frontend, aparece la funcion `_page_sidebar()` en la que parte de ella se produce el renderizado del historial de conversaciones del chat que está guardado en `st.session_state["history"]` tras cada respuesta final que devuelve la API. Básicamente existe un bloque dentro de esta función, que recorre con un bucle esta variable de streamlit pintando el texto de la consulta y de la respuesta que aparecerá en la parte inferior izquierda todo el historial pintado.

6.3. Ejecución de la evaluación de Modelos

El proceso de evaluación masiva de modelos comienza cuando el usuario selecciona esta opción en la interfaz del **frontend** a través del control `tab`. Esto desencadena la ejecución de la función `_evaluation_in_bulk()`, la cual orquesta la evaluación de los modelos en varios pasos, detallados a continuación.

6.3.1. Paso 1 - Importación fichero 'Source of Truth'

El primer paso consiste en la importación de un archivo `.csv`, denominado **source of truth**, que contiene dos columnas: `'pregunta'` y `'respuesta'`. Este archivo almacena las consultas y respuestas de referencia que serán utilizadas en la evaluación de los modelos.

La carga del archivo se gestiona a través del control `st.file_uploader()` de `Streamlit`, permitiendo la subida de archivos a un almacenamiento temporal. Una vez subido, la API es invocada mediante una solicitud `POST` al endpoint `/upload`:

```
1 requests.post(f'{config.frontend.base_api_url}:{config.frontend.
    base_api_port}/upload', files=files)
```

En la API, el método `upload_files()` obtiene el servicio `upload_file_importer` desde el contenedor y lo ejecuta sobre los archivos cargados. Internamente, este servicio utiliza la clase `FileLoader` para procesar archivos en formato `.csv` Figura 5, extraer su contenido y almacenarlo en memoria para su posterior evaluación. Finalmente, los archivos temporales son eliminados.

Algoritmo 22: Carga y procesamiento de archivos

Input: Lista de archivos `files` enviados

Output: JSON con estado de la carga o mensaje de error

```
// Obtener archivos del request
files ← request.files.getlist("files") ;
if files está vacío then
    return jsonify({'error': "No se encontraron archivos"}), 400 ;
else
    // Obtener servicio de importación de archivos
    file_importer ←
     container.upload_file_importer(app.config['UPLOAD_FOLDER']) ;
    // Procesar archivos con el importador
    return jsonify(file_importer.upload_files(files)), 200 ;
end
```

6.3.2. Paso 2 - Selección de Modelos para Evaluación

Una vez importadas las preguntas y respuestas, el usuario selecciona los modelos a evaluar mediante un control `multiselect` en `Streamlit`:

```
selected_options = st.multiselect(
    "Selecciona los modelos para evaluar:",
    options=options,
    default=st.session_state["selected_models"],
    key="model_selection"
)
```

6.3.3. Paso 3 - Ejecución de la Evaluación

Tras la selección de los modelos, el usuario inicia la evaluación mediante el botón de la interfaz `st.button('Evaluar')`, lo que genera una solicitud a la API REST de Flask a través del endpoint `POST /bulk_evaluate`:

```
requests.post(f'{config.frontend.base_api_url}:{config.frontend.
    base_api_port}/bulk_evaluate',
```

```
2                 json={'models': selected_options, 'querys_references'
                      : combined_documents})
```

Esta solicitud envía los modelos seleccionados junto con la base de datos de preguntas y respuestas obtenida en el Paso 1. En la API, los datos son procesados y la evaluación se ejecuta en segundo plano mediante la biblioteca `threading`, evitando el bloqueo de la interfaz gráfica y permitiendo al usuario continuar con otras tareas mientras se lleva a cabo la evaluación.

Cada evaluación se identifica con un **ID único**, generado dinámicamente a partir de la fecha y hora de inicio del proceso. Además, tanto los resultados como posibles errores se almacenan en archivos para su posterior análisis.

Algoritmo 23: Ejecución de Evaluación Masiva - bulk_evaluate()

Input: Lista de consultas de referencia `querys_references`, Lista de modelos `models`

Output: JSON con mensaje de inicio y ID de evaluación

```
// Obtener datos de la solicitud JSON
if querys_references o models están vacíos then
    return jsonify({'error': 'Datos insuficientes'}), 400 ;
else
    // Generar un identificador único de evaluación
    evaluation_id ← datetime.now().strftime("%Y-%m-%d_%H-%M-%S") ;
    // Ejecutar evaluación en segundo plano
    thread ← Thread(target=run_bulk_evaluation,
     args=(querys_references, models, evaluation_id)) ;
    thread.start() ;
    return jsonify({'message': 'Evaluación en proceso',
     'evaluation_id': evaluation_id, 'status': 'processing'}) ;
end
```

Internamente, dentro del método `run_bulk_evaluation()`, se inicializan las clases `RetrieverHandler()` y `AskHandler()`. Para cada pregunta, el sistema invoca el método `query_handler.get_response(question, retrieved_contexts)`, obteniendo la respuesta del modelo. Posteriormente, esta respuesta es evaluada a través de la clase `RagasEvaluator`, la cual implementa la interfaz `EvaluatorBase` diseñada para encapsular el cálculo de métricas utilizando la librería `Ragas`.

Para cada consulta contenida en el archivo *source of truth*, se instancia un objeto

`RagasEvaluator` y se llama a su método `evaluate()`, pasando como parámetros la pregunta, los contextos recuperados de la base de datos vectorial, la respuesta generada por el modelo y la respuesta de referencia contenida en el archivo CSV:

```
ragas_evaluator.evaluate(
    question, retrieved_contexts, response_model["response_query"],
        reference
)
```

Algoritmo 24: Evaluación en Segundo Plano - run_bulk_evaluation()

Input: Lista de referencias `querys_references`, Modelos a evaluar `models`, ID de evaluación `evaluation_id`

Output: Resultados guardados en archivos JSON y CSV

```
// Inicialización
foreach modelo en models do
    query_handler ← AskHandler(modelo) ;
    foreach pregunta, referencia en querys_references do
        contexto ← retriever.get_retrieved_context(pregunta) ;
        respuesta ← query_handler.get_response(pregunta, contexto) ;
        // Evaluación con Ragas
        evaluator ← RagasEvaluator() ;
        evaluacion ← evaluator.evaluate(pregunta, contexto, respuesta,
         referencia) ;
        results[modelo][pregunta] ← evaluacion ;
        error results[modelo][pregunta] ← 'error': str(error) ;
    end
end
// Guardar resultados
save_evaluation_to_file('evaluacion_' + evaluation_id + '.json',
 results) ;
csv_path ← save_evaluation_to_csv('evaluacion_' + evaluation_id +
 '.csv', results) ;
if csv_path es None then
    Lanzar excepción: 'Error al guardar CSV' ;
end
```

Algoritmo 25: Evaluación de Respuesta con Ragas - evaluate()

Input: Consulta `query`, Contextos `retrieved_contexts`, Respuesta generada `response`, Respuesta de referencia `reference`

Output: Diccionario con métricas de evaluación

```
// Liberar memoria
// Preparar datos para evaluación
processed_context ← concatenar contenidos de retrieved_contexts ;
dataset ← Dataset.from_dict({'query', 'response', 'context',
 'reference'}) ;
Try // Ejecutar evaluación con Ragas
evaluation_result ← evaluate(dataset, métricas) ;
df ← evaluation_result.to_pandas() ;
// Extraer métricas relevantes
return df.to_dict() ;
Catcherror ;
Lanzar excepción con mensaje de error ;
```

Este flujo permite una evaluación eficiente y escalable, asegurando que los modelos sean analizados de manera flexible y sin afectar la experiencia del usuario en la interfaz.

6.4. Construcción de la evaluación de Modelos

El proceso comienza cuando el usuario, a través de la **interfaz gráfica**, inicia la ejecución de la función `_display_evaluation_files()`, la cual constituye el punto de entrada del procedimiento en el **frontend**. Esta función se encarga de gestionar y presentar los archivos de evaluación generados previamente. Veámos los pasos en detalle:

1. **Obtención de Archivos Disponibles:** Se ejecuta la función `_get_available_files()`, que recupera tanto los archivos de evaluación (`.csv`) como los registros de errores generados en evaluaciones previas.

2. **Gestión de Errores:** En caso de que existan errores, se invoca la función `_show_error_logs(e`, la cual muestra los detalles de los registros de errores almacenados en archivos JSON.

3. **Visualización de Evaluaciones:** Si existen archivos de evaluación, se genera

un resumen consolidado mediante la función:

```
df_summary = _extract_evaluation_summary(csv_files)
```

Este resumen es posteriormente mostrado en un **DataFrame interactivo** en Streamlit mediante:

```
st.dataframe(df_summary)
```

Dicho resumen contiene:

- **Listado de archivos disponibles**.
- **Número total de preguntas evaluadas**.
- **Modelos utilizados en cada evaluación**.

4. **Selección del Archivo de Evaluación:** Se despliega un **selector interactivo** que permite elegir el archivo `.csv` deseado mediante la instrucción:

```
selected_csv = st.selectbox("Selecciona un archivo CSV:",
    csv_files)
```

5. **Vista Previa del Archivo Seleccionado:** Una vez seleccionado el archivo, se muestra un extracto de su contenido (*head*) para confirmar su validez antes de proceder con el procesamiento:

```
st.dataframe(pd.read_csv(selected_csv_path).head())
```

6. **Inicio del Proceso de Construcción de Evaluación:** Finalmente, se habilita un botón de acción que permite iniciar el procesamiento de la evaluación sobre el archivo seleccionado:

```
st.button("Construcción Evaluación")
```

Una vez que el usuario pulsa el botón de ejecución, el sistema invoca la función `_process_selected_csv(selected_csv_path)`, a la cual se le pasa la ruta del archivo `.csv` que contiene los datos de evaluación. Este paso es fundamental, ya que permite la carga y estructuración de la información necesaria para proceder con el análisis de modelos de lenguaje.

A continuación, los datos se procesan y se realiza la representación gráfica de los resultados. Para ello, se llama a la función:

```
1 _draw_graphics_evaluation(df, models)
```

, donde:

- `df`: Contiene los datos extraídos y organizados desde el fichero `.csv`.
- `models`: Representa la lista de modelos de lenguaje seleccionados para la evaluación.

Esta función genera las visualizaciones necesarias para la interpretación de los resultados, facilitando la comparación entre diferentes modelos y su rendimiento en función de las métricas establecidas. Una vez completado el proceso de renderización, el sistema prepara las métricas obtenidas para su posterior exportación. Esto se realiza almacenando las métricas en la variable de estado de Streamlit mediante la siguiente asignación:

```
1 st.session_state["metrics"] = metrics
```

Este almacenamiento en `st.session_state` permite que las métricas queden disponibles globalmente dentro de la sesión de ejecución, asegurando su accesibilidad en la etapa de exportación de resultados.

Este flujo garantiza una interacción fluida con el usuario, permitiéndole seleccionar y procesar evaluaciones de manera eficiente y sin ambigüedades.

A continuación, se presenta la implementación del proceso descrito:

Algoritmo 26: Visualización y Selección de Evaluaciones - _display_evaluation_files()

Input: Directorio de evaluaciones `EVALUATION_DIR`
Output: Resumen de archivos disponibles y selección del usuario

```
// Obtener archivos de evaluación
(csv_files, error_files) ← _get_available_files() ;
// Mostrar errores si existen
if error_files ≠ ∅ then
    _show_error_logs(error_files)
end
if csv_files está vacío then
    return Mostrar advertencia "No se encontraron archivos CSV"
else
    // Mostrar resumen de evaluaciones
    df_summary ← _extract_evaluation_summary(csv_files) ;
    st.dataframe(df_summary) ;
    // Selector de archivos
    selected_csv ← st.selectbox("Selecciona un archivo", csv_files) ;
    selected_csv_path ← EVALUATION_DIR + selected_csv ;
    // Mostrar contenido del archivo
    st.dataframe(pd.read_csv(selected_csv_path).head()) ;
    // Botón para iniciar evaluación
    if st.button(Çonstrucción Evaluación") then
        _process_selected_csv(selected_csv_path)
    end
end
```

Algoritmo 27: Obtención de Archivos de Evaluación - _get_available_files()

Input: Directorio de evaluación `EVALUATION_DIR`
Output: Lista de archivos CSV y lista de errores

```
if EVALUATION_DIR no existe then
    return (∅, ∅)
end
// Listar archivos ordenados por fecha
files ← sorted(os.listdir(EVALUATION_DIR), reverse=True) ;
// Filtrar archivos CSV y errores
csv_files ← [f para f en files si f termina en ".csv"] ;
error_files ← [f para f en files si f inicia con .erroror_"] ;
return (csv_files, error_files)
```

Algoritmo 28: Extracción de Resumen de Evaluaciones - _extract_evaluation_summary(csv_files)

Input: Lista de archivos CSV `csv_files`
Output: Resumen de evaluaciones en DataFrame
`summary` ← lista vacía ;
foreach *file en csv_files* **do**
    `file_path` ← `EVALUATION_DIR + file` ;
    `df` ← `pd.read_csv(file_path)` ;
    `num_questions` ← tamaño de `df` ;
    `models` ← `df["Model"].unique()` ;
    `summary.append({."Archivo": file, "Preguntas Evaluadas": num_questions, "Modelos": models})` ;
    Error `summary.append({."Archivo": file, "Preguntas Evaluadas": ."Error", "Modelos": ."Error"})` ;
end
return `pd.DataFrame(summary)`

Algoritmo 29: Gestión y Visualización de Errores de Evaluación - _show_error_logs(error_files)

Input: Lista de archivos de error `error_files`
Output: Mensajes de error mostrados en la interfaz
if *error_files no está vacío* **then**
    `// Mostrar advertencia en la interfaz`
    `st.warning(' Se han detectado errores en evaluaciones recientes.');`
    foreach *error_file en error_files* **do**
        `// Construir la ruta completa del archivo`
        `error_path` ← `os.path.join(EVALUATION_DIR, error_file);`
        `// Abrir y leer el archivo JSON`
        `f` ← `open(error_path, 'r', encoding=ütf-8");`
        `error_data` ← `json.load(f);`
        `f.close();`
        `// Mostrar el error en la interfaz`
        `st.error(f' {error_file}: {error_data['error']}');`
    end
end

Algoritmo 30: Procesamiento de CSV y Generación de Gráficos - _process_selected_csv

Input: `selected_csv_path`: Ruta del archivo CSV
Output: `metrics`: Métricas de evaluación procesadas
`df` ← `pd.read_csv(selected_csv_path)` ;
`metrics` ← columnas en `df` excepto ["Model", "Question", ...] ;
`models` ← valores únicos en `df["Model"]` ;
_draw_graphics_evaluation(`df`, `models`) ;
`st.session_state["metrics"]` ← `df.fillna("null")` ;
return `metrics` ;

Siguiendo por último con la renderización propia de la evaluación dentro de la función `_draw_graphics_evaluation(df, models)`, dentro de ella se instancia la clase `GraphicsPlotter` que alberga todas los métodos necesarios para mostrar los **Resultados por métrica y modelo**,**Análisis de medias por métrica y modelo** y **Valoración final de modelos**.

En caso de fallos en la renderización, la función captura la excepción y muestra un mensaje en la interfaz, asegurando que la aplicación continúe funcionando sin interrupciones.

Con esta implementación, los usuarios pueden analizar de manera visual y efectiva los resultados obtenidos en la evaluación de modelos RAG, facilitando la toma de decisiones basada en datos.

Algoritmo 31: Renderización de Evaluación de Modelos - _draw_graphics_evaluation(...)

Input: `df`: Datos de evaluación, `models`: Modelos evaluados
Output: Visualización de métricas y valoración de modelos

```
plotter ← GraphicsPlotter(df, models) ;
```
Mostrar Resultados de Evaluación ;
```
plotter.render_table_by_metrics() ;
plotter.render_distributions() ;
plotter.render_scatter_plots() ;
```
Mostrar Medias por Métrica/Modelo ;
```
table_data ← plotter.render_table_summary() ;
plotter.render_bar_chart_summary(table_data) ;
plotter.render_combined_bar_line_chart() ;
plotter.render_radar_chart() ;
plotter.render_correlation_heatmap() ;
plotter.render_scatter_plots_summary() ;
plotter.render_regression_plots() ;
```
Mostrar Valoración Final ;
```
plotter.render_weights() ;
plotter.render_radar_chart_overall(include_overall_score=True) ;
plotter.render_distribution_overall() ;
plotter.render_correlation_heatmap_overall() ;
plotter.render_overall_score_correlation_heatmap() ;
plotter.render_overall_score_table_with_winner() ;
```
Manejo de Errores ;
if *Error en generación de gráficos* **then**
```
    st.write("No se pudo dibujar el gráfico") ;
```
end

6.4.1. Metodología utilizada

Evaluar la calidad de modelos Retrieval-Augmented Generation (RAG) es un desafío debido a la necesidad de garantizar la precisión, medir qué tan bien la respuesta generada por el modelo está respaldada por la información recuperada y la relevancia semántica de las respuestas generadas. Para abordar este problema, se ha implementado **RAGAS (Retrieval-Augmented Generation Assessment Suite)** [56], una suite de métricas diseñadas específicamente para evaluar estos modelos de manera efectiva.

6.5. Exportación de las métricas de la evaluación de Modelos

El proceso comienza cuando el usuario, a través de la **interfaz gráfica**, inicia la ejecución de la función `_export_comparation_evaluation_metrics(st.session_state['metrics'])`, la cual constituye el punto de entrada del procedimiento en el **frontend**. A esta función, se le pasa como parámetro `st.session_state['metrics']`, que es la variable de sesión de streamlit encargada de contener la última métrica evaluada. Esta función se encarga de realizar la exportación de dicha información en formato `.csv` o `.JSON`. Veámos los pasos en detalle:

Algoritmo 32: Exportación de Métricas en CSV o JSON - _export_comparation_evaluation_metrics

Input: Métricas `metrics`, Formato de exportación `export_format`
Output: Archivo descargable en formato CSV o JSON

```
// Mostrar opciones de exportación
export_format ← st.selectbox(['CSV', 'JSON']).lower() ;
if st.button('Exportar') then
    // Convertir métricas en DataFrame si es necesario
    if metrics es una lista then
        metrics ← pd.DataFrame(metrics)
    else
        metrics ← metrics.fillna('null')
    end
    // Enviar datos a la API para la exportación
    response ← requests.post(API_EXPORT, json={'metrics': metrics,
     'format': export_format}) ;
    if response.status_code == 200 then
        st.download_button('Descargar métricas', response.content)
    else
        st.error('Error en exportación')
    end
end
```

Siguiendo con el flujo, a continuación se describe el flujo en detalle dentro de la llamada a la API:

1. **Recepción de datos**: La API recibe las métricas generadas en el proceso de evaluación.

2. **Selección del formato**: Se determina si la exportación se realizará en formato **CSV** o **JSON**.

3. **Conversión de datos**: Si los datos no están en un formato adecuado, se convierten a un `DataFrame` de `pandas`.

4. **Normalización**: Se reemplazan valores nulos por la cadena 'null' para evitar inconsistencias.

5. **Aplicación de patrón diseño**: Se selecciona el **patrón estrategia (Strategy Pattern)** en la exportación a través del servicio `FormatsExporter` que es obtenido a través del contenedor de dependencias (`container`).

6. **Generación del archivo**: Se convierte el `DataFrame` al formato seleccionado y se envía como archivo descargable.

Algoritmo 33: Exportación de métricas en formato CSV o JSON - export_metrics()

Input: Lista de métricas `metrics`, formato de exportación `export_format`
Output: Archivo exportado en formato solicitado o mensaje de error
if *`metrics` está vacío* **then**
  return `jsonify({'error': 'No metrics provided'})`, 400
else
  `// Convertir a DataFrame si es necesario`
  if *`metrics` es una lista* **then**
    `metrics ← pd.DataFrame(metrics)`
  end
  `// Rellenar valores NaN con 'null'`
  `metrics ← metrics.fillna('null')`
  `// Obtener servicio de exportación`
  `export_context ← container.formats_exporter(export_format)`
  `// Exportar datos con la estrategia seleccionada`
  `exported_data ← export_context.export(metrics)`
  return `exported_data`, 200, {'Content-Type': tipo MIME correspondiente}
end

La implementación de la clase `FormatsExporter` dentro del patrón Strategy, permite cambiar fácilmente entre distintos formatos sin modificar el código base.

Algoritmo 34: Selector de formato para exportación de métricas

Input: Formato de exportación `format`

Output: Objeto exportador adecuado

if *format = 'csv'* **then**
  return `CSVExporter()`
else
  if *format = 'json'* **then**
    return `JSONExporter()`
  else
    return `ValueError('Formato no soportado')`
  end
end

En concreto para nuestra implementación en el sistema, define la estrategia de la clase adecuada (`CSVExporter` o `JSONExporter`) bajo la misma interface (clase `ExporterBase`) según el formato deseado por el usuario. Dando un paso más en el detalle de estas clases, la clase `CSVExporter` convierte los datos en un archivo CSV utilizando la librería `pandas` y la clase `JSONExporter` serializa los datos a formato JSON para su almacenamiento o intercambio.

Algoritmo 35: Exportación en formato CSV - CSVExporter

Input: Datos `data`

Output: Archivo CSV generado

```
// Convertir datos a DataFrame si es necesario
df ← pd.DataFrame(data)
// Exportar DataFrame a CSV
output ← StringIO() df.to_csv(output, index=False)
return output.getvalue()
```

Algoritmo 36: Exportación en formato JSON - JSONExporter

Input: Datos `data`

Output: Archivo JSON generado

if *data es un DataFrame* **then**
  `data ← data.to_dict(orient='records')`
else
end

```
// Convertir datos a JSON
return json.dumps(data, indent=2)
```

En resumen, el sistema de exportación ha sido diseñado para ofrecer una solución altamente flexible y eficiente, alineándose con los principios de **Responsabilidad Úni-**

ca (SRP) [57] y **Abierto/Cerrado (OCP)** [58]. Esto permite una fácil extensibilidad para la incorporación de nuevos formatos sin alterar la estructura base del sistema garantizando una arquitectura escalable y adaptable a futuros requerimientos. De este modo, los datos de evaluación pueden exportarse en distintos formatos según las necesidades del usuario, asegurando compatibilidad con diversos entornos y herramientas de análisis de terceros.

6.6. Resumen API Rest

La API REST desarrollada proporciona un conjunto de endpoints diseñados para el sistema RAG. Esta permite la gestión de consultas, la recuperación de contexto relevante, la ejecución de evaluaciones en segundo plano y la exportación de métricas. Aquí aparece a continuación un resumen de ella:

Algoritmo 37: Servidor Flask para Evaluación de Modelos RAG - `rag_api.py`

```
app = Flask('RAG Evaluation API')
CORS(app) // Permitir peticiones desde cualquier origen
container = Container() // Gestión de dependencias
config = Config() // Configuración del sistema
UPLOAD_FOLDER = tempfile.gettempdir()
EVALUATION_DIR = config.backend.path_evaluation
os.makedirs(EVALUATION_DIR, exist_ok=True)
current_model = None // Modelo en memoria
current_model_name = None // Nombre del modelo cargado
@app.route('/query', methods=['POST']) // Procesa consultas de usuario
@app.route('/bulk_evaluate', methods=['POST']) // Evaluación masiva
@app.route('/upload', methods=['POST']) // Subida de archivos
@app.route('/export_metrics', methods=['POST']) // Exportación de
  métricas
if script ejecutado como principal then
   app.run(host='0.0.0.0', debug=True,
 port=config.frontend.base_api_port)
end if
```

Describiendo el código anterior, comienza con la inicialización de la aplicación Flask (`app = Flask('RAG Evaluation API')`) y se habilita **CORS** para permitir el acceso a la API desde cualquier origen. A continuación, se importan las bibliotecas necesarias y se configura el entorno de ejecución. La configuración del sistema se obtiene

desde el archivo de parámetros `config.yaml`, donde se define la ruta para almacenar las evaluaciones (config.backend.path_evaluation y se asigna a `EVALUATION_DIR = config.backend.path_evaluation`). Si esta carpeta no existe, se crea automáticamente con:

```
1 os.makedirs(EVALUATION_DIR, exist_ok=True)
```

Además, se establecen variables globales para gestionar la caché del modelo utilizado en el chatbot, como `current_model`, que almacena el modelo en memoria, y `current_model_name`, que guarda el nombre del modelo cargado. Por último, se define `UPLOAD_FOLDER` como el directorio temporal donde se almacenan los archivos de referencia (*source of truth*) utilizados en la evaluación.

A continuación, se detallan las principales rutas y su funcionalidad:

- `/query` permite procesar consultas de usuario de manera eficiente. Para ello, se recupera el contexto relevante utilizando la clase `RetrieverHandler`, lo que facilita la obtención de información contextualizada. Posteriormente, se selecciona el modelo adecuado mediante `AskHandler`, el cual se encarga de gestionar la interacción con el modelo de lenguaje seleccionado ayudándose de la clase `HuggingFaceProvider` actuando como proveedor de modelos de lenguaje basados en Hugging Face. Una vez definido el modelo y el contexto, se genera una respuesta utilizando el modelo de lenguaje. Finalmente, la calidad de la respuesta es evaluada mediante `GraphicsAnalyzer`, que además genera gráficos para su posterior análisis. La respuesta, junto con los gráficos generados, es devuelta en formato JSON, proporcionando así un informe detallado de la consulta realizada.

- `/bulk_evaluate` permite ejecutar la evaluación de múltiples consultas en segundo plano, optimizando el rendimiento y evitando bloqueos en la interfaz. Para ello, recibe un conjunto de preguntas y respuestas de referencia, junto con una lista de modelos a evaluar. La evaluación se ejecuta de **manera asíncrona mediante la biblioteca** `threading`, permitiendo la ejecución de la evaluación sin afectar la interacción del usuario con la interfaz. En este proceso, `RetrieverHandler` y `AskHandler` son utilizados para recuperar el contexto adecuado y generar respuestas del modelo. Posteriormente, se emplea la clase `RagasEvaluator` para evaluar las respuestas generadas con base en diversas métricas de calidad. Finalmente, los resultados obtenidos son almacenados en archivos `JSON` y `CSV`,

permitiendo su posterior análisis y comparación siendo a cargo esta responsabilidad a cargo del frontend.

- `/upload` facilita la carga de archivos a la API, permitiendo la integración de datos adicionales en el sistema. Los archivos subidos son almacenados temporalmente en un directorio predefinido en la variable `UPLOAD_FOLDER`, asegurando una gestión eficiente del almacenamiento. Para procesar los archivos subidos, es realizada a través de un **servicio extraído del contenedor de dependencias** `Container` especializado configurado en la clase `UploadFileImporter`, que se encarga de interpretar y gestionar los archivos en función de su formato. Dentro de los archivos procesados, se incluyen bases de datos de preguntas y respuestas que serán utilizadas en las evaluaciones posteriores, asegurando la correcta incorporación de los datos en el sistema.

- `/export_metrics` permite la exportación de métricas de evaluación en diferentes formatos, tales como `CSV` y `JSON`. Para ello, los datos son convertidos en un `DataFrame` de la biblioteca `pandas`, facilitando su manipulación y análisis. Además, el sistema maneja datos nulos y valida el formato de exportación para evitar inconsistencias en los resultados generados. La exportación es realizada a través de un **servicio extraído del contenedor de dependencias** `Container` especializado configurado en la clase `container.formats_exporter()`, lo que permite una gestión flexible y adaptable a diferentes necesidades. De este modo, los usuarios pueden obtener métricas estructuradas en el formato que mejor se adapte a su análisis, asegurando la accesibilidad y reutilización de los datos.

Finalmente, si el script se ejecuta como el script principal, la aplicación Flask se inicia en el puerto configurado desde la configuración del fichero `config.yaml`. en la ruta `config.frontend.base_api_port`

6.7. Interfaz de Usuario

A continuación aparecen la implementación de la interfaz de usuario. También puede ver unos vídeos donde se explica dicha interfaz Anexo B.2.11.1:

6.7.1. Configuración Inicial

En la carga inicial del sistema, se ejecuta la función `_page()` que a su vez consta de la función `_page_config()`, la cual configura los elementos globales de la interfaz. Esto incluye la carga de estilos CSS, la definición del icono y el título de la aplicación web.

Para ello, se emplea la funcionalidad de `streamlit` con el siguiente código:

```
1 st.set_page_config(page_title="Prototipo de Sistema de Acceso a
     Información Oficial de la Comunidad Autónoma de Andalucía con Té
     cnicas de PLN",
2 page_icon="",
3 layout="wide")
```

Adicionalmente, los estilos de la aplicación se cargan dinámicamente utilizando `st.markdown()`, asegurando una personalización adecuada, donde f es la lectura del fichero .css asignado desde el fichero de configuración de la aplicación .yaml (config.frontend.css_path):

```
1 st.markdown(f'<style>{f.read()}</style>', unsafe_allow_html=True)
```

6.7.2. Reinicio de la Aplicación

El proceso de reinicio se encuentra definido dentro de `_page_sidebar()`, específicamente en la función `_page_sidebar_reset_application()`. Si el usuario pulsa el botón **Reiniciar Aplicación**, se ejecuta el siguiente código:

```
1 if st.sidebar.button('Reiniciar Aplicación'):
```

Este proceso consiste en restablecer la aplicación a su estado inicial, vaciando todas las variables de estado de `streamlit` mediante la asignación de valores vacíos:

```
1 st.session_state[<variable>] = ""
```

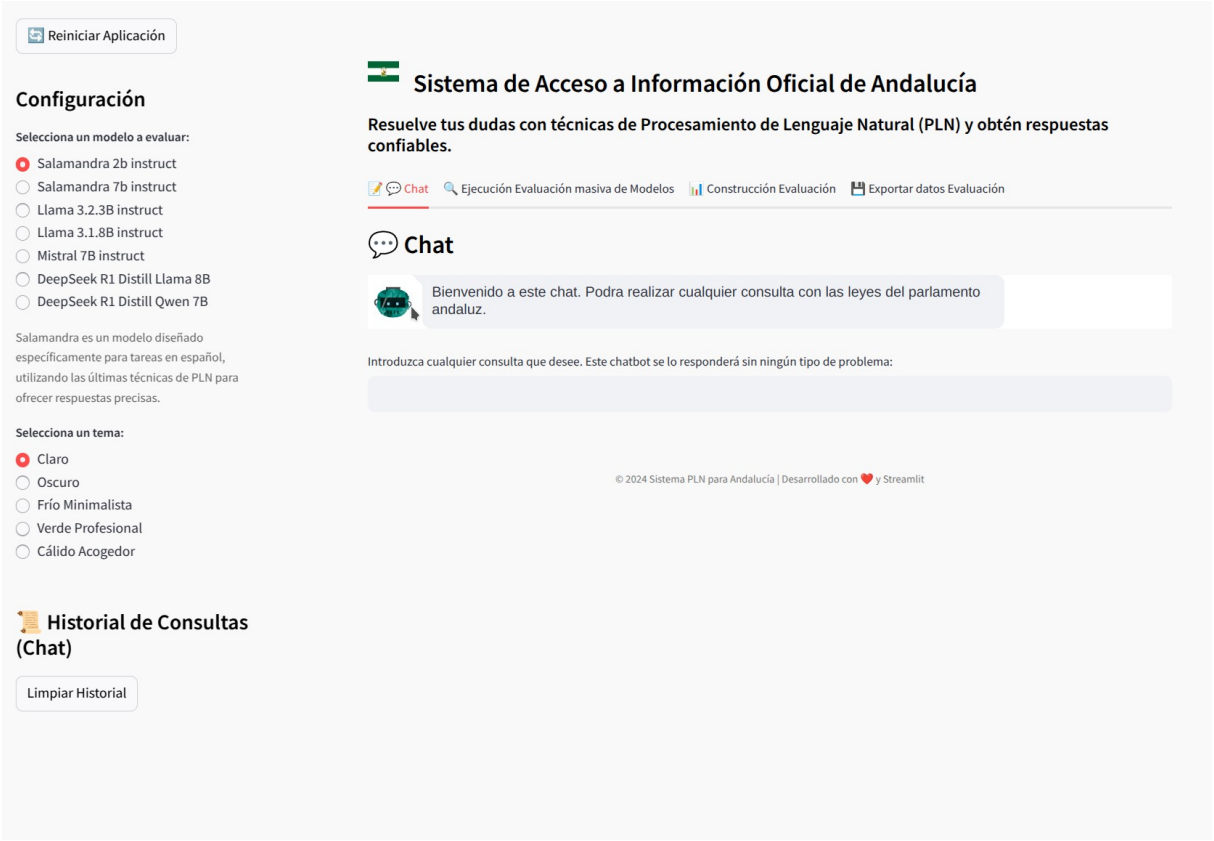


Figura 6.1: Página principal de la aplicación cliente.

Además, se establece por defecto el primer modelo y tema disponibles, y se fuerza la recarga de la interfaz con:

```
1 st.rerun()
```

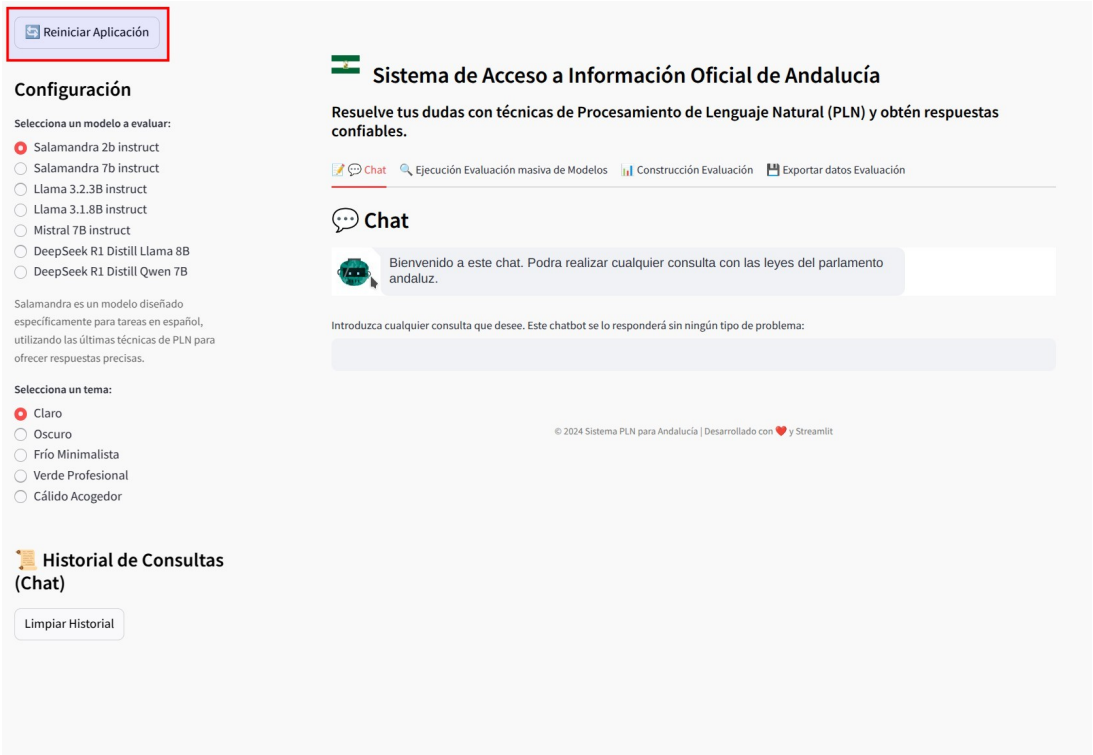


Figura 6.2: Reinicio de la aplicación cliente.

6.7.3. Selección del Modelo a Evaluar

Dentro de la barra lateral (`_page_sidebar()`), se presenta la opción de seleccionar el modelo a evaluar. La lista de modelos se obtiene del archivo de configuración `config.yaml` y se carga en un control de selección de `streamlit`:

```
1 model_keys = config.get_section_keys('models')
2 selected_model = st.sidebar.radio('Selecciona un modelo a evaluar:'
    , list(model_keys), key="selected_model")
```

Cuando el usuario selecciona un modelo, este se almacena en la variable de estado:

```
1 st.session_state['model_name'] = selected_model
```

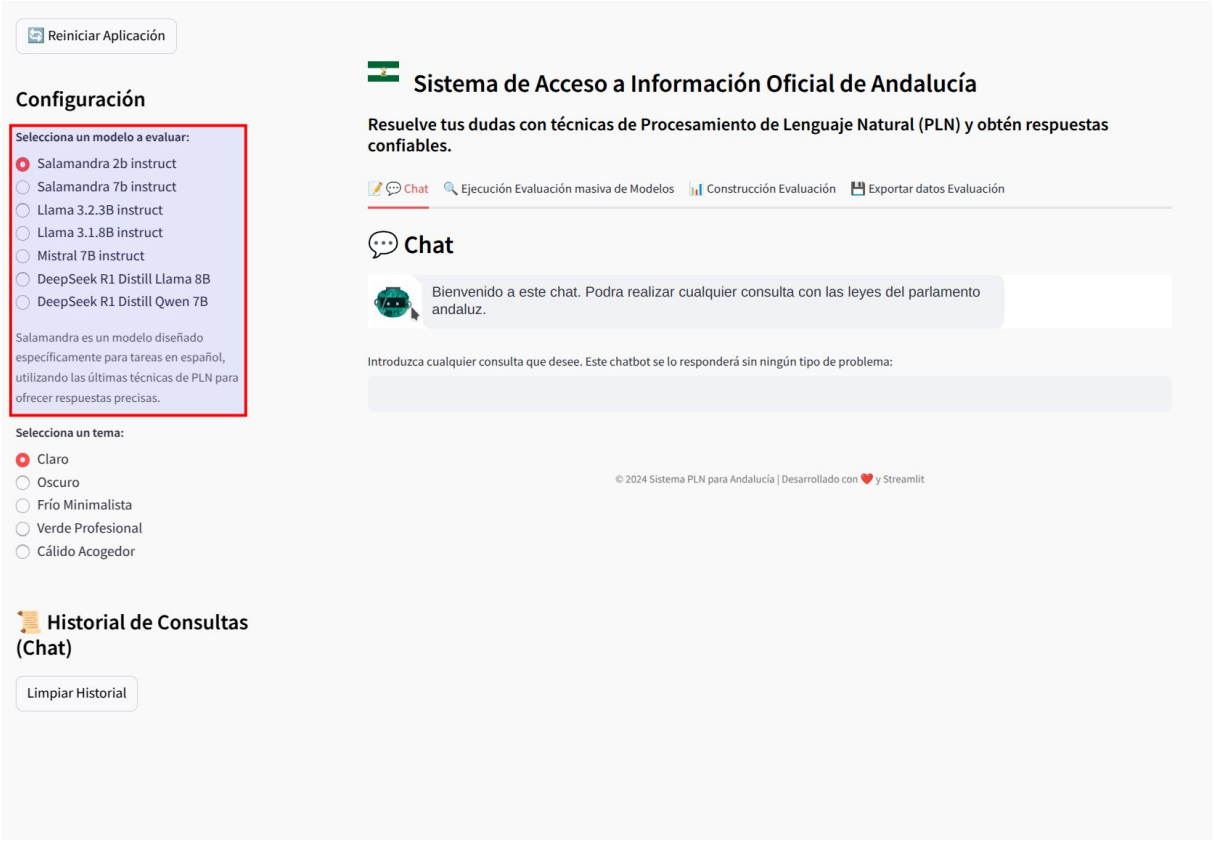


Figura 6.3: Selección del modelo para el chat.

6.7.4. Selección de Tema o Aspecto

De manera similar, la selección del tema se gestiona desde `_page_sidebar()`. Los temas disponibles se extraen del archivo de configuración y se presentan en un control de selección:

```
1 theme_by_default_tmp = config.frontend.themes
2 theme_name = st.sidebar.radio("Selecciona un tema:", list(
    theme_by_default_tmp.keys()), key="selected_theme")
```

Al seleccionar un tema, este queda registrado en la variable de estado de `streamlit st.session_state['model_name']`

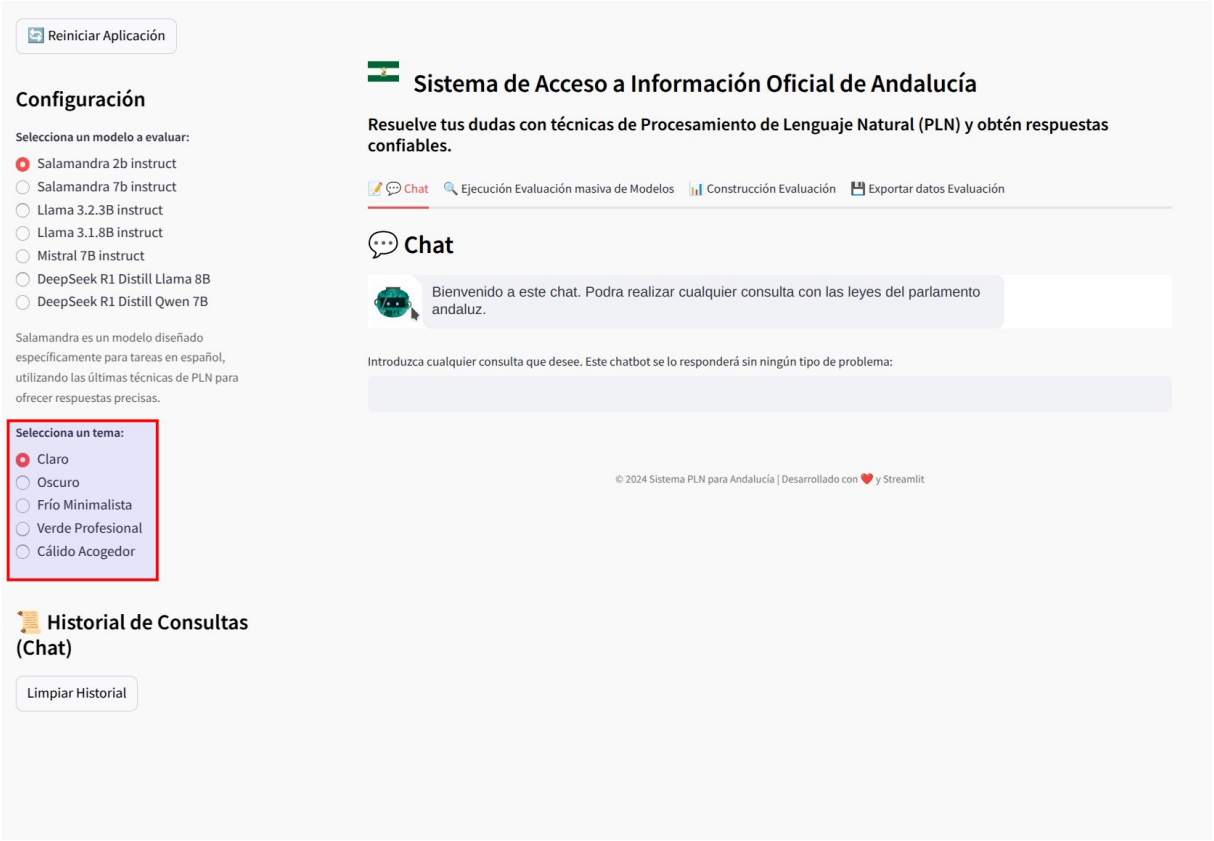


Figura 6.4: Tema/Aspecto de la aplicación cliente.

Figura 6.5: Cambio del tema (Verde Profesional) en la aplicación cliente.

6.7.5. Historial de Consultas

Desde la barra lateral (`_page_sidebar()`), se ofrece la opción de visualizar el historial de consultas. Este listado muestra las consultas realizadas y sus respuestas, recorriendo la lista almacenada en la variable de estado:

```
for item in st.session_state["history"]:
    st.write("Consulta:", item['query'])
    st.write("Respuesta:", item['response'])
```

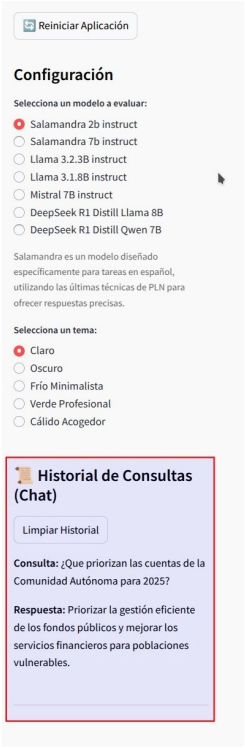


Figura 6.6: Historial de consultas en la aplicación cliente.

Adicionalmente, se incluye un botón para limpiar el historial, que al ser pulsado, vacía la variable de estado correspondiente:

```
if st.sidebar.button('Limpiar Historial'):
    st.session_state['history'] = []
```

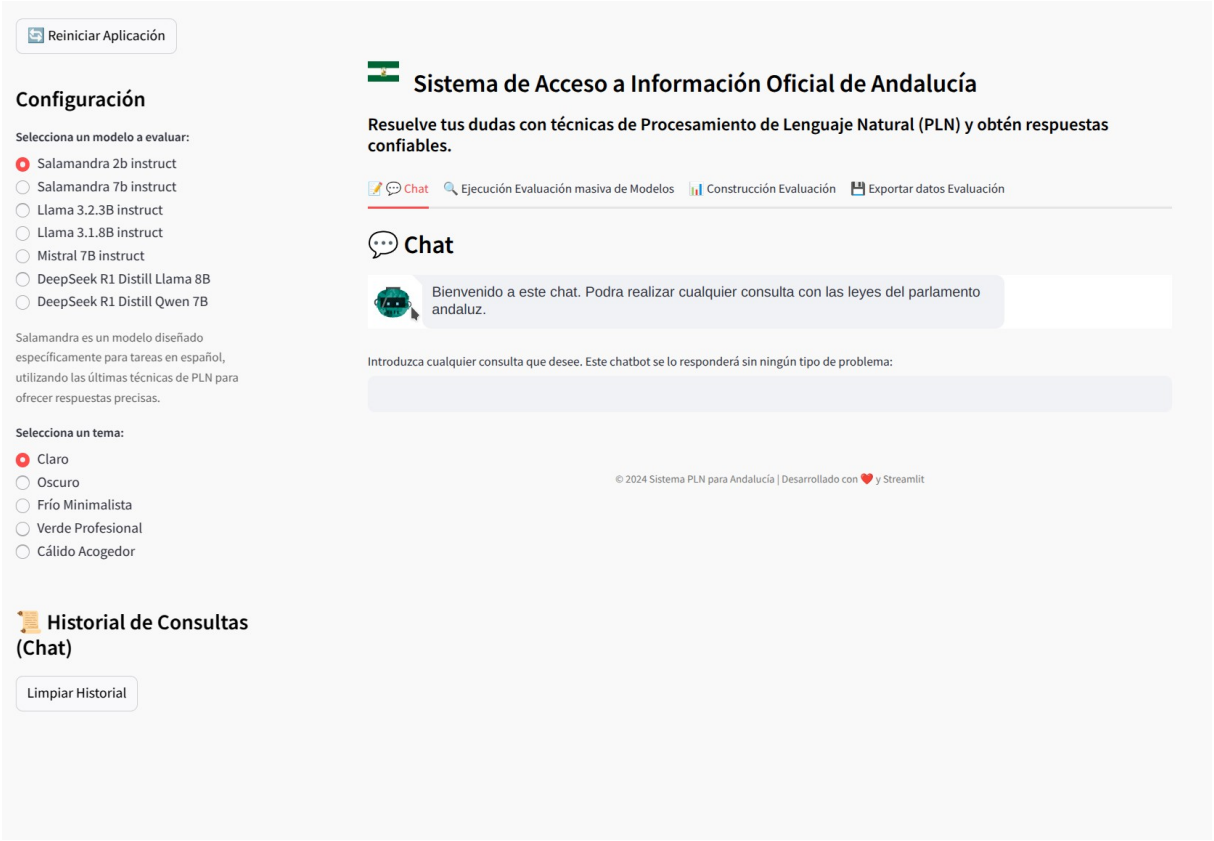


Figura 6.7: Limpieza del Historial de consultas en la aplicación cliente.

6.7.6. Chat

La interfaz principal de la aplicación incluye un control de pestañas (`tabs`), donde se agrupan las distintas funcionalidades del sistema. Este control es generado mediante `streamlit` utilizando la función:

```
1 tabs = st.tabs(["Chat", "Evaluación de Modelos",...])
```

Al seleccionar la primera pestaña `Chat`, se ejecuta automáticamente el módulo correspondiente:

```
1 with tabs[0]:
2     _display_messages()
3     st.text_input("Introduce tu consulta", key="user_input",
        on_change=process_input)
```

Este componente permite al usuario introducir mensajes o preguntas, activando la función `process_input()` al detectar cambios en la entrada.

Cuando el usuario envía una consulta, se realiza un preprocesamiento del texto antes de enviarlo a la API mediante la siguiente llamada:

```
1 with st.session_state["thinking_spinner"], st.spinner("Procesando
    ..."):
2     model_name = st.session_state["model_name"]
3     agent_response = requests.post(f"{config.frontend.base_api_url
        }:{config.frontend.base_api_port}/query",
4     json={"query": user_query, "model_name": model_name}).json()
```

Aquí, se recupera el modelo seleccionado y se envía la consulta al servidor Flask mediante una solicitud POST al endpoint `/query`, proporcionando como parámetros la consulta del usuario y el modelo seleccionado.

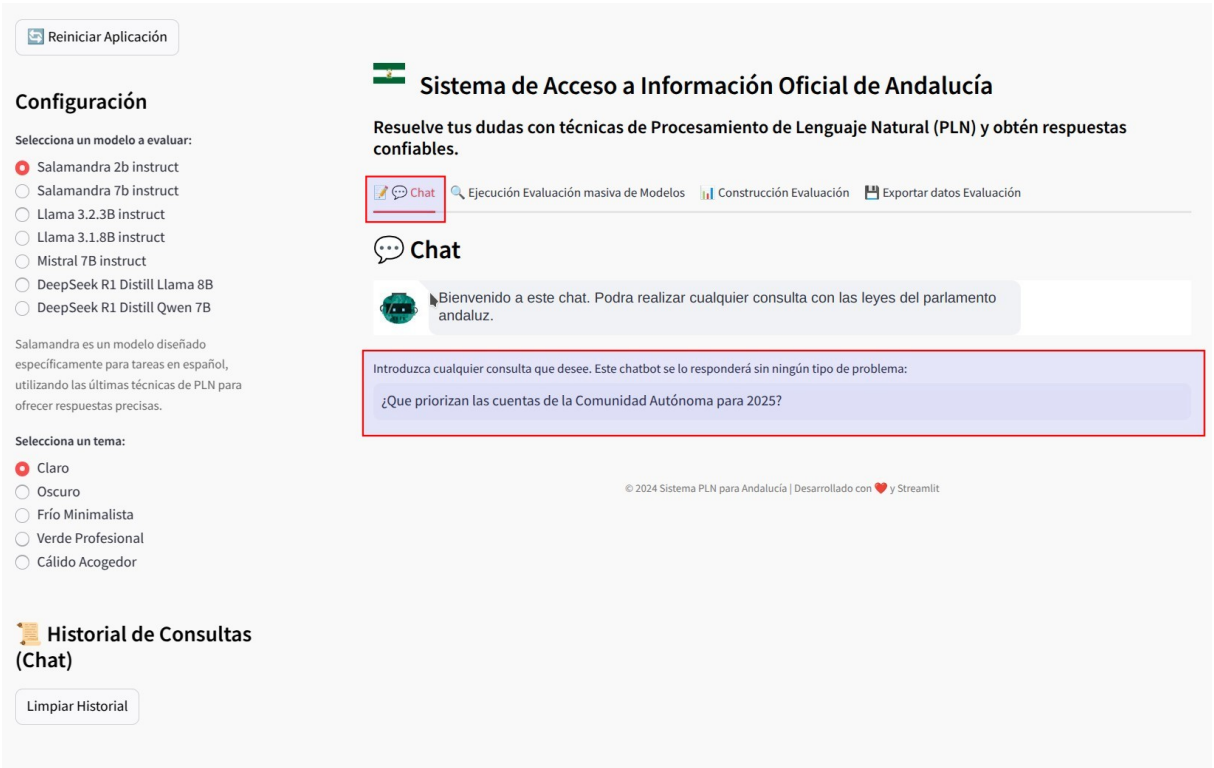


Figura 6.8: Introducción de consulta al chat.

6.7.6.1. Visualización de Respuestas

Una vez que el modelo genera la respuesta, el sistema actualiza la conversación y almacena la consulta en el historial. Para ello, los mensajes se agregan a la lista de mensajes de `streamlit` diferenciando entre usuario y asistente:

```
st.session_state["messages"].append((user_query, True))
st.session_state["messages"].append((agent_response["response_query
    "], False))
```

Si la respuesta incluye gráficos analíticos, estos se integran en el chat:

```
if agent_response["graphics"]:
    st.session_state["messages"].append(("graphics_chat",
        agent_response["graphics_data"]))
```

Asimismo, la consulta y la respuesta se almacenan en el historial de interacción:

```
st.session_state["history"].append({
    "query": user_query,
    "response": agent_response["response_query"]
})
```

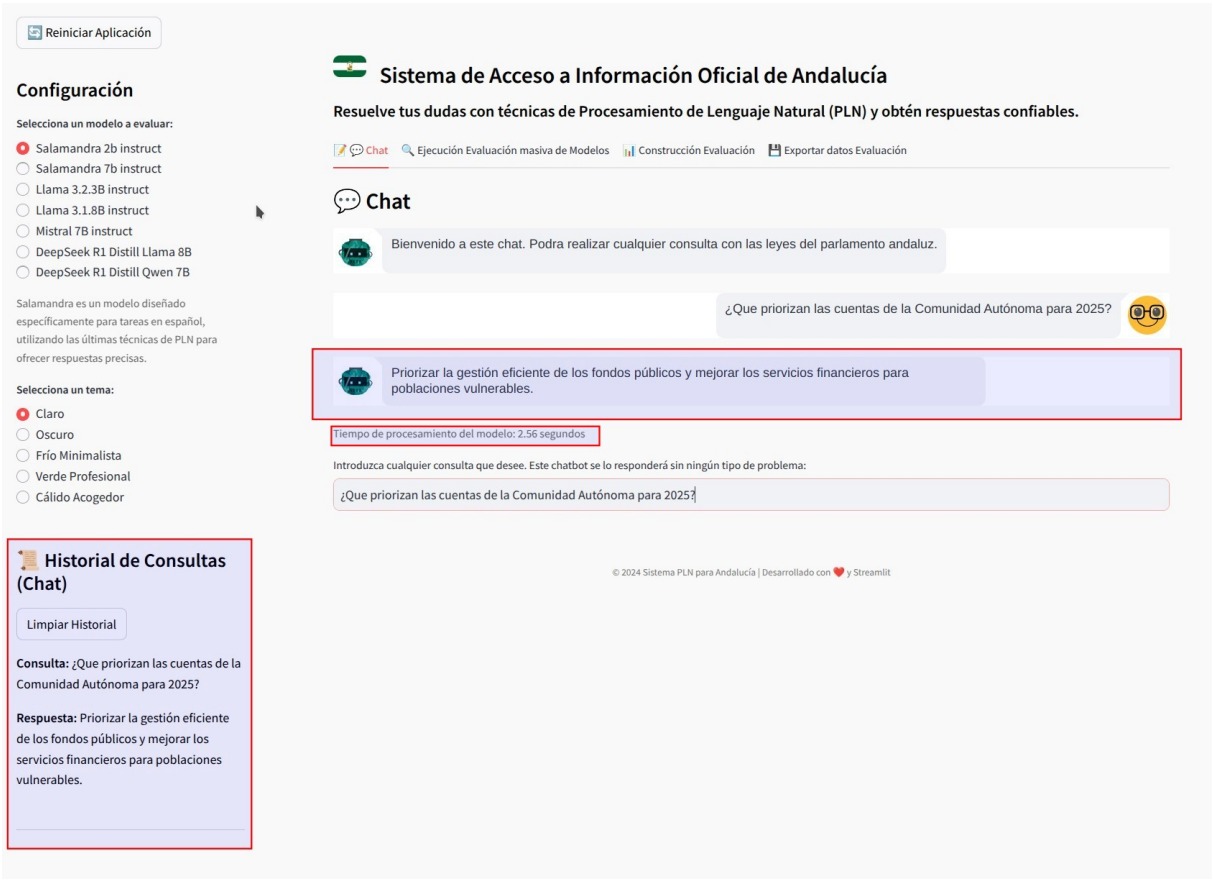


Figura 6.9: Respuesta del chat bot y adición en el historial.

Figura 6.10: Respuesta del chat bot con Gráfico comparativo de barras.

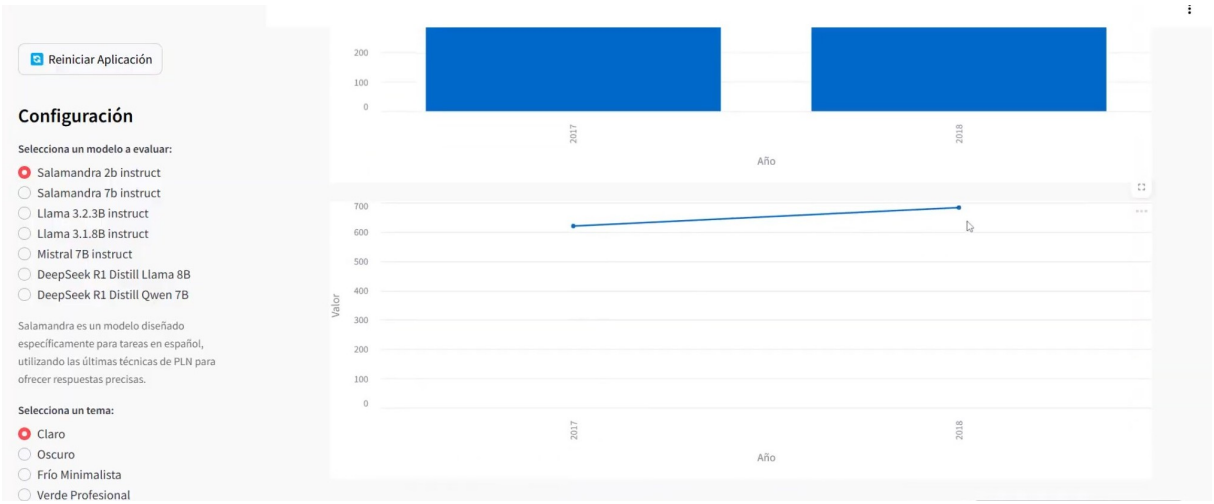


Figura 6.11: Respuesta del chat bot con Gráfico comparativo de líneas.

6.7.7. Ejecución de Evaluación Masiva de Modelos

Desde la interfaz principal, en la segunda pestaña del control de `tabs`, se encuentra la funcionalidad de evaluación de modelos de lenguaje. Al seleccionar la pestaña correspondiente, se ejecuta el módulo:

```
with tabs[1]:
    _evaluation_in_bulk()
```

Este proceso permite evaluar múltiples modelos con una serie de preguntas y respuestas de referencia (*source of truth*). La evaluación sigue los siguientes pasos:

6.7.7.1. Paso 1: Importación de Ficheros

Para cargar el conjunto de datos de evaluación, se proporciona un control de carga de archivos (`st.file_uploader()`) que permite seleccionar archivos CSV:

```
st.file_uploader("Sube un archivo CSV con preguntas y respuestas",
                 type=["csv"], accept_multiple_files=True)
```

Los archivos subidos se almacenan en una carpeta temporal del sistema para su posterior procesamiento. Si ya hay archivos cargados, se envían a la API mediante la

solicitud:

```
requests.post(f"{config.frontend.base_api_url}:{config.frontend.
    base_api_port}/upload", files=files)
```

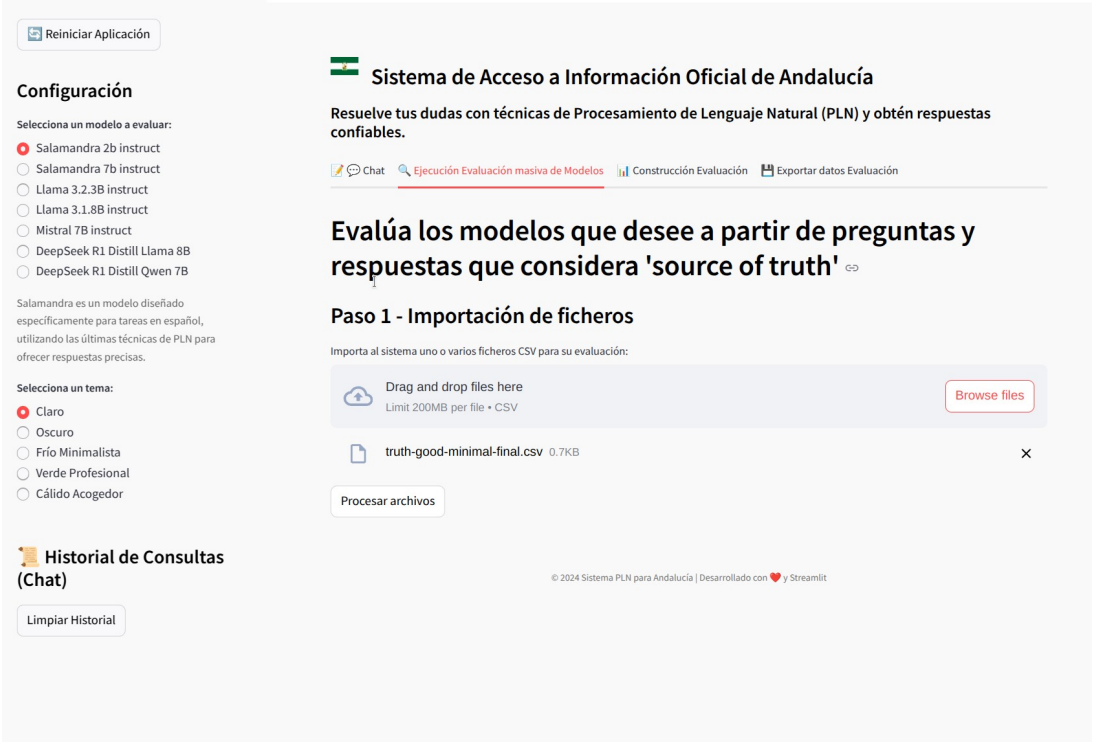


Figura 6.12: Subida de ficheros para evaluación masiva.

Esto permite cargar en memoria las preguntas y respuestas del *source of truth*, listas para la evaluación.

6.7.7.2. Paso 2: Selección de Modelos

Una vez importadas las preguntas y respuestas de referencia, el usuario puede seleccionar los modelos a evaluar utilizando el siguiente control:

```
selected_models = st.multiselect("Selecciona los modelos a evaluar:
    ",
                                 options=model_options,
                                 default=model_options)
```

Aquí, `model_options` representa la lista de modelos disponibles para evaluación.

Figura 6.13: Selección de Modelos para evaluación masiva.

6.7.7.3. Paso 3: Ejecución de la Evaluación

El usuario inicia la evaluación pulsando el botón **Evaluar**, lo que genera una llamada a la API:

```
requests.post(f"{config.frontend.base_api_url}:{config.frontend.
    base_api_port}/bulk_evaluate",
                  json={"models": selected_models, "querys_references":
                      loaded_questions})
```

Este proceso se **ejecuta en segundo plano para evitar bloqueos en la interfaz** y permitir que el usuario continúe utilizando el sistema. Además muestra un ID en

Figura 6.14: Evaluación Terminada.

el mensaje de color verde con la fecha y hora que se produce, que será finalmente llamado el fichero generado por esta ejecución.

6.7.8. Construcción de la Evaluación

Una vez ejecutado el proceso de evaluación, los resultados se presentarán de la siguiente manera pulsando el tab **Construcción de la Evaluación**. Cuando el usuario pulsa esta opción, se ejecuta la llamada a:

```
1 _display_evaluation_files()
```

y con ella se presenta los resultados en de la siguiente manera:

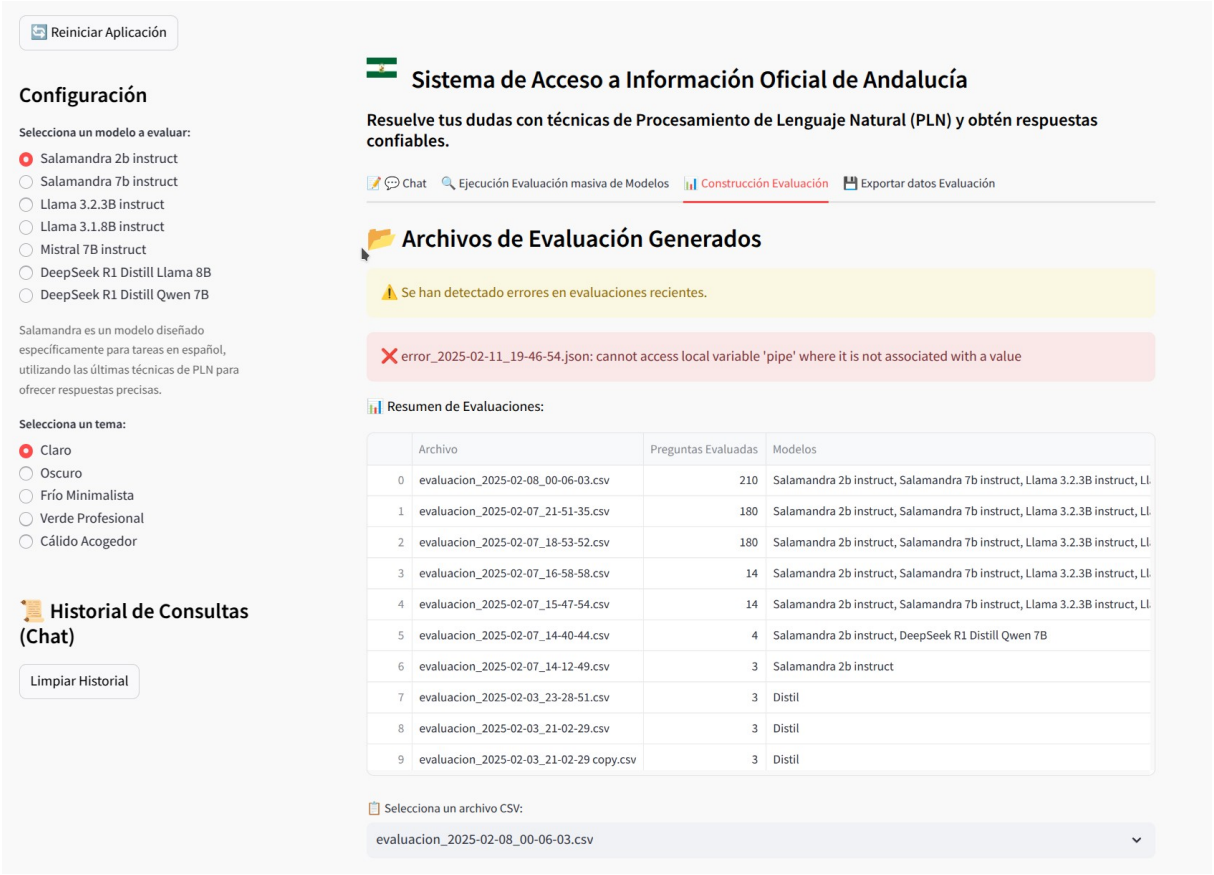


Figura 6.15: Construcción de la Evaluación de modelos.

En ella destaca dos partes:

- Un posible **listado de errores** de ejecución del proceso de evaluación anterior, mostrando el motivo del fallo. Esto lo realiza a su vez, llamando el frontend a una función propia llamada `_show_error_logs(error_files)` pasando como argumento el listado de todos los ficheros .json de errores:

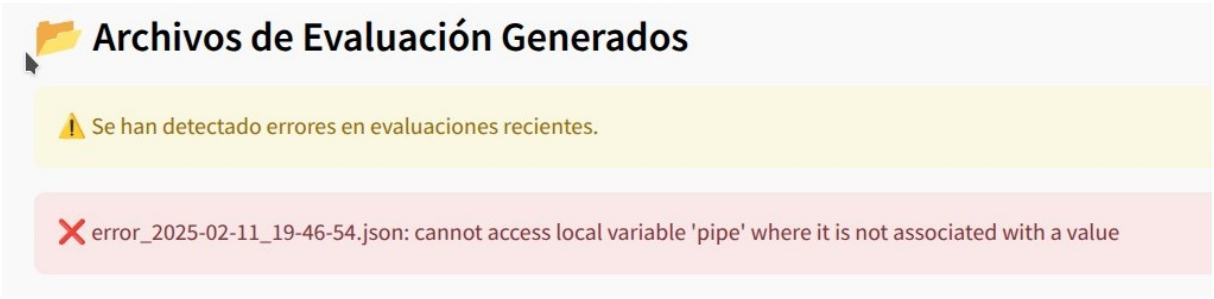
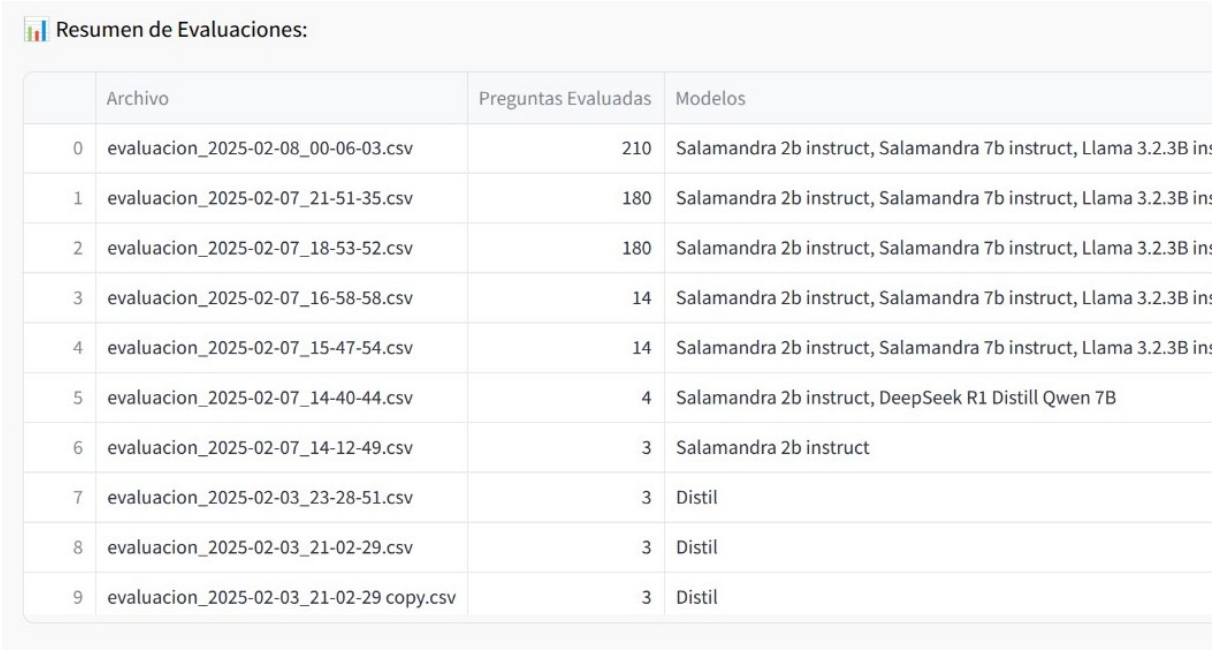


Figura 6.16: Listado de errores ejecución de evaluación.

- Y por otra parte aparece propiamente el **listado resumen de las evaluaciones** realizadas con éxito, donde aparecen el archivo, el total de preguntas evaluadas y los modelos utilizados. Para ello, se muestra llamando a la función `_extract_evaluation_summary(csv_files)` pasando como argumento el listado de todos los ficheros .csv generados de evaluación.



	Archivo	Preguntas Evaluadas	Modelos
0	evaluacion_2025-02-08_00-06-03.csv	210	Salamandra 2b instruct, Salamandra 7b instruct, Llama 3.2.3B ins
1	evaluacion_2025-02-07_21-51-35.csv	180	Salamandra 2b instruct, Salamandra 7b instruct, Llama 3.2.3B ins
2	evaluacion_2025-02-07_18-53-52.csv	180	Salamandra 2b instruct, Salamandra 7b instruct, Llama 3.2.3B ins
3	evaluacion_2025-02-07_16-58-58.csv	14	Salamandra 2b instruct, Salamandra 7b instruct, Llama 3.2.3B ins
4	evaluacion_2025-02-07_15-47-54.csv	14	Salamandra 2b instruct, Salamandra 7b instruct, Llama 3.2.3B ins
5	evaluacion_2025-02-07_14-40-44.csv	4	Salamandra 2b instruct, DeepSeek R1 Distill Qwen 7B
6	evaluacion_2025-02-07_14-12-49.csv	3	Salamandra 2b instruct
7	evaluacion_2025-02-03_23-28-51.csv	3	Distil
8	evaluacion_2025-02-03_21-02-29.csv	3	Distil
9	evaluacion_2025-02-03_21-02-29 copy.csv	3	Distil

Figura 6.17: Listado de resumen de evaluaciones.

Una vez identificada qué evaluación el usuario desea visualizar, aparece un desplegable para poder seleccionar la evaluación solicitada. Este desplegable es construido desde el frontend con la llamada a

```
selected_csv = st.selectbox("Selecciona un archivo CSV:", csv_files
    )
```

mostrando la siguiente en pantalla:

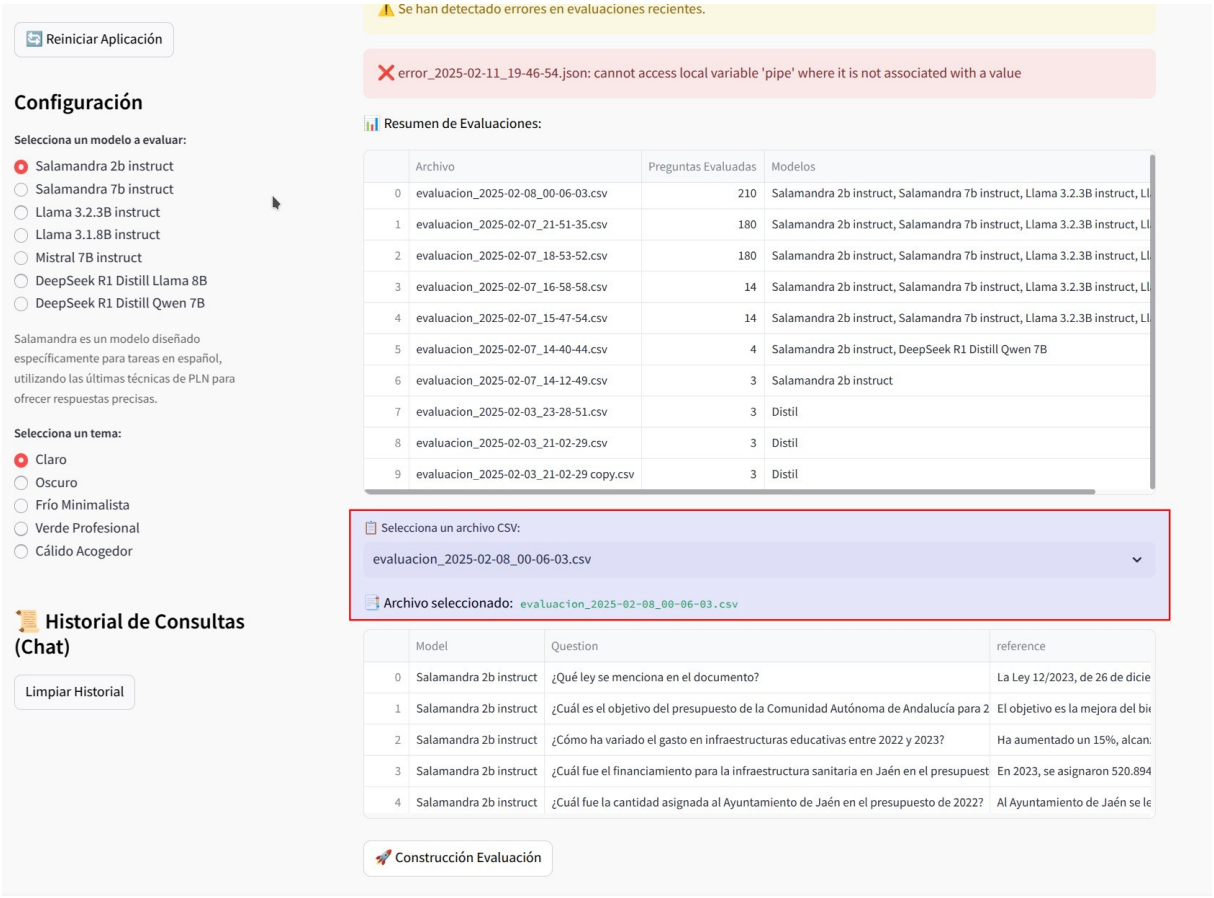


Figura 6.18: Selección de la evaluación.

Una vez seleccionado aparece un desglose a modo de información de la información que contiene esa evaluación, que se visualiza desde:

```
1 st.dataframe(pd.read_csv(selected_csv_path).head())
```

y el botón para empezar la construcción de la evaluación que se muestra desde:

```
1 st.button("Construcción␣Evaluación"))
```

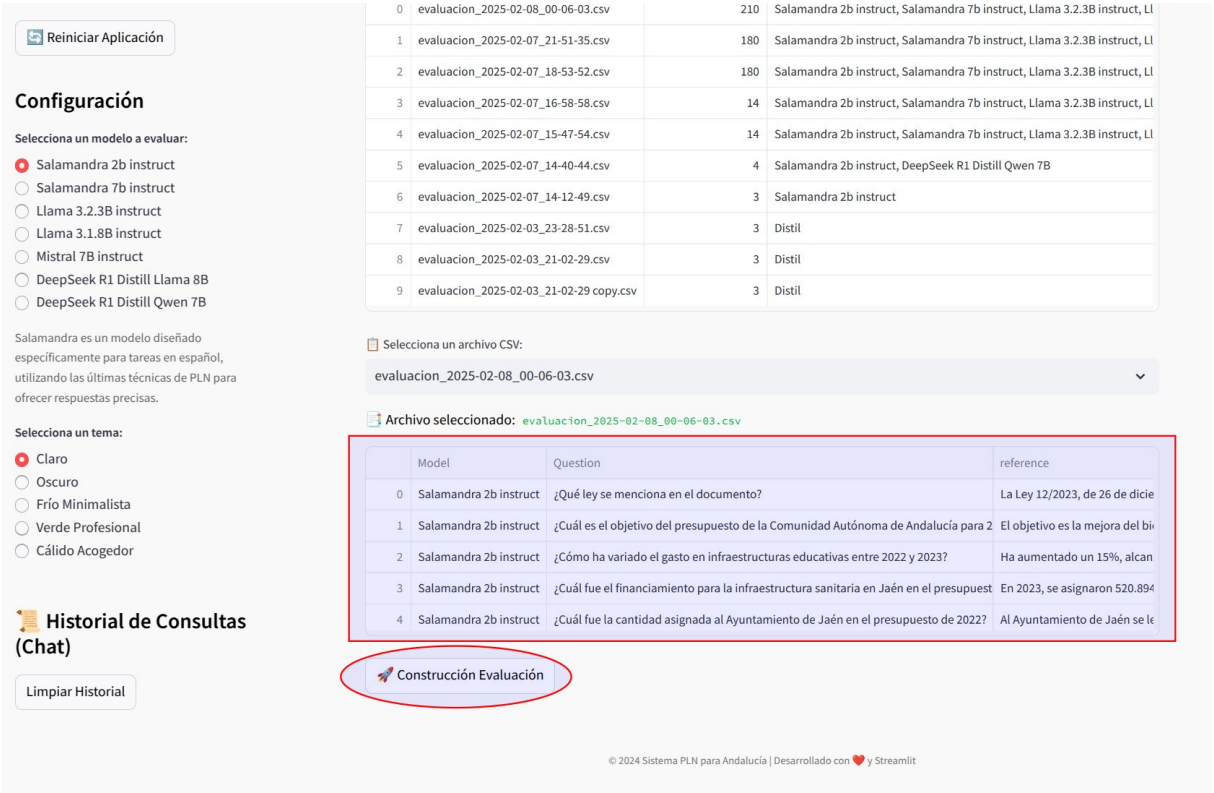


Figura 6.19: Resumen de Selección de la evaluación y Botón Construcción evaluación.

Una vez que el usuario pulsa el botón, se produce la evaluación de modelos que se divide este proceso en tres bloques fácilmente diferenciados, en los que cada uno de ellos aparecen una serie de gráficas que están contenidos por un contenedor:

```
1 with st.container():
```

6.7.8.1. Datos por métrica/modelo

Cada pregunta evaluada es analizada de manera independiente para cada modelo de lenguaje, permitiendo una evaluación detallada de su desempeño. Esta información se presenta en una tabla que muestra los valores obtenidos para cada métrica evaluada por modelo. El proceso de análisis se lleva a cabo mediante la instancia del objeto `GraphicsPlotter(df, models)`, donde:

- `df` representa los datos extraídos del archivo de evaluación en formato CSV.
- `models` contiene la lista de modelos de lenguaje seleccionados para la evaluación.

Dentro de la función `_draw_graphics_evaluation(df, models)`, se realiza la llamada al método `plotter.render_table_by_metrics()`, el cual se encarga de generar la tabla de análisis, proporcionando una visualización estructurada del rendimiento de cada modelo en función de las métricas definidas:

```
1 plotter.render_table_by_metrics()
```

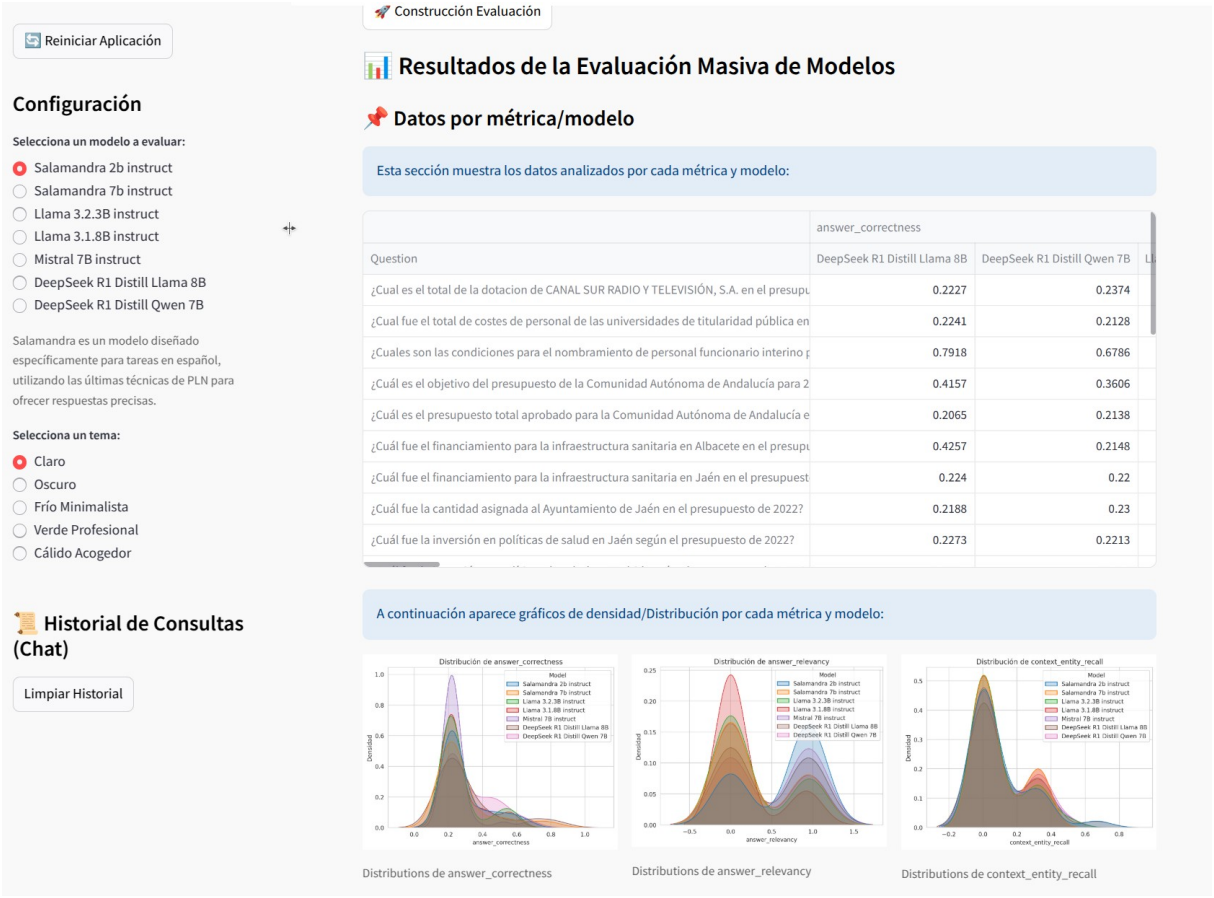


Figura 6.20: Tabla pivot/table Pregunta y evaluación por métricas/modelo.

Siguiendo con la ejecución, se muestran unos gráficos de **distribución** por métricas de RAGAS, llamando al método

```
1 plotter.render_distributions()
```

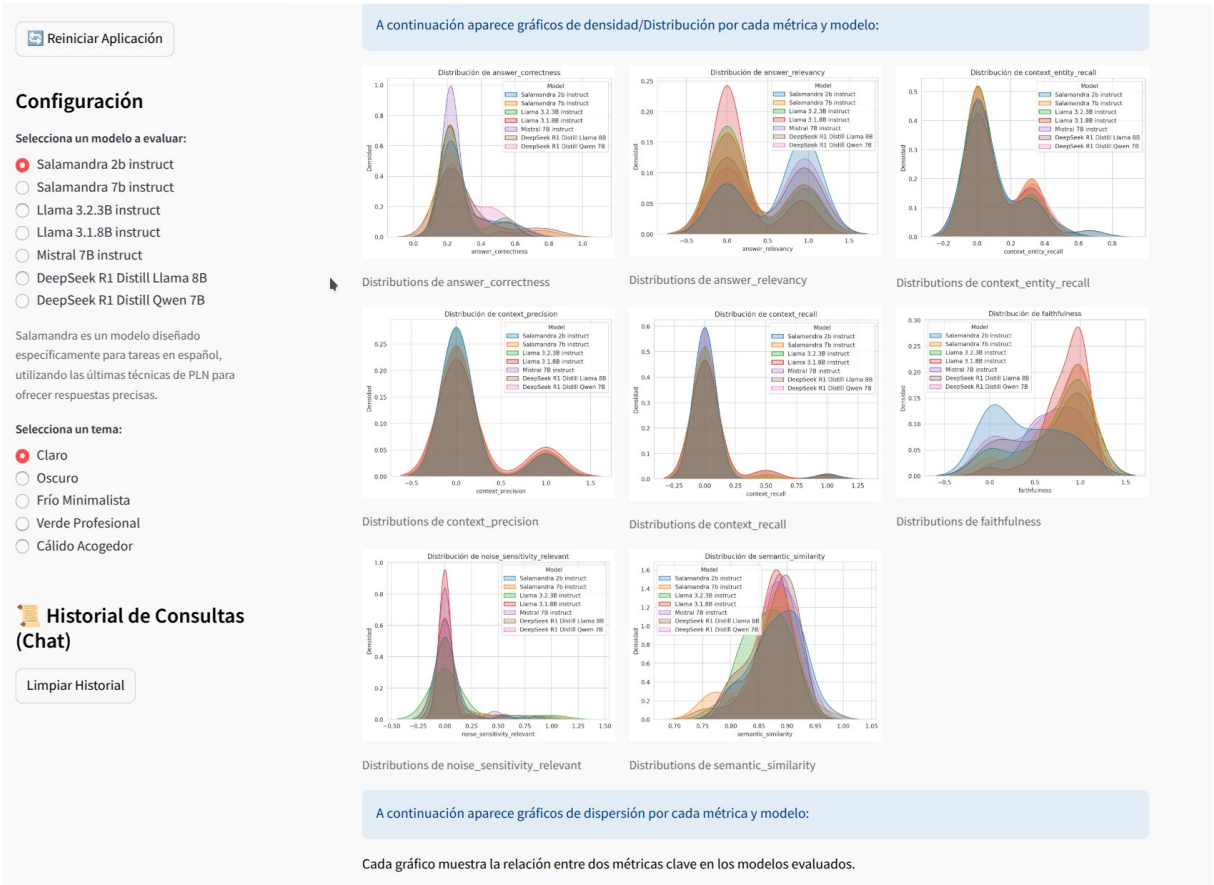


Figura 6.21: Gráficas de distribución por métricas y modelos.

De la misma manera, se muestran unos gráficos de **dispersión** por métricas de RAGAS, llamando al método

```
1 plotter.render_scatter_plots()
```

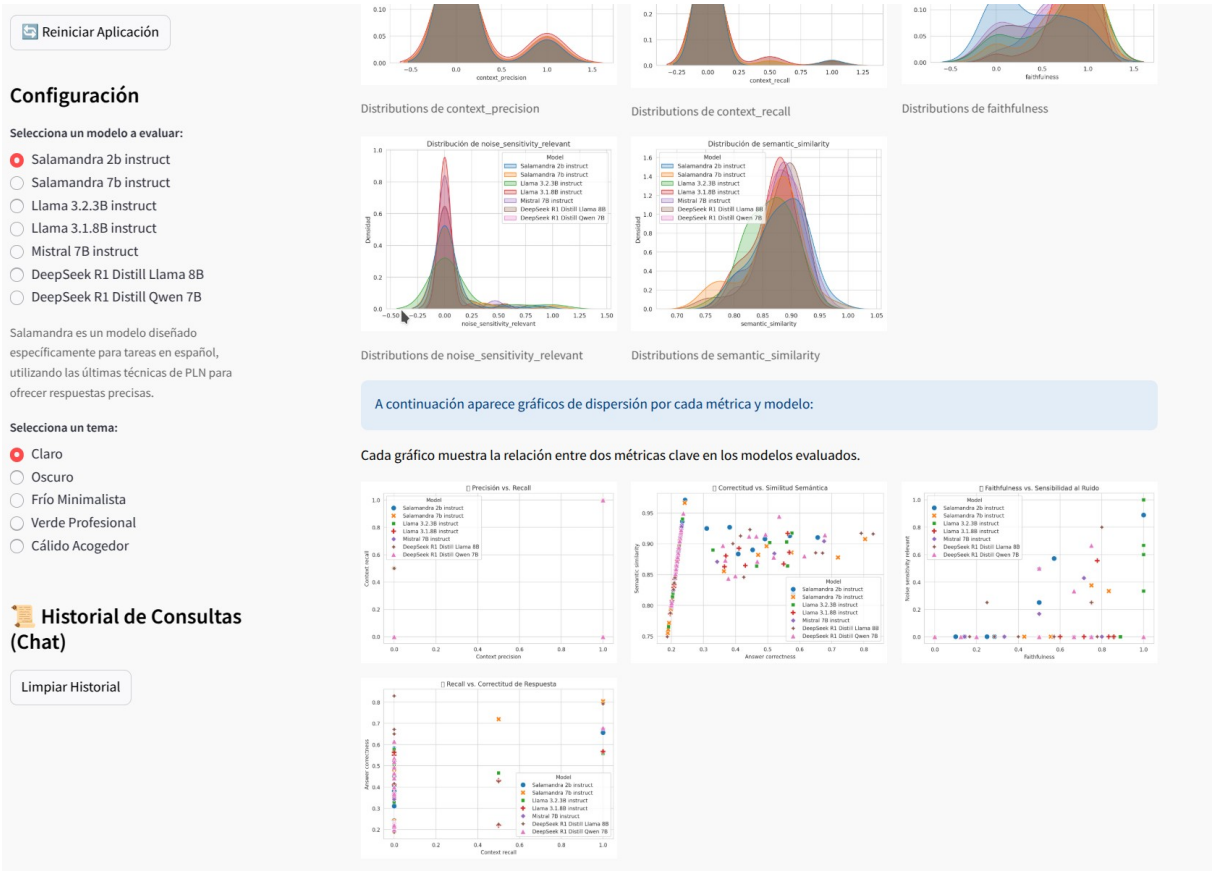


Figura 6.22: Gráficas de dispersión por métricas y modelos.

6.7.8.2. Evaluación Media Ponderada

Para obtener una visión general del desempeño de los modelos, se calcula un promedio ponderado de las métricas obtenidas por cada modelo. Esta tabla es mostrada, llamando a la función:

```
1 plotter.render_table_summary()
```

Además aparece una gráficas de barras, llamando a la función:

```
1 plotter.render_bar_chart_summary(table_data)
```

donde `table_data` es el valor de los datos ponderados previamente calculados:

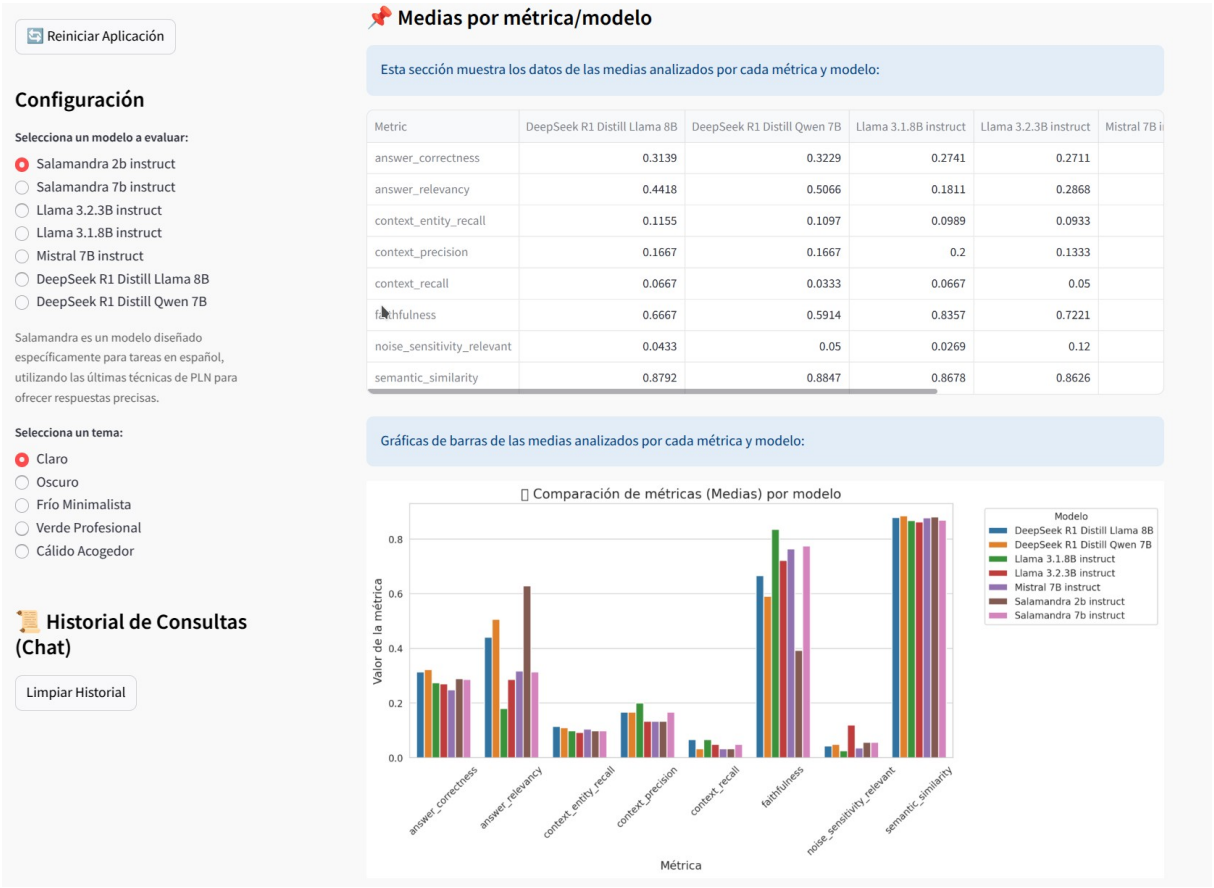


Figura 6.23: Gráficas de barras por promedio de métricas y modelos.

A continuación aparecen gráficas de radar, mapa de calor, dispersión y tendencias a nivel de medias, llamando a la función:

```
plotter.render_combined_bar_line_chart()
plotter.render_radar_chart()
plotter.render_correlation_heatmap()
plotter.render_scatter_plots_summary()
plotter.render_regression_plots()
```

Figura 6.24: Gráficas de barras lineas y radar por promedio de métricas y modelos.

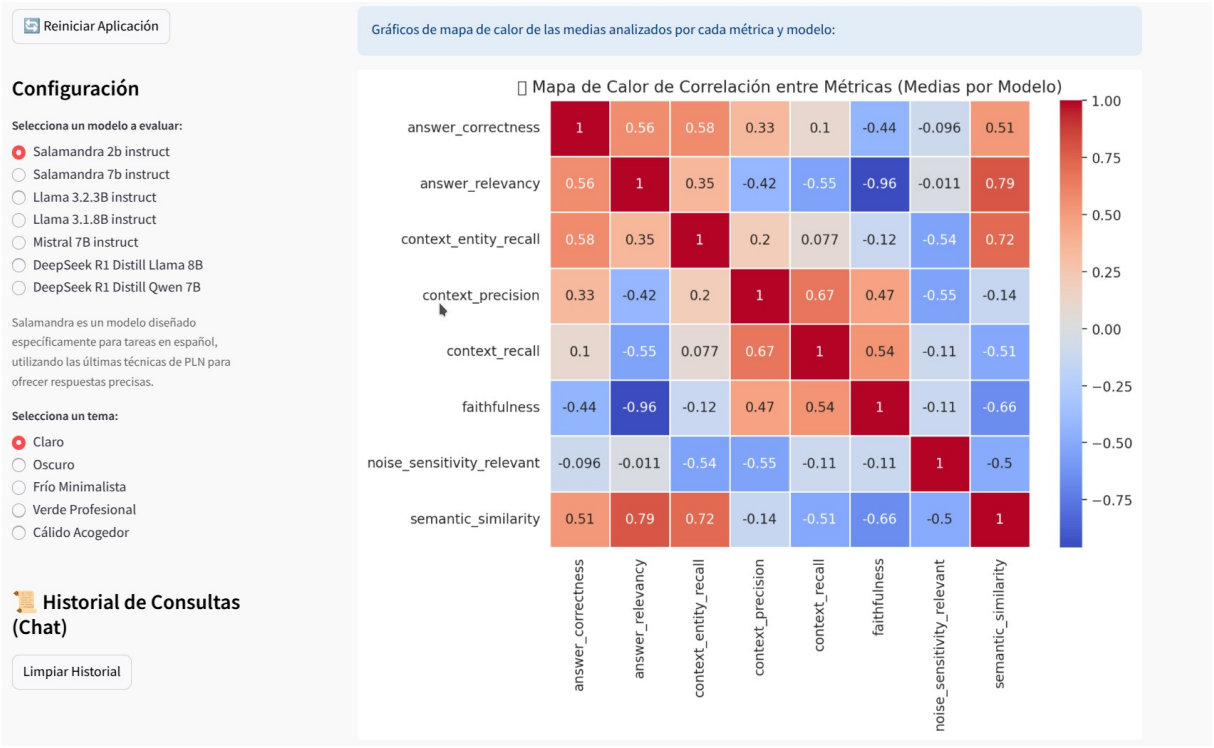


Figura 6.25: Gráfica de calor por promedio de métricas y modelos.

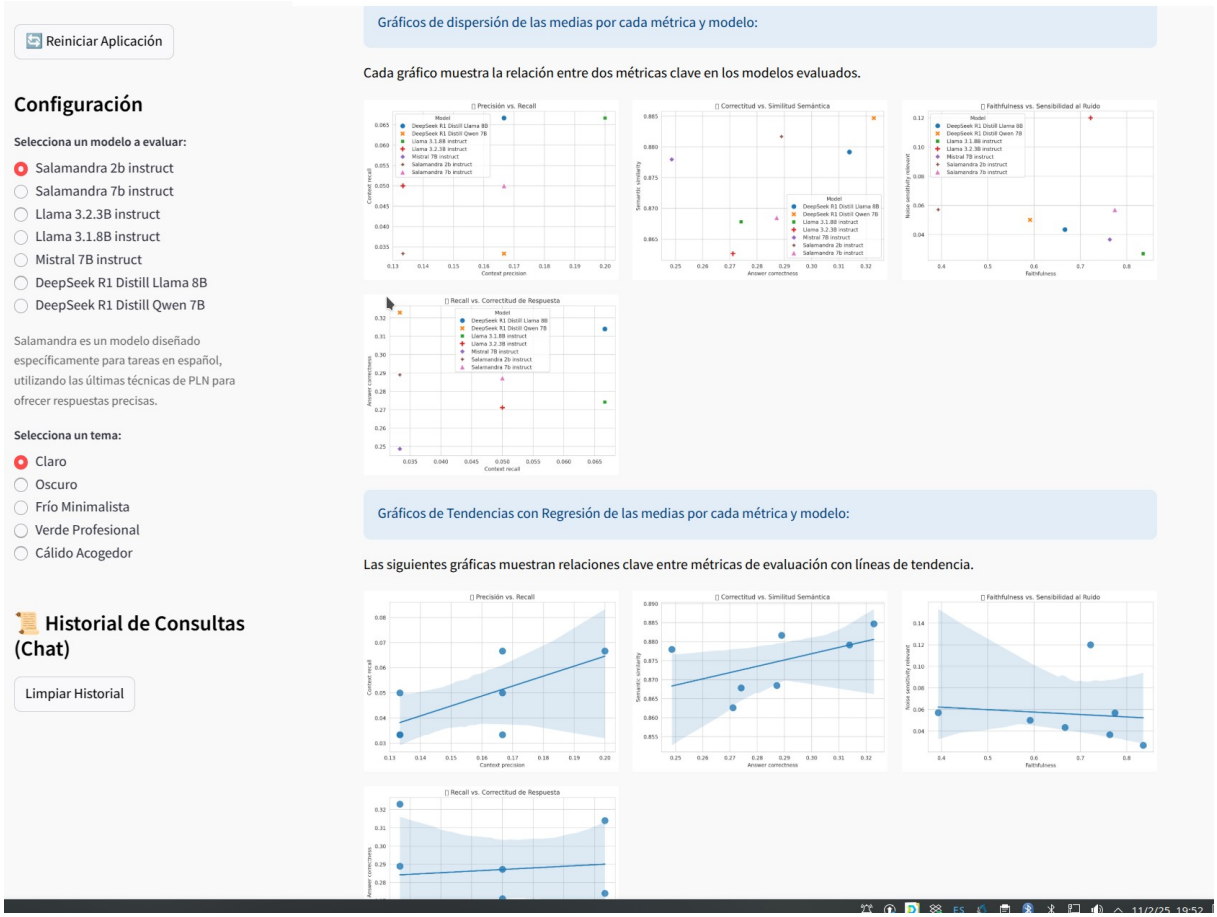


Figura 6.26: Gráfica de dispersión y Regresión por promedio de métricas y modelos.

6.7.8.3. Evaluación por Pesos

Por último, en función de la importancia de cada métrica, se pueden ajustar los pesos relativos de cada una para realizar un análisis más preciso. Estos pesos son configurados desde el fichero de configuración del sistema (`config.yaml`) y son mostrados en pantalla. También desde este punto, aparecen gráficas de radar, mapa de calor, dispersión y tendencias a nivel de medias ponderadas, llamando a la función:

```
plotter.render_weights()
plotter.render_radar_chart_overall()
plotter.render_distribution_overall()
plotter.render_correlation_heatmap_overall()
render_overall_score_correlation_heatmap()
```

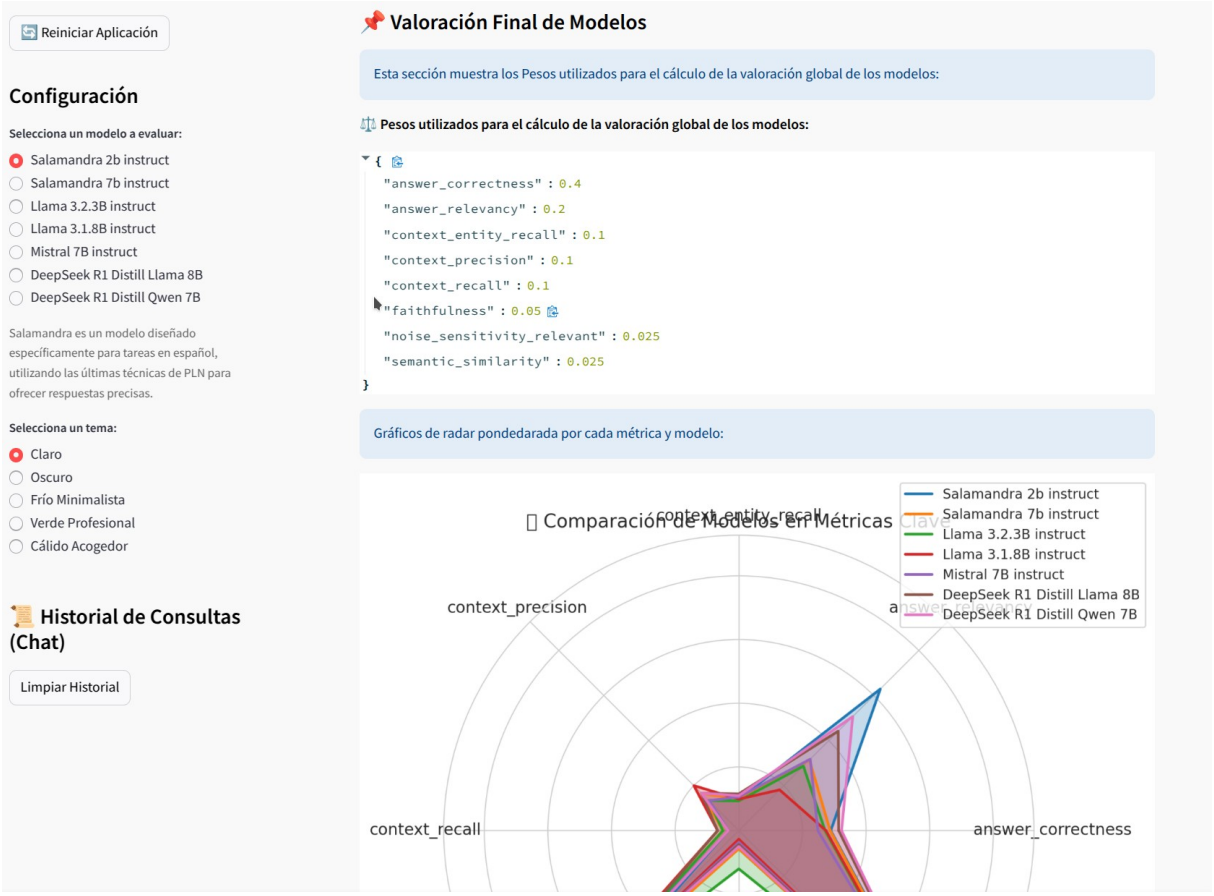


Figura 6.27: Tabla de pesos por métricas y gráfica de radas por promedio ponderado de pesos.

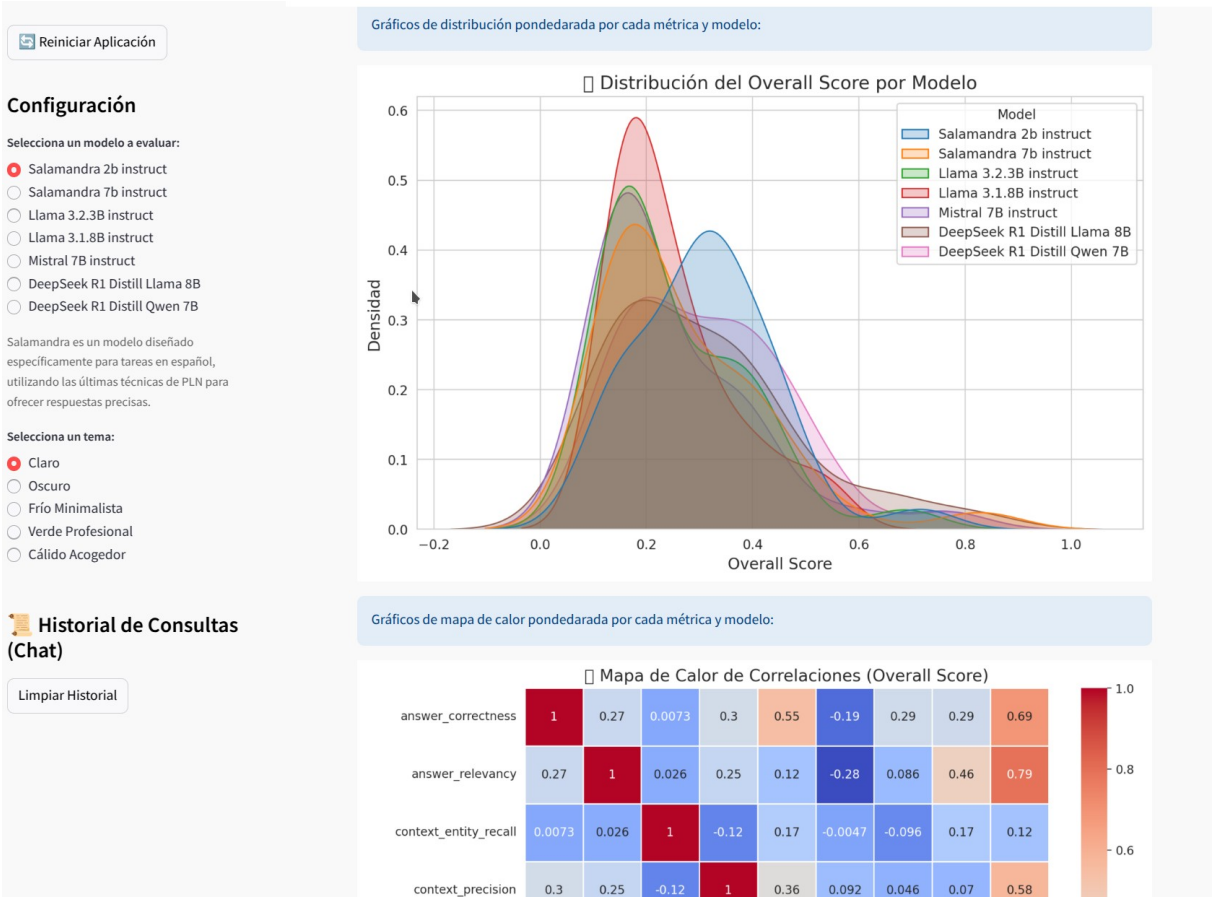


Figura 6.28: Gráfica de distribución por promedio ponderado de pesos.

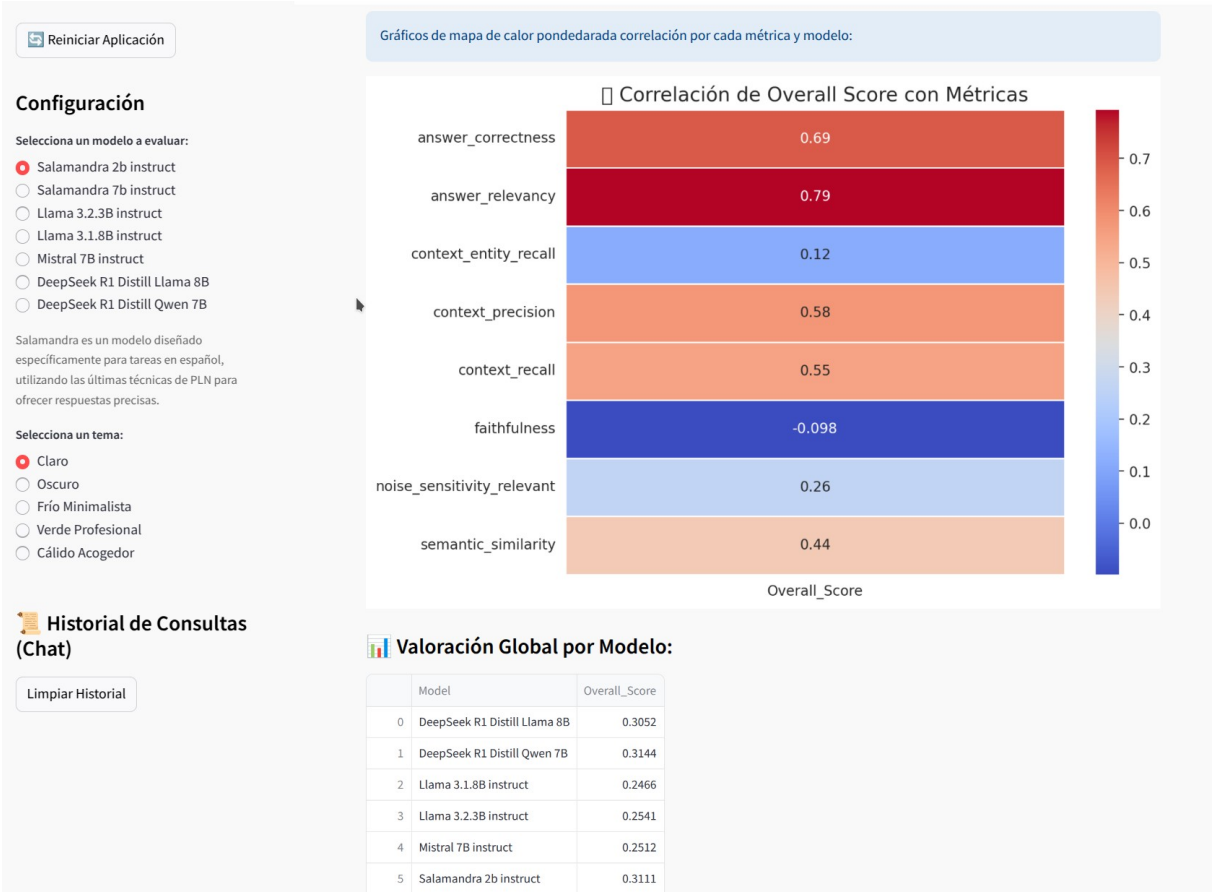


Figura 6.29: Gráfica de calor por promedio ponderado de pesos.

Por ultimo muestra la valoración global por modelo asigna en función de los valores de las evaluaciones aplicando estas medias ponderadas, llamando a la función:

```
1 plotter.render_overall_score_table_with_winner()
```

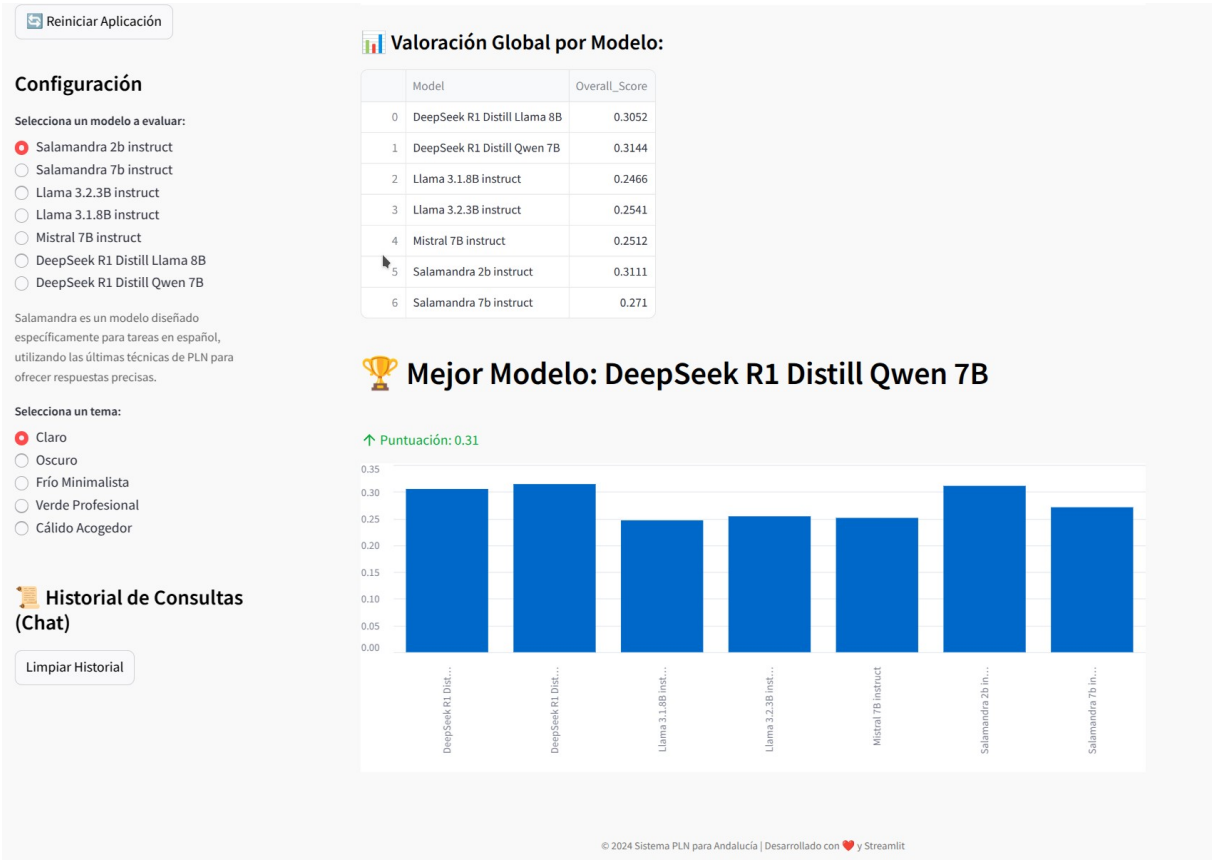


Figura 6.30: Tabla Valoración Global por pesos por modelos y Gráfica de comparación final.

6.7.9. Exportación de la Evaluación

Los resultados pueden exportarse en distintos formatos (CSV o JSON) para su posterior análisis. Esto se logra mediante la funcionalidad de exportación de `streamlit`:

```
st.download_button("Descargar Resultados", data=csv_data, file_name="resultados.csv", mime="text/csv")
```

Esto permite al usuario descargar los datos generados y analizarlos externamente.

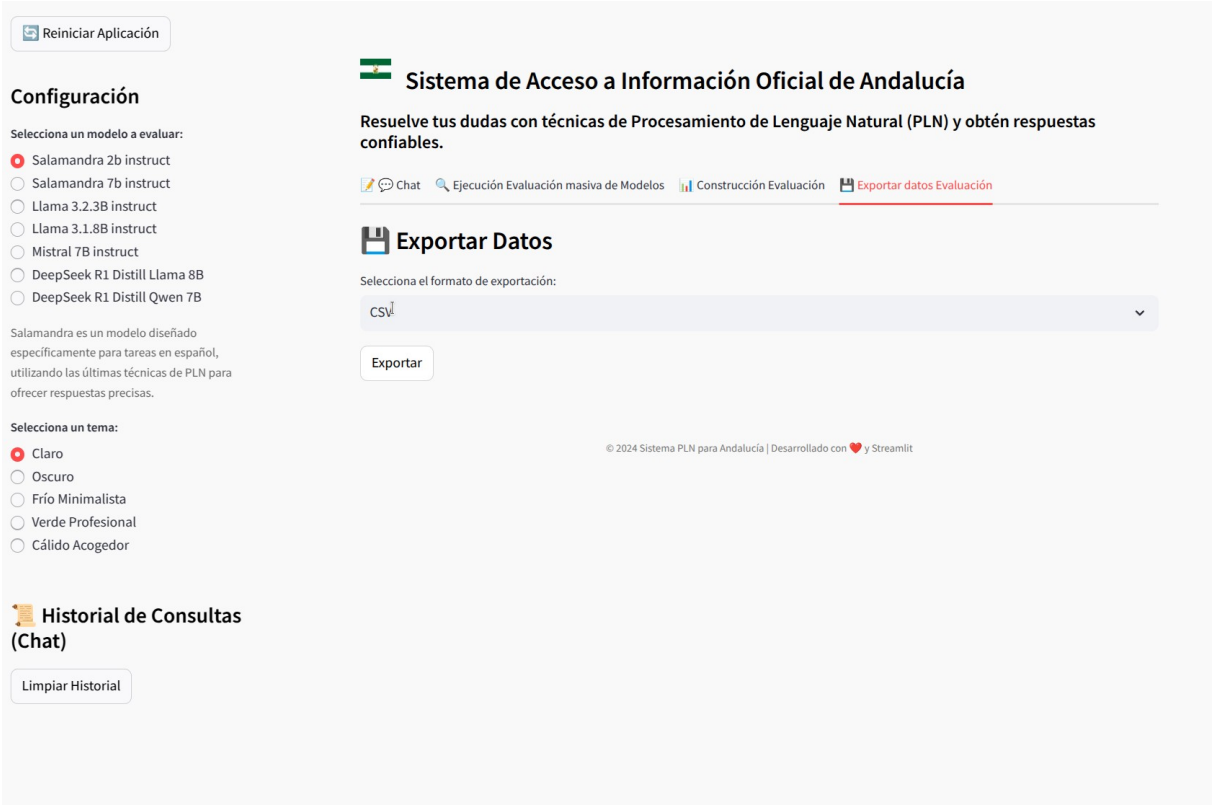


Figura 6.31: Exportación de los resultados de la evaluación.

Figura 6.32: Exportación de los resultados de la evaluación.

Figura 6.33: Exportación de los resultados de la evaluación.

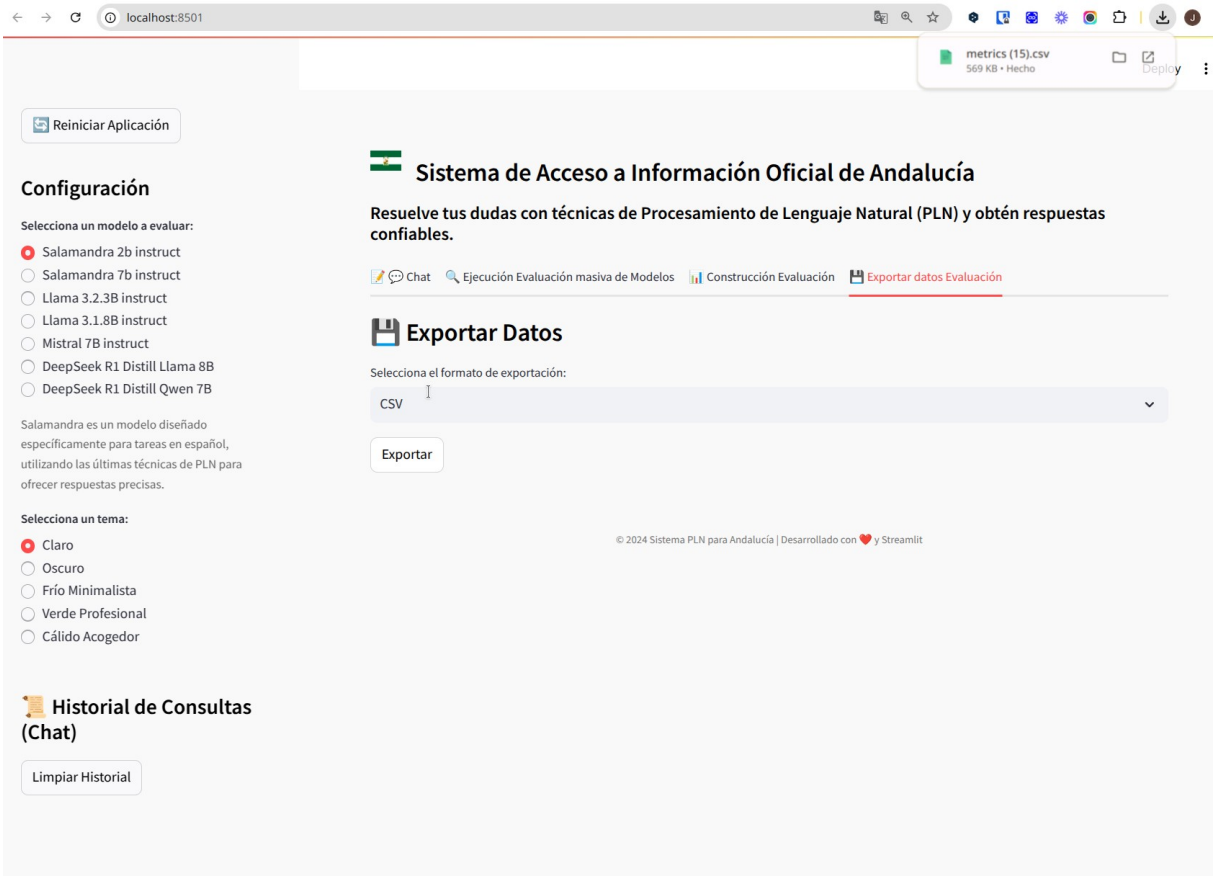
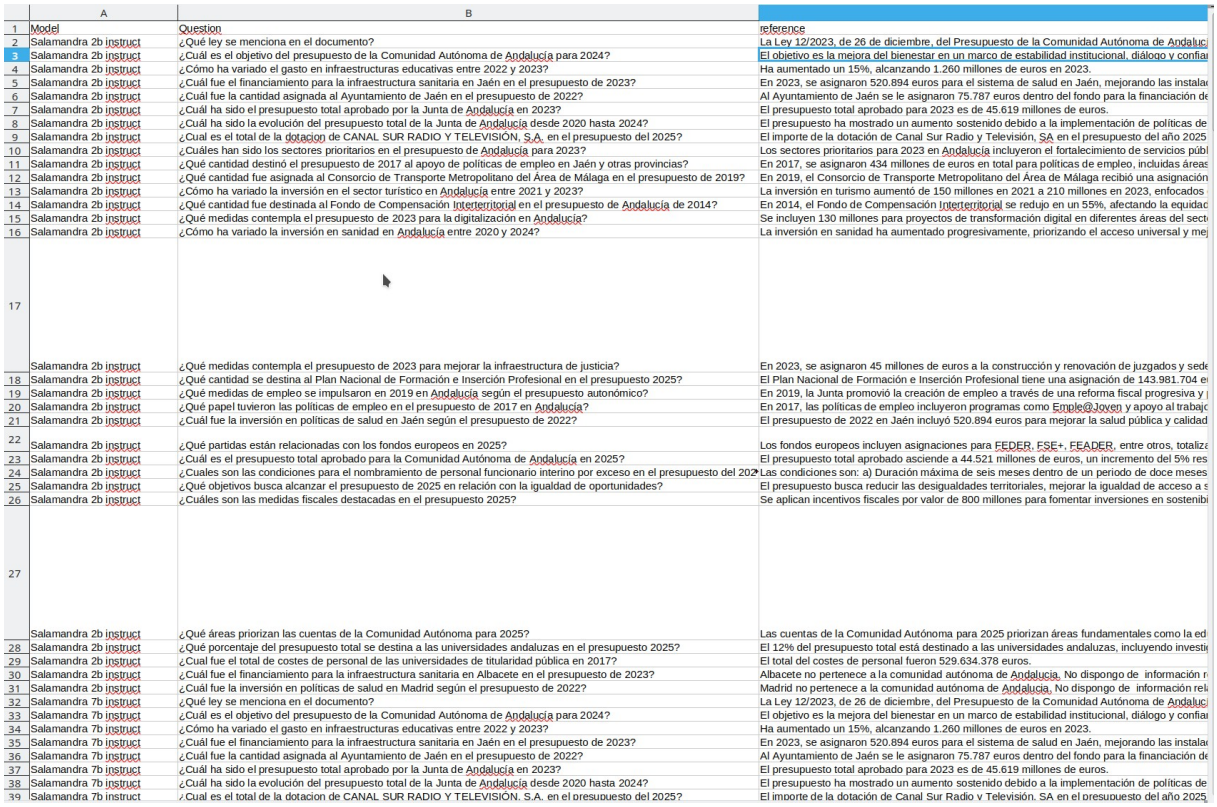


Figura 6.34: Exportación de los resultados de la evaluación.



	A	B	
1	Model	Question	reference
2	Salamandra 2b instruct	¿Qué ley se menciona en el documento?	La Ley 12/2023, de 26 de diciembre, del Presupuesto de la Comunidad Autónoma de Andaluc
3	Salamandra 2b instruct	¿Cuál es el objetivo del presupuesto de la Comunidad Autónoma de Andalucía para 2024?	El objetivo es la mejora del bienestar en un marco de estabilidad institucional, diálogo y confia
4	Salamandra 2b instruct	¿Cómo ha variado el gasto en infraestructuras educativas entre 2022 y 2023?	Ha aumentado un 15%, alcanzando 1.260 millones de euros en 2023.
5	Salamandra 2b instruct	¿Cuál fue el financiamiento para la infraestructura sanitaria en Jaén en el presupuesto de 2023?	En 2023, se asignaron 520.894 euros para el sistema de salud en Jaén, mejorando las instala
6	Salamandra 2b instruct	¿Cuál fue la cantidad asignada al Ayuntamiento de Jaén en el presupuesto de 2022?	Al Ayuntamiento de Jaén se le asignaron 75.787 euros dentro del fondo para la financiación de
7	Salamandra 2b instruct	¿Cuál ha sido el presupuesto total aprobado por la Junta de Andalucía en 2023?	El presupuesto total aprobado para 2023 es de 45.619 millones de euros.
8	Salamandra 2b instruct	¿Cuál ha sido la evolución del presupuesto total de la Junta de Andalucía desde 2020 hasta 2024?	El presupuesto ha mostrado un aumento sostenido debido a la implementación de políticas de
9	Salamandra 2b instruct	¿Cual es el total de la dotacion de CANAL SUR RADIO Y TELEVISIÓN, S.A. en el presupuesto del 2025?	El importe de la dotación de Canal Sur Radio y Televisión, SA en el presupuesto del año 2025
10	Salamandra 2b instruct	¿Cuáles han sido los sectores prioritarios en el presupuesto de Andalucía para 2023?	Los sectores prioritarios para 2023 en Andalucía incluyeron el fortalecimiento de servicios públ
11	Salamandra 2b instruct	¿Qué cantidad destinó el presupuesto de 2017 al apoyo de políticas de empleo en Jaén y otras provincias?	En 2017, se asignaron 434 millones de euros en total para políticas de empleo, incluidas áreas
12	Salamandra 2b instruct	¿Qué cantidad fue asignada al Consorcio de Transporte Metropolitano del Área de Málaga en el presupuesto de 2019?	En 2019, el Consorcio de Transporte Metropolitano del Área de Málaga recibió una asignación
13	Salamandra 2b instruct	¿Cómo ha variado la inversión en el sector turístico en Andalucía entre 2021 y 2023?	La inversión en turismo aumentó de 150 millones en 2021 a 210 millones en 2023, enfocados
14	Salamandra 2b instruct	¿Qué cantidad fue destinada al Fondo de Compensación Interterritorial en el presupuesto de Andalucía de 2014?	En 2014, el Fondo de Compensación Interterritorial se redujo en un 55%, afectando la equidad
15	Salamandra 2b instruct	¿Qué medidas contempla el presupuesto de 2023 para la digitalización en Andalucía?	Se incluyen 130 millones para proyectos de transformación digital en diferentes áreas del secto
16	Salamandra 2b instruct	¿Cómo ha variado la inversión en sanidad en Andalucía entre 2020 y 2024?	La inversión en sanidad ha aumentado progresivamente, priorizando el acceso universal y mej
17	Salamandra 2b instruct	¿Qué medidas contempla el presupuesto de 2023 para mejorar la infraestructura de justicia?	En 2023, se asignaron 45 millones de euros a la construcción y renovación de juzgados y sede
18	Salamandra 2b instruct	¿Qué cantidad se destina al Plan Nacional de Formación e Inserción Profesional en el presupuesto 2025?	El Plan Nacional de Formación e Inserción Profesional tiene una asignación de 143.981.704 e
19	Salamandra 2b instruct	¿Qué medidas de empleo se impulsaron en 2019 en Andalucía según el presupuesto autonómico?	En 2019, la Junta promovió la creación de empleo a través de una reforma fiscal progresiva y
20	Salamandra 2b instruct	¿Qué papel tuvieron las políticas de empleo en el presupuesto de 2017 en Andalucía?	En 2017, las políticas de empleo incluyeron programas como Emple@Joven y apoyo al trabajo
21	Salamandra 2b instruct	¿Cuál fue la inversión en políticas de salud en Jaén según el presupuesto de 2022?	El presupuesto de 2022 en Jaén incluyó 520.894 euros para mejorar la salud pública y calidad
22	Salamandra 2b instruct	¿Qué partidas están relacionadas con los fondos europeos en 2025?	Los fondos europeos incluyen asignaciones para FEDER, FSE+, FEADER, entre otros, totaliza
23	Salamandra 2b instruct	¿Cuál es el presupuesto total aprobado para la Comunidad Autónoma de Andalucía en 2025?	El presupuesto total aprobado asciende a 44.521 millones de euros, un incremento del 5% res
24	Salamandra 2b instruct	¿Cuales son las condiciones para el nombramiento de personal funcionario interino por exceso en el presupuesto del 202	Las condiciones son: a) Duración máxima de seis meses dentro de un periodo de doce meses
25	Salamandra 2b instruct	¿Qué objetivos busca alcanzar el presupuesto de 2025 en relación con la igualdad de oportunidades?	El presupuesto busca reducir las desigualdades territoriales, mejorar la igualdad de acceso a s
26	Salamandra 2b instruct	¿Cuáles son las medidas fiscales destacadas en el presupuesto 2025?	Se aplican incentivos fiscales por valor de 800 millones para fomentar inversiones en sostenibi
27	Salamandra 2b instruct	¿Qué áreas priorizan las cuentas de la Comunidad Autónoma para 2025?	Las cuentas de la Comunidad Autónoma para 2025 priorizan áreas fundamentales como la ed
28	Salamandra 2b instruct	¿Qué porcentaje del presupuesto total se destina a las universidades andaluzas en el presupuesto 2025?	El 12% del presupuesto total está destinado a las universidades andaluzas, incluyendo investi
29	Salamandra 2b instruct	¿Cual fue el total de costes de personal de las universidades de titularidad pública en 2017?	El total del costes de personal fueron 529.634.378 euros.
30	Salamandra 2b instruct	¿Cuál fue el financiamiento para la infraestructura sanitaria en Albacete en el presupuesto de 2023?	Albacete no pertenece a la comunidad autónoma de Andalucía. No dispongo de información r
31	Salamandra 2b instruct	¿Cuál fue la inversión en políticas de salud en Madrid según el presupuesto de 2022?	Madrid no pertenece a la comunidad autónoma de Andalucía. No dispongo de información rel
32	Salamandra 7b instruct	¿Qué ley se menciona en el documento?	La Ley 12/2023, de 26 de diciembre, del Presupuesto de la Comunidad Autónoma de Andaluc
33	Salamandra 7b instruct	¿Cuál es el objetivo del presupuesto de la Comunidad Autónoma de Andalucía para 2024?	El objetivo es la mejora del bienestar en un marco de estabilidad institucional, diálogo y confia
34	Salamandra 7b instruct	¿Cómo ha variado el gasto en infraestructuras educativas entre 2022 y 2023?	Ha aumentado un 15%, alcanzando 1.260 millones de euros en 2023.
35	Salamandra 7b instruct	¿Cuál fue el financiamiento para la infraestructura sanitaria en Jaén en el presupuesto de 2023?	En 2023, se asignaron 520.894 euros para el sistema de salud en Jaén, mejorando las instala
36	Salamandra 7b instruct	¿Cuál fue la cantidad asignada al Ayuntamiento de Jaén en el presupuesto de 2022?	Al Ayuntamiento de Jaén se le asignaron 75.787 euros dentro del fondo para la financiación de
37	Salamandra 7b instruct	¿Cuál ha sido el presupuesto total aprobado por la Junta de Andalucía en 2023?	El presupuesto total aprobado para 2023 es de 45.619 millones de euros.
38	Salamandra 7b instruct	¿Cuál ha sido la evolución del presupuesto total de la Junta de Andalucía desde 2020 hasta 2024?	El presupuesto ha mostrado un aumento sostenido debido a la implementación de políticas de
39	Salamandra 7b instruct	¿Cual es el total de la dotacion de CANAL SUR RADIO Y TELEVISIÓN, S.A. en el presupuesto del 2025?	El importe de la dotación de Canal Sur Radio y Televisión, SA en el presupuesto del año 2025

Figura 6.35: Exportación de los resultados de la evaluación.

	F	G	H	I	J	K	L	M	N	O	P
1	answer_correctness	answer_relevancy	context_entity_recall	context_precision	context_recall	faithfulness	noise_sensitivity_relevant	semantic_similarity			
2	0.2230110634537510	0.8404690600338077	0.0	0.0	0.0	0.6666666666666666	0.0	0.8920442538150051			
3	0.2150811737123270	0.9569453596721362	0.0	0.0	0.0	0.0	0.0	0.860324694849309			
4	0.1985640985393690	0.0	0.3333333322222220	0.0	0.0	0.3333333333333333	0.0	0.7942563941574795			
5	0.3816994228324570	0.0	0.1666666663888880	0.0	0.0	0.2857142857142857	0.0	0.9268635245332462			
6	0.2341211247283540	0.9673536277373176	0.0	0.0	0.0	0.0	0.0	0.9364844989134168			
7	0.1989056277140450	0.0	0.3333333322222220	0.0	0.0	1.0	0.0	0.7956225108561806			
8	0.2109624175992390	0.9244806125507083	0.0	0.0	0.0	0.0	0.0	0.8438496703969594			
9	0.242763265398256	0.9473699470234076	0.3333333322222220	0.0	0.0	0.0	0.0	0.971053061593024			
10	0.2290974694399230	0.99496228990773	0.2499999996874990	0.0	0.0	0.6666666666666666	0.0	0.9165319818819362			
11	0.4083749282248770	0.0	0.3333333327777770	0.0	0.0	0.75	0.0	0.8834997128995106			
12	0.2004246709726610	0.0	0.0	0.0	0.0	0.0	0.0	0.8016986838906448			
13	0.2200351227421730	0.9178533051632448	0.1249999998437490	0.0	0.0	0.1428571428571428	0.0	0.8801404909686928			
14	0.2313576187288340	0.994558020002013	0.6666666644444440	0.0	0.0	1.0	0.0	0.9254304749153376			
15	0.4533053001730280	0.993366072944763	0.0	0.0	0.0	1.0	0.8888888888888888	0.8901442776151927			
16	0.2138992628893910	0.0	0.0	0.0	0.0	0.8571428571428571	0.0	0.8555970515575659			
17	0.2192815369310890	0.0	0.3333333322222220	0.0	0.0	0.1	0.0	0.877126147724357			
18	0.2167786930704320	0.0	0.0	0.0	0.0	1.0	0.0	0.8671147722817293			
19	0.4916837048617020	0.93287832285463	0.0	0.0	0.0	0.0	0.0	0.9079112900350432			
20	0.5691564295333840	0.9270516503269344	0.0	0.0	0.0	0.5714285714285714	0.5714285714285714	0.9130027554972542			
21	0.2297827677009880	0.953287901580311	0.0	0.0	0.0	0.25	0.0	0.9191310708039546			
22	0.2154055422490330	0.92691975653393	0.0	0.9999999999	0.0	0.5	0.0	0.8616221689961356			
23	0.2227258947911010	0.9853851647208564	0.0	0.0	0.0	0.0	0.0	0.890903579164406			
24	0.6561414809316540	0.994592280351574	0.0	0.9999999999	1.0	0.5	0.25	0.9102802094409024			
25	0.2131512924306570	0.8540475584318216	0.0	0.9999999999	0.0	1.0	0.0	0.8526051697226292			
26	0.2085536937778000	0.8487864035218537	0.0	0.0	0.0	0.5	0.0	0.8342498716070335			
27	0.3101922705099980	0.9404006996249682	0.0909090908264460	0.0	0.0	0.0	0.0	0.9249796083557816			
28	0.2294249156774650	0.9725191543093477	0.0	0.0	0.0	0.0	0.0	0.9176996627098604			
29	0.2286047309796850	0.9982071379319998	0.0	0.9999999999	0.0	0.0	0.0	0.9144189239187412			
30	0.2167080424220340	0.0	0.0	0.0	0.0	0.6666666666666666	0.0	0.8669086849799195			
31	0.5797127671527800	0.0	0.0	0.0	0.0	0.0	0.0	0.8188510686111228			
32	0.2240641392501480	0.8169668485750199	0.0	0.0	0.0	1.0	0.0	0.8963009410830077			
33	0.2238019441666430	0.9505250721817744	0.0	0.0	0.0	0.0	0.0	0.8952077766665733			
34	0.1928894315197580	0.0	0.3333333322222220	0.0	0.0	0.6666666666666666	0.0	0.7715577260790327			
35	0.2179739054295620	0.0	0.1666666663888880	0.0	0.0	1.0	0.0	0.8718956217182516			
36	0.2201434340932150	0.0	0.0	0.0	0.0	1.0	0.0	0.8805737363728616			
37	0.2190317179634780	0.0	0.3333333322222220	0.0	0.0	0.0	0.0	0.8761268718539155			
38	0.2121660514380220	0.0	0.0	0.0	0.0	1.0	0.0	0.8486642057520886			
39	0.2416168955449870	0.9841066679830196	0.3333333322222220	0.0	0.0	1.0	0.0	0.9664675821799518			

Figura 6.36: Exportación de los resultados de la evaluación.

Capítulo 7

RESULTADOS

7.1. Pasos para la Optimización del Sistema de Recuperación

Uno de los elementos fundamentales en la metodología *Retrieval-Augmented Generation* (RAG) es el mecanismo de recuperación de información basado en *embeddings*. Para ello, se han evaluado múltiples formas y modelos de generación de representaciones semánticas con el objetivo de encontrar la combinación óptima para garantizar una recuperación precisa y relevante de información presupuestaria.

Se han considerado y testeado tres enfoques principales:

- **FastEmbed**, utilizando el modelo `BAAI/bge-small-en-v1.5`, el cual permite la generación eficiente de representaciones semánticas optimizadas para tareas de recuperación de información.
- **HuggingFaceEmbeddings**, probando diversos modelos con el fin de analizar su desempeño en la recuperación semántica de información. Entre los modelos evaluados se incluyen:
  - `hiiamsid/sentence_similarity_spanish_es`
  - `PlanTL-GOB-ES/RoBERTalex`
  - `sentence-transformers/all-MiniLM-L6-v2`
  - `sentence-transformers/distiluse-base-multilingual-cased-v2`
  - `emer-software/robertalex-mlm-bcn-mnrl-msmarco-es`

  - `Stern5497/sbert-legal-xlm-roberta-base`
  - Y el propio `BAAI/bge-small-en-v1` aplicado desde HuggingFaceEmbeddings.

- **CustomEmbeddings**: Implementado desde mi parte y aplicando la técnica de Mean pooling (Promedia los embeddings de todos los tokens en la última capa oculta para producir un único vector) a los anteriores modelos.

También los resultados iniciales mostraron que las respuestas generadas a partir de estos modelos no eran óptimas cuando se utilizaba la búsqueda por la **Distancia Euclidiana** (*L2*) como métrica de comparación, lo que indicaba que este método no era adecuado para el contexto de recuperación semántica en RAG ya que depende de la magnitud del embedding, no es invariante al escalado y no mide similitud semántica bien.

Sin embargo se realizó una importante optimización del sistema cambiando a la búsqueda por la distancia a la **Similitud del Coseno**, ya que proporcionó una representación más precisa de la relación semántica entre documentos, es invariante a la magnitud y funciona mejor con embeddings normalizados.

Para maximizar el rendimiento del sistema, también se utilizó la técnica de **mean pooling**, la cual consiste en promediar los embeddings de todos los tokens en una secuencia para obtener un único vector de salida representativo. Esto permite una mejor comparación entre textos de diferente longitud y mejora la recuperación semántica.

Tras un análisis comparativo del rendimiento de estos embeddings a nivel global, y tras una **evaluación empírica sorprendentemente con mejores resultados** (Puede observarlo con más detalle en el análisis de la Sección 7.3.3.1 y apartado *Comparación context precision entre embeddings*), se ha optado por utilizar **HuggingFaceEmbeddings** utilizando el modelo **BAAI/bge-small-en-v1.5** con la implementación de *embeddings* **normalizados**.

7.1.0.1. Resumen

Gracias a la combinación de:

- **Cambio de métrica** de comparación de embeddings (Distancia Euclidiana a Similitud del Coseno).
- **Normalización de los embeddings**.

- **Aplicación de Mean Pooling** como técnica de agregación de vectores.

- **Uso de HuggingFaceEmbeddings**, con sorprendentemente efectividad, el modelo `BAAI/bge-small-en-v1.5`.

Se logró una mejora significativa en la recuperación de documentos relevantes en relación a su contenido(en la que un gran porcentaje de documentos son tablas de valores) dentro del sistema RAG, aumentando la precisión y coherencia de las respuestas generadas por el modelo.

7.2. Pasos para la Optimización de la Generación de Respuestas

Además de mejorar la fase de recuperación, se realizaron optimizaciones en la fase de generación de respuestas, con el objetivo de proporcionar respuestas más precisas y alineadas con la legislación presupuestaria de la Junta de Andalucía.

7.2.0.1. Ajuste de Parámetros del Modelo Generativo

Para obtener respuestas de mayor calidad, se optimizaron diversos parámetros del modelo de lenguaje:

- **Reducción de la temperatura**: Se ajustó la temperatura a valores más bajos para evitar respuestas demasiado creativas y asegurar que fueran más precisas y alineadas con el contenido esperado.

- **Optimización del Prompt**: Se diseñó un **prompt estructurado** que guía al modelo en la generación de respuestas concisas y alineadas con la normativa presupuestaria.

- **Selección del Modelo de Generación**: Se evaluaron diversas familias de modelos (Llama 3, Salamandra, Mistral, DeepSeek) y se seleccionó lo más apropiados.

7.3. Resultados Finales

Estos resultados han sido obtenidos gracias a la utilización de **RAGAS** (Retrieval-Augmented Generation Assessment Suite), que define un conjunto de métricas que permiten evaluar diferentes dimensiones de la calidad del modelo. A continuación, se presentan sus definiciones y fórmulas matemáticas [59]

7.3.1. Métricas Especializadas en RAGAS

Las métricas de RAGAS pueden clasificarse en dos categorías principales: **Generation Metrics (métricas de generación)** y **Retrieval Metrics (métricas de recuperación)**. Veamos qué métricas corresponden a cada tipo:

Generation Metrics (métricas de generación):

- **Faithfulness:** Determina la precisión factual de la respuesta con respecto a los documentos recuperados. Es decir, mide cuál precisa es la respuesta del modelo respecto al contexto recuperado. *Previene alucinaciones y asegura que la información generada esté respaldada por hechos verificables.*

$$\text{Faithfulness} = \frac{\text{Número de Afirmaciones Correctas}}{\text{Total de Afirmaciones en la Respuesta}} \tag{7.1}$$

- **Answer Relevancy:** Evalúa la relevancia de la respuesta con respecto a la consulta del usuario. Es decir, determina si la respuesta generada es relevante para la pregunta planteada. *Es crucial para evitar respuestas correctas pero fuera de contexto.*

$$\text{Answer Relevancy} = \frac{\text{Número de Conceptos Relevantes en la Respuesta}}{\text{Total de Conceptos en la Respuesta}} \tag{7.2}$$

- **Answer Correctness:** Analiza si la respuesta generada es correcta en términos de contenido semántico. Es decir, evalúa si la respuesta generada es completamente correcta en términos de contenido. *Es esencial para aplicaciones críticas como medicina o derecho.*

$$\text{Answer Correctness} = \frac{\text{Número de Respuestas Correctas}}{\text{Total de Respuestas Generadas}} \tag{7.3}$$

- **Answer Similarity:** Mide la similitud entre la respuesta generada y la referen-

cia esperada. *Es útil para verificar si la respuesta está alineada con respuestas esperadas.*

$$\text{Answer Similarity} = \text{Similitud Coseno}(\text{Respuesta Generada}, \text{Referencia}) \quad (7.4)$$

Retrieval Metrics (métricas de recuperación):

- **Context Precision:** Mide la precisión de los documentos recuperados en relación con la consulta. En otras palabras, evalúa la relevancia del contexto recuperado respecto a la consulta. *Es fundamental para evitar información irrelevante que pueda sesgar la generación de respuestas.*

$$\text{Context Precision} = \frac{\text{Número de Sentencias Relevantes Recuperadas}}{\text{Total de Sentencias Recuperadas}} \quad (7.5)$$

- **Context Recall:** Evalúa qué tan bien los documentos recuperados cubren todos los aspectos relevantes de la consulta. Es decir, mide cuánta información relevante ha sido recuperada respecto al total de información existente. *Es crucial para asegurar que el modelo tenga acceso a todos los datos necesarios.*

$$\text{Context Recall} = \frac{\text{Número de Sentencias Relevantes Recuperadas}}{\text{Total de Sentencias Relevantes Disponibles}} \quad (7.6)$$

- **Context Entity Recall:** Evalúa qué tan bien la respuesta preserva las entidades clave del contexto original. Es decir, mide la recuperación de entidades clave desde el contexto recuperado. *Garantiza que la respuesta contenga información específica de relevancia.*

$$\text{Context Entity Recall} = \frac{\text{Número de Entidades Recuperadas Correctamente}}{\text{Total de Entidades Relevantes en el Contexto}} \quad (7.7)$$

- **Noise Sensitivity:** Determina la robustez del modelo ante documentos con ruido en la recuperación de información. *Es importante para asegurar estabilidad ante variaciones menores en las entradas.* En concreto:
  - Permite evaluar la **robustez** del modelo ante errores tipográficos y reformulaciones de preguntas.
  - Modelos con **alta sensibilidad al ruido** pueden producir respuestas inconsistentes con pequeñas modificaciones en la entrada.
  - Es fundamental para garantizar la estabilidad en aplicaciones reales de RAG donde los usuarios pueden ingresar consultas con variaciones naturales.

$$\text{Noise Sensitivity} = 1 - \text{Similitud}(\text{Respuesta Original}, \text{Respuesta con Ruido}) \tag{7.8}$$

Donde:

- Similitud$()$ es una métrica de distancia, típicamente la **similitud coseno** entre los embeddings de las dos respuestas.
- Un valor cercano a 1 indica que la respuesta del modelo cambia significativamente con pequeñas perturbaciones en la consulta.
- Un valor cercano a 0 significa que el modelo es robusto frente a ruido en la entrada.

Ejemplo:

- Se compara la salida del modelo cuando la consulta original tiene palabras clave modificadas.
- Si la variación es alta, la sensibilidad al ruido es elevada (lo cual es negativo).

7.3.1.1. Uso de OpenAI en RAGAS

RAGAS requiere modelos de embeddings y generación de texto para evaluar respuestas generadas, permitiendo contrastar la información proporcionada con una *referencia específica* siendo el *'juez'* en esta implementación los modelos de **OpenAI.** Para ello, RAGAS realiza solicitudes a la API de OpenAI, donde internamente se utilizan los siguientes modelos:

- **GPT-4o-mini-2024-07-18:** Se emplea para evaluar Faithfulness y Answer Relevancy, ya que estos aspectos requieren una comprensión semántica profunda.
- **Text-embedding-ada-002-v2:** Se utiliza para convertir las respuestas en vectores y calcular métricas como Answer Similarity y Context Precision.

7.3.1.2. Conclusión

RAGAS ofrece una solución integral para evaluar modelos RAG, asegurando que las respuestas generadas sean no solo relevantes, sino también objetivamente correctas y basadas en la evidencia proporcionada por los documentos recuperados. Su integración con OpenAI y sus modelos avanzados como **GPT-4o-mini y text-embedding-**

ada-002-v2, garantiza evaluaciones precisas, aunque se podrían explorar alternativas de código abierto para mayor flexibilidad.

7.3.2. Modelos seleccionados

En este apartado se describen los modelos de lenguaje seleccionados para la evaluación, destacando sus características técnicas, evolución y las razones que justifican su elección.

- **Modelos Salamandra 2B Instruct y Salamandra 7B Instruct**: Estos forman parte del **Proyecto ALIA** (*Artificial Language Intelligence for Administration*), una iniciativa del Gobierno español para desarrollar modelos de IA de código abierto que refuercen la soberanía tecnológica y minimicen la dependencia de corpus extranjero [60]. Ambos modelos han sido diseñados para optimizar la eficiencia computacional sin comprometer el rendimiento en generación de texto y procesamiento multilingüe. Sus características destacadas incluyen:
  - **Arquitectura Transformer**: Adaptada para ejecución eficiente en hardware de gama media.
  - **Multilingüismo**: Soporte para 35 idiomas europeos, optimizado para tareas de NLP.
  - **Versiones Instruct**: Mejoradas para generación interactiva basada en instrucciones del usuario.
  - **Soporte para código**: Capacidad de procesamiento y generación en múltiples lenguajes de programación.
  - **Código Abierto**: Disponibles en Hugging Face con licencias abiertas [61] los hacen una alternativa viable especialmente en aplicaciones que requieren privacidad y control sobre los datos.

  Las diferencias entre Salamandra 2B y 7B son que Salamandra 2B Instruct es un modelo compacto para tareas de inferencia rápida en entornos con restricciones de memoria y Salamandra 7B Instruct posee una mayor capacidad de generación y precisión en tareas más complejas.
- **Llama 3.2.3B Instruct y Llama 3.1.8B Instruct**: La familia **Llama 3**, desarrollada por Meta, es la última evolución de la serie de modelos de lenguaje de la compañía. Destaca por sus mejoras en eficiencia computacional y en la calidad de generación de texto. Como características destacadas incluyen:

- **Llama 3.2.3B Instruct**: Versión optimizada con 3.2B de parámetros, enfocada en tareas de bajo consumo computacional, sin sacrificar calidad en generación de instrucciones.
- **Llama 3.1.8B Instruct**: Modelo de mayor capacidad que mejora en tareas de razonamiento, comprensión de contexto y generación semánticamente más precisa. Es una evolución directa de Llama 2, mejorando en eficiencia y precisión.

Los modelos Llama han sido seleccionados por su capacidad de optimización y por la comunidad activa que los respalda, lo que facilita su personalización y mejora en aplicaciones específicas.

- **Mistral 7B Instruct**: Desarrollado por **Mistral AI**, este modelo es una de las referencias en términos de balance entre rendimiento y eficiencia computacional. Su arquitectura incorpora dos mecanismos avanzados:
  - **Grouped-Query Attention (GQA)** [62]: Acelera la inferencia y reduce la carga de memoria.
  - **Sliding Window Attention (SWA)** [63]: Optimiza la capacidad del modelo para manejar secuencias largas con menor coste computacional.

  Mistral 7B Instruct se ha entrenado con datos de instrucciones públicamente disponibles, lo que permite su adaptación a múltiples tareas sin necesidad de reentrenamiento costoso. Ha sido seleccionado debido a su rendimiento superior en tareas de generación y razonamiento, superando a modelos de mayor tamaño en pruebas comparativas.

- **DeepSeek R1 Distill Llama 8B y DeepSeek R1 Distill Qwen 7B**: En los últimos meses, los modelos **DeepSeek R1 Distill** han ganado una gran notoriedad dentro de la comunidad de inteligencia artificial debido a su combinación de rendimiento eficiente y capacidades avanzadas de generación de lenguaje. Estos modelos representan una evolución en la tendencia de modelos distilados [64] [65], que buscan mantener una alta calidad en la generación de texto con un menor costo computacional.

  Características Técnicas:

  - **DeepSeek R1 Distill Llama 8B** [66]: Un modelo basado en la arquitectura *LLaMA*, optimizado para tareas de generación de texto de alta calidad con un menor costo de inferencia. Diseñado para escenarios donde la comprensión semántica avanzada es clave.

- **DeepSeek R1 Distill Qwen 7B** [67]: Variante basada en la arquitectura *Qwen*, que destaca en generación estructurada de respuestas, razonamiento matemático y codificación, con una optimización específica en tareas de IA conversacional.

Razón de su uso y evaluación en este sistema:

- **Balance entre eficiencia y calidad**: Comparar su rendimiento en relación con modelos más grandes como Mistral 7B o LLaMA 3, observando si mantienen respuestas coherentes con menor costo de computación.
- **Capacidad de recuperación de información**: Evaluar cómo aprovechan los documentos recuperados para generar respuestas factuales y contextualmente relevantes.
- **Robustez en tareas de generación**: Medir su desempeño en generación estructurada, respuestas a preguntas y razonamiento en entornos conversacionales.
- **Investigar sobre estos nuevos modelos**: Comprobar de alguna manera lo que están diciendo de estos modelos, como que ha optimizado sus modelos con un esquema de **Refuerzo con Retroalimentación Humana (RLHF)** [68] más refinado, reduciendo errores en la generación y mejorando la coherencia factual basado en un entrenamiento en datos de alta calidad y dominio específico y por último, para comprobar como afecta a la respuestas del sistema las optimizaciones que poseen en su arquitectura Transformer, como **FlashAttention [69] y Grouped Query Attention (GQA)** [62], que permiten manejar secuencias largas con menor latencia.

En definitiva, estos modelos han sido seleccionados debido a su capacidad de mantener alta precisión con un menor consumo de recursos, permitiendo una escalabilidad óptima en entornos como el de este sistema con restricciones computacionales.

7.3.3. Análisis

A continuación se detalla los resultados de la evaluación de los 7 modelos escogidos para tal fin con unas 30 preguntas consideradas como source of truth (Ver apéndice 10.2). Para ello se va a proceder a comentar la interpretación y análisis de los gráficos. Finalmente en la sección **Apéndice B** aparece todos los datos de los tres mejores modelos (Ver apéndice 10.1):

7.3.3.1. Valores individuales

Gráficos de Distribución En términos generales, aparece la mayoría de los modelos presentan un pico de densidad en torno a 0.2-0.3 un poco bajos lo que sugiere que la muchas de las respuestas generadas no son completamente correctas al cien por cien.

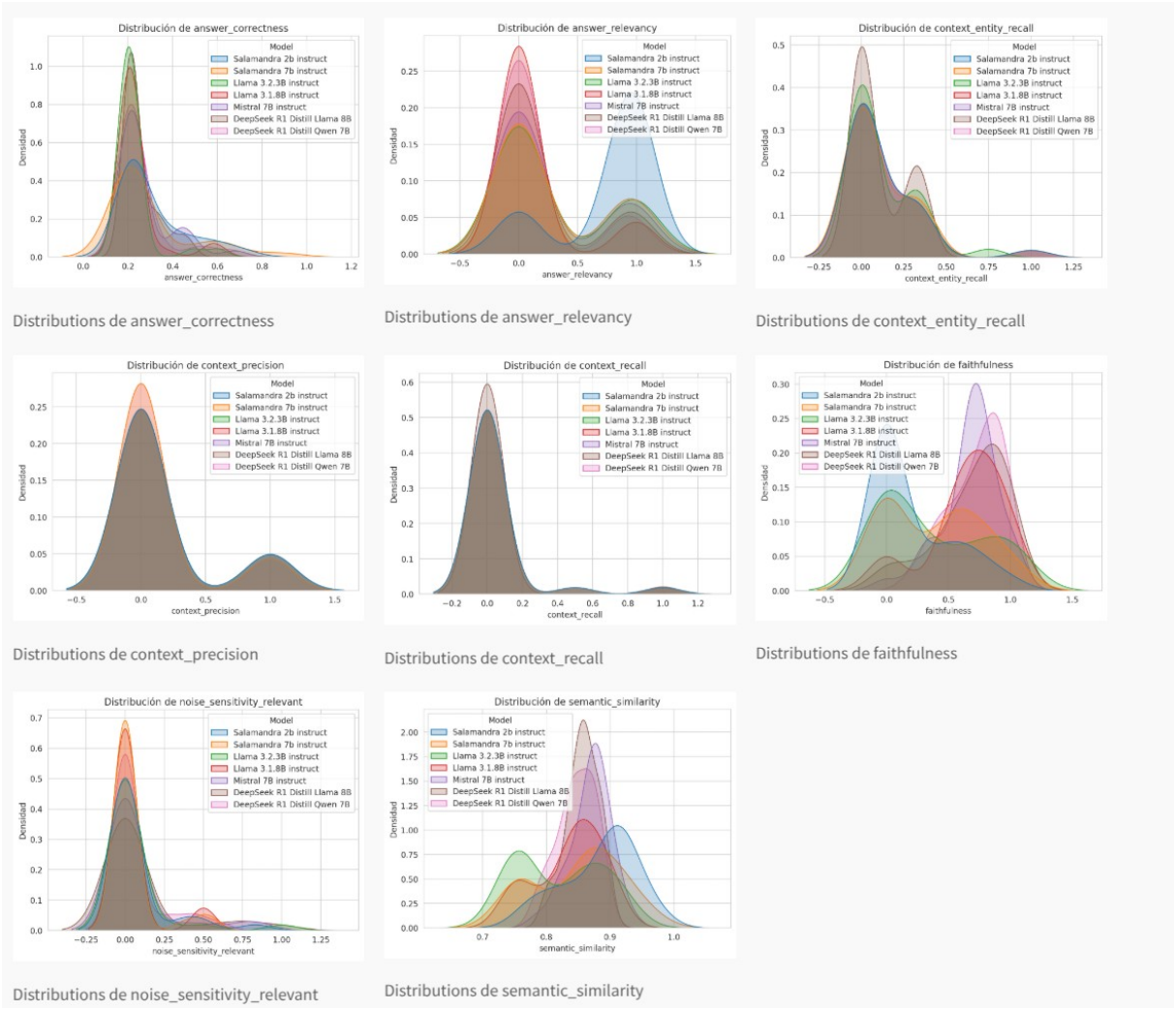


Figura 7.1: Gráficos de distribución - Valores individuales.

Diferencias entre Modelos:

- **Llama 3.2.3B Instruct** (verde) y **Llama 3.1.8B Instruct** (rojo) muestran los picos de densidad más pronunciados en valores más altos, lo que sugiere que estos modelos ofrecen respuestas con una mayor corrección en comparación con el resto.
- **Salamandra 7B Instruct** (naranja) presenta una mayor dispersión, con una menor concentración en valores bajos y una mayor probabilidad de generar res-

puestas correctas. Tiene una densidad no despreciable en valores superiores a 0.5, lo que sugiere que en algunos casos es capaz de generar respuestas con un alto grado de corrección.

- **Mistral 7B Instruct** (morado) y **DeepSeek R1 Distill Llama 8B** (marrón) tienen distribuciones más amplias, lo que implica que pueden tener una mayor variabilidad en la corrección de sus respuestas.

- **DeepSeek R1 Distill Qwen 7B** (rosado) muestra una distribución similar a la de otros modelos, pero con un rango más amplio, lo que indica que ocasionalmente puede generar respuestas más correctas.

Primer Acercamiento con Gráficos de Distribución:

- **Llama 3.2.3B** y **Llama 3.1.8B** parecen ser los modelos con mejor desempeño en términos de corrección de respuestas, seguidos de **Salamandra 7B**.

- Los modelos **DeepSeek** y **Mistral** muestran una mayor dispersión, lo que indica respuestas más inconsistentes en términos de corrección.

- Si la corrección de respuesta es prioritaria en el sistema RAG, **Llama 3.2.3B** y **Salamandra 7B** podrían ser opciones más adecuadas para la tarea.

Comparación context precision entre embeddings: En concreto, si nos fijamos en la métrica de **context_precision** utilizando `BAAI/bge-small-en-v1.5` y comparando con esta evaluación en la que se utilizó el embedding `PlanTL-GOB-ES/RoBERTalex` se observa que:

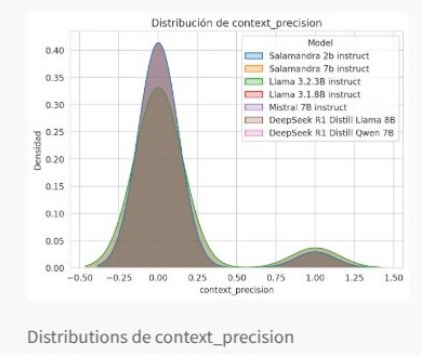


Figura 7.2: Context_precision para PlanTL-GOB-ES/RoBERTalex - Valores indiv.

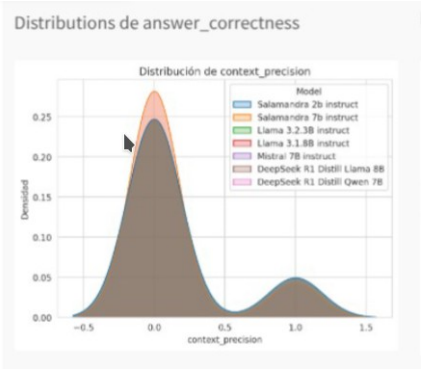


Figura 7.3: Context_precision BAAI/bge-small-en-v1.5 - Valores indiv.

- En la gráfica anterior, RoBERTalex, la densidad se concentra mayormente alrededor de context_precision cercano a 0.0, con una distribución simétrica y una menor separación entre modelos.
- En la gráfica del análisis (BAAI/bge-small-en-v1.5), la densidad es más concentrada en valores más altos, lo que indica una mejor correlación entre contexto y precisión de respuesta.

Concluyendo de forma empíricamente demostrado que **BAAI/bge-small-en-v1.5** nos proporciona **mejores resultados.**

7.3.3.2. Gráficos de Dispersión

Permite identificar patrones de comportamiento entre las métricas:

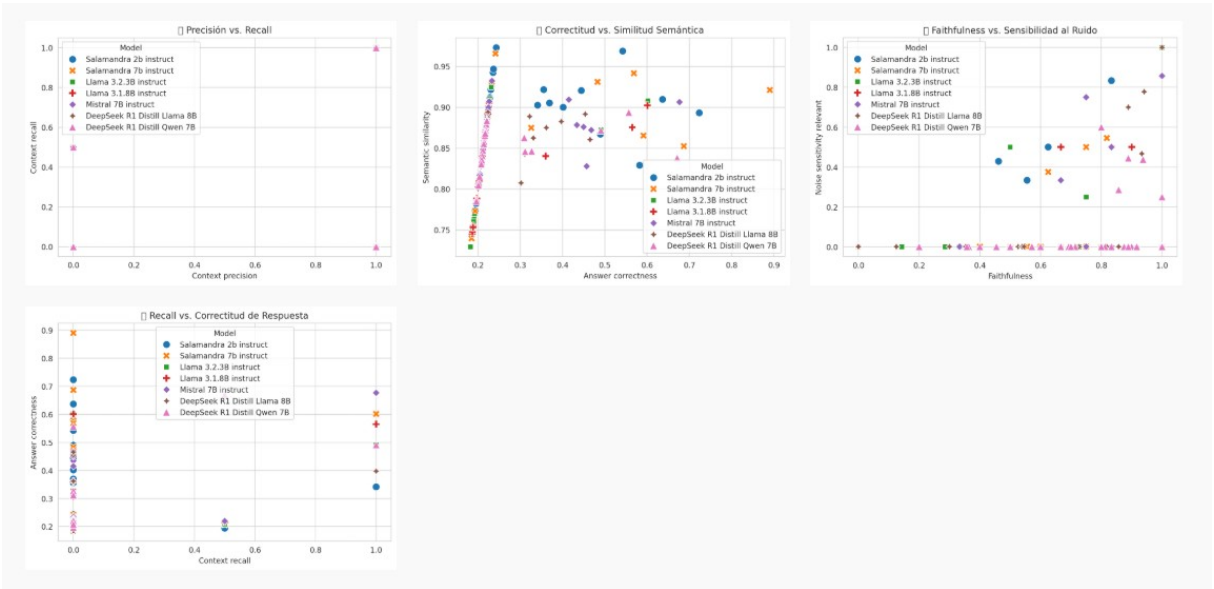


Figura 7.4: Gráficos de dispersión - Valores individuales.

Precisión vs. Recall (Gráfico Superior Izquierdo)

- La mayoría de los modelos tienen valores de precisión muy bajos y valores de recall relativamente más altos.
- La mayoría de los modelos recuperan información relevante (buen recall), pero no necesariamente con precisión alta.
- **Salamandra 7B** y **Llama 3.2.3B** (naranja y verde, respectivamente) tienen mejor equilibrio entre precisión y recall en comparación con otros.

Correctitud de Respuesta vs. Similitud Semántica (Gráfico Superior Centro)

- Se observa una clara tendencia positiva, lo que indica que cuando un modelo responde correctamente, su similitud semántica con la respuesta esperada es mayor.
- Los modelos con mayor correctitud tienden a generar respuestas más similares semánticamente, pero algunos modelos tienen problemas de estabilidad en la calidad de la respuesta.

Fidelidad vs. Sensibilidad al Ruido (Gráfico Superior Derecho)

- Se observa que los modelos con alta fidelidad (*faithfulness*) pueden presentar alta o baja sensibilidad al ruido.
- Modelos con menor sensibilidad al ruido ofrecen respuestas más confiables y menos afectadas por datos irrelevantes.
- **Salamandra 7B** y **Llama 3.2.3B** parecen ser las mejores opciones en este aspecto.

Recall vs. Correctitud de Respuesta (Gráfico Inferior Izquierdo)

- Se observa que un mayor recall no necesariamente implica una mayor correctitud de respuesta.
- Un alto recall no garantiza que la respuesta sea correcta, y los modelos mejor equilibrados son **Salamandra 7B** y **Llama 3.2.3B**.

Primer Acercamiento con Gráficos de Dispersión:

- **Salamandra 7B** y **Llama 3.2.3B** son los modelos con mejor equilibrio entre correctitud, similitud semántica, fidelidad y baja sensibilidad al ruido.
- **DeepSeek R1 Distill Qwen 7B** muestra valores extremos en recall y sensibilidad al ruido, lo que sugiere que es menos estable en sus respuestas.
- Los modelos **DeepSeek** tienden a recuperar mucha información (alto recall), pero con menor precisión y correctitud.
- Si el objetivo del sistema RAG es maximizar precisión y fidelidad sin ser afectado por ruido, los modelos recomendados serían **Salamandra 7B** y **Llama 3.2.3B**.

7.3.4. Pruebas de Robustez ante Preguntas Fuera de Contexto

En el conjunto de preguntas diseñadas, se han incluido dos específicamente orientadas a evaluar la robustez de los modelos y su capacidad para reconocer situaciones fuera de contexto en el ámbito de Andalucía. Estas preguntas permiten analizar hasta qué punto los modelos pueden identificar y gestionar posibles alucinaciones en sus respuestas.

Pregunta	**Modelo**	**Respuesta**
¿Cuál fue el financiamiento para la infraestructura sanitaria en Albacete en el presupuesto de 2023?	Salamandra 2b instruct	La cantidad exacta no está disponible debido a restricciones de acceso al contexto. Sin embargo, en España existen fondos como el FEDER y el MRR destinados a mejorar la infraestructura sanitaria. Estos fondos pueden variar según la disponibilidad de recursos y factores específicos.
	Salamandra 7b instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
	Llama 3.2.3B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
	Llama 3.1.8B instruct	No se encontró información suficiente en el contexto proporcionado. Se menciona la Red de Albergues con ahorro en el erario público, pero no se relaciona con la infraestructura sanitaria en Albacete.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
	DeepSeek R1 Distill Llama 8B	Se menciona el PRTR con medidas para rehabilitación urbana y sostenibilidad, pero no se especifican cifras para Albacete ni para la infraestructura sanitaria en particular.
	DeepSeek R1 Distill Qwen 7B	Se revisó el contexto y no se encontraron menciones específicas sobre financiamiento para la infraestructura sanitaria en Albacete en el presupuesto de 2023.
¿Cuál fue la inversión en políticas de salud en Madrid según el presupuesto de 2022?	Salamandra 2b instruct	Según el presupuesto de 2022, la inversión en políticas de salud en Madrid asciende a 1.100 millones de euros.
	Salamandra 7b instruct	En el contexto provisto, no se encuentra información sobre la inversión específica en políticas de salud durante el presupuesto de 2022 en Madrid. "No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza."

Pregunta	Modelo	Respuesta
	Llama 3.2.3B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto proporcionado es el PROYECTO DEL PRESUPUESTO DE LA COMUNIDAD AUTÓNOMA DE ANDALUCÍA 2023, que no menciona la inversión en Madrid.
	Llama 3.1.8B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza, ya que el texto se refiere al presupuesto de Andalucía y no al de Madrid. Sin embargo, la dotación para la política de Sanidad en Andalucía para 2023 asciende a 13.688,1 millones de euros, con un incremento del 10,9 %.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto solo proporciona información sobre Andalucía y no sobre Madrid.
	DeepSeek R1 Distill Llama 8B	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. Se menciona la política de Sanidad en Andalucía, pero no hay datos para Madrid.
	DeepSeek R1 Distill Qwen 7B	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El presupuesto de 2023 menciona Andalucía, pero no hay referencias a 2022 en Madrid.

Tabla. 7.1: Tabla de Preguntas fuera de Contexto (Análisis de alucinaciones)

Los resultados obtenidos en las pruebas realizadas muestran que algunos modelos presentan una mayor capacidad para detectar la falta de información en el contexto proporcionado y evitar la generación de respuestas incorrectas o alucinaciones. En particular, los modelos que han demostrado una correcta identificación de situaciones fuera de contexto son los siguientes:

- **Salamandra 7B**
- **Llama 3.2.3B**
- **Llama 3.1.8B**
- **Mistral 7B**
- **DeepSeek R1 Distill Llama 8B**
- **DeepSeek R1 Distill Qwen 7B**

Estos modelos han demostrado una sólida capacidad de detección del contexto, evitando proporcionar información no sustentada cuando el contenido del corpus no contiene datos relevantes para la pregunta planteada. Por otro lado, se ha observado que el modelo **Salamandra 2B instruct** muestra una cierta tendencia a generar respuestas sin estar claramente respaldadas por la información disponible en el contexto. Esto sugiere que aún hay margen de mejora en la capacidad de estos modelos para reconocer sus propias limitaciones y prevenir alucinaciones en sus respuestas. A partir de

este análisis, se puede intuir que modelos de mayor tamaño, como **DeepSeek, Llama y Mistral**, presentan un mejor desempeño en la tarea de evitar alucinaciones, mientras que modelos más pequeños, como **Salamandra 2B**, parecen ser más propensos a introducir información no verificada. Esto resalta la importancia de continuar optimizando los modelos en términos de control del contexto y precisión en la generación de respuestas en escenarios de información limitada.

7.4. Conclusión de los Resultados

A partir de los análisis presentados, se pueden extraer las siguientes conclusiones y recomendaciones sobre los modelos evaluados:

- **Mejores Modelos en General**:
  - **Llama 3.2.3B Instruct** y **Salamandra 7B Instruct** son los modelos que destacan en la mayoría de las métricas evaluadas.
  - Ambos modelos muestran picos de densidad más pronunciados en valores altos, lo que indica una mayor probabilidad de generar respuestas correctas.
  - Tienen un mejor equilibrio entre recuperar información relevante (recall) y asegurar que esa información sea precisa (precisión).
  - Ambos modelos son capaces de identificar cuándo una pregunta no tiene información relevante en el contexto, evitando alucinaciones.
- **Modelos con Desempeño Intermedio**:
  - **Llama 3.1.8B Instruct**: Aunque tiene un buen desempeño en correctitud y detección de contexto, es ligeramente inferior a Llama 3.2.3B y Salamandra 7B en términos de equilibrio entre precisión y recall.
  - **Mistral 7B Instruct**: Muestra una mayor variabilidad en la corrección de respuestas y una distribución más amplia, lo que sugiere menos consistencia en comparación con los modelos líderes.
  - **DeepSeek R1 Distill Llama 8B** y **DeepSeek R1 Distill Qwen 7B**: Estos modelos tienen un alto recall, pero su precisión y correctitud son menores. Además, muestran una mayor sensibilidad al ruido, lo que los hace menos confiables en escenarios donde la fidelidad es crítica.
- **Modelos con Desempeño Inferior**:

  - **Salamandra 2B Instruct**: Este modelo tiende a generar respuestas no respaldadas por el contexto (**alucinaciones**) y tiene un desempeño significativamente inferior en comparación con los modelos más grandes. No es recomendable para tareas que requieran alta precisión y fidelidad.

- **Recomendaciones Finales**:

  - Si el objetivo es maximizar la correctitud de las respuestas y minimizar la sensibilidad al ruido, los modelos recomendados son **Llama 3.2.3B Instruct** y **Salamandra 7B Instruct**.
  - Si el objetivo es recuperar la mayor cantidad de información relevante (aunque con menor precisión), los modelos **DeepSeek R1 Distill Llama 8B** y **DeepSeek R1 Distill Qwen 7B** podrían ser considerados, aunque con reservas debido a su inconsistencia.
  - Para tareas donde es crítico evitar respuestas no respaldadas por el contexto, los modelos **Llama 3.2.3B**, **Salamandra 7B** y **Mistral 7B** son las mejores opciones.

En resumen, **Llama 3.2.3B Instruct** y **Salamandra 7B Instruct** son los modelos más recomendados para tareas que requieran alta correctitud, fidelidad y equilibrio entre precisión y recall.

Capítulo 8

CONCLUSIONES

8.1. Análisis sobre Consultas de Datos en Tablas

El presente trabajo ha permitido profundizar en el desarrollo de un sistema basado en **Generación Aumentada por Recuperación (RAG)** aplicando *Grandes Modelos de Lenguaje* (LLMs) para la extracción, análisis y generación de información a partir de las leyes del presupuesto de Andalucía. Durante el proceso, se han adquirido conocimientos fundamentales sobre **indexación semántica, embeddings, bases de datos vectoriales, procesamiento de lenguaje natural (PLN)** y **evaluación de modelos de IA.**

Uno de los aspectos clave ha sido comprender cómo estructurar grandes volúmenes de información para facilitar su búsqueda y recuperación eficiente. Para este tipo de información, queda de manifiesto la **dificultades en el procesamiento de la información tabular (en forma de tablas)** en la que la información no es tan directa de deducir y extraer. Es por ello que en el análisis nos muestra unos resultados y rendimiento bajos y hasta a veces aparecen alucinaciones, lo que implica que las métricas de recuperación son muy bajas y a veces sin contexto de calidad. Por ejemplo, podemos verlo para esta pregunta (v. tabla 8.1):

Mientras tanto, para una información no tabular sí aparece información del retriever (v. tabla 8.2):

Pregunta	Retriever Context
¿Cuál es el total de la dotación de CANAL SUR RADIO Y TELEVISIÓN, S.A. en el presupuesto del 2025?	PRESUPUESTO DE LA COMUNIDAD AUTÓNOMA DE ANDALUCÍA 2025 5 CANAL SUR RADIO Y TELEVISIÓN, S.A. PRESUPUESTO DE LA COMUNIDAD AUTÓNOMA DE ANDALUCÍA 2025 114 CANAL SUR RADIO Y TELEVISIÓN, S.A. PRESUPUESTO DE LA COMUNIDAD AUTÓNOMA DE ANDALUCÍA 2025 11 CANAL SUR RADIO Y TELEVISIÓN, S.A.

Tabla. 8.1: Contexto recuperado para la pregunta sobre Canal Sur en el presupuesto 2025

8.2. Aportaciones y Logros Alcanzados

Este trabajo ha conseguido diseñar e implementar un sistema funcional y altamente eficiente, cumpliendo con los objetivos iniciales planteados. Entre las principales aportaciones se destacan:

- **Diseño e implementación de una arquitectura escalable:** Se ha desarrollado un sistema modular con un backend eficiente basado en Flask y un frontend interactivo en Streamlit, permitiendo la consulta y evaluación de modelos sin afectar el rendimiento de la interfaz.
- **Optimización del Sistema con Arquitectura Modular:** El desarrollo de la aplicación siguiendo principios de **buenas prácticas** y utilizando **patrones de diseño** ha permitido construir un sistema altamente modular y flexible. Gracias a esta arquitectura bien definida, se puede modificar y sustituir componentes clave como:
  - **Embeddings**: Se pueden intercambiar distintos modelos de embeddings según la necesidad del sistema sin afectar el resto de la implementación.
  - **Base de datos vectorial**: La aplicación permite cambiar entre diferentes soluciones de almacenamiento vectorial con un impacto mínimo en la estructura del código.
  - **Modelos de lenguaje**: Es posible integrar nuevos modelos sin necesidad de realizar modificaciones significativas, lo que facilita la actualización y mejora del sistema.

Pregunta	Retriever Context
¿Cuales son las condiciones para el nombramiento de personal funcionario interino por exceso en el presupuesto del 2020?	['de Presupuestos.ículo 22. Nombramiento de personal funcionario interino por razones expresamentede necesidad y urgencia.1. De conformidad con lo dispuesto en el artículo 10.1 d de la Ley 7/2007, de 12 de, del Estatuto Básico del Empleado Público, podrá efectuarse el nombramiento defuncionario interino por causas expresamente justificadas de necesidad y ur-originadas por exceso o acumulación de tareas, con las siguientes condiciones:) La duración del nombrami ento no podrá ser superior a seis meses dentro deperiodo de doce meses.) Los nombramientos, que se efectuarán con cargo al capítulo I del Presu-, requerirán autorización de la Consejería de Hacienda y Administración Pública.) El personal funcionario interino nom brado por este motivo no ocupará plazasla relación de puestos de trabajo. Legislativo 5/2015, de 30 de octubre, podrá efectuarse el nombramiento de personal interino por exceso o acumulación de tareas con las siguientes condiciones:) La duración del nombramiento no podrá ser superior a seis meses dentro de un íodo de doce meses.) Los nombramientos, que se efectuarán con cargo al Capítulo I del Presupuesto, án autorización de la Consejería competente en materia de Administración ública, que será emitida en el plazo de un mes a contar desde la recepción del expediente , previo informe favorable de la Consejería competente en materia de Hacienda.) El personal funcionario interino nombrado por este motivo no ocupará plazas de la ón de puestos de trabajo.2. Asimismo, de acuerdo con lo dispuesto en el artículo 10, apartado 1, párrafo c), texto refundido de la Ley del Estatuto Básico del Empleado Público, podrá efectuarse nombramiento de personal funcionario interino para la ejecución de programas de PRESUPUESTO DE LA COMUNIDAD AUTÓNOMA DE ANDALUCÍA PARA EL AÑO 2010 2. Asimismo, de acuerdo con lo dispuesto en el artículo 10.1 c de la Ley 7/2007, podrá efectuarse el nombramiento de personal funcionario interino para la ón de acciones de carácter temporal cuya financiación provenga de fondos de Unión Europea, con las siguientes condiciones:) La duración del nombramiento no podrá exceder la de la ejecución de la ón a la que se adscriba y, en todo caso, no superará el plazo de dos años.) Los nombramientos, que se efectuarán con cargo a la aplicación que financie el programa afectado, requerirán autorización de la ía de Justicia y Administración Pública, previo informe favorable de la ía de Economía y Hacienda.) El personal funcionario interino nombrado por este motivo no ocupará de la relación de puestos de trabajo. 3. Las retribuciones básicas y las complementarias asignadas al personal al']

Tabla. 8.2: Contexto recuperado para la pregunta sobre interinos en el presupuesto 2020

- **Nuevos formatos de importación y lectura de tipos de ficheros**: De la misma manera, es posible integrar nuevos formatos de ficheros sin necesidad de realizar modificaciones significativas.

Esta capacidad de adaptación y escalabilidad no solo optimiza el mantenimiento y la evolución de la aplicación, sino que también permite experimentar con diferentes configuraciones para mejorar el rendimiento sin comprometer la estabilidad del sistema. En definitiva, el uso de un diseño sólido ha convertido a

la aplicación en una plataforma versátil y fácilmente configurable con el mínimo esfuerzo.

- **Integración de un motor de recuperación semántica:** Se ha aplicado ChromaDB como base de datos vectorial para indexar y recuperar documentos de manera eficiente, permitiendo la búsqueda semántica en grandes volúmenes de información.
- **Evaluación comparativa de modelos de lenguaje:** Se ha implementado una metodología que permite medir la calidad de los modelos LLM en base a múltiples métricas ponderadas, asegurando que la selección del modelo más adecuado se realice en función del rendimiento real.
- **Optimización del procesamiento en demanda:** Se ha logrado que el sistema ejecute evaluaciones masivas sin bloquear la interfaz, permitiendo que el usuario continúe utilizando el chat y otras funcionalidades en paralelo.
- **Automatización del flujo de trabajo:** Se ha diseñado un sistema que permite la ingestión de nuevos datos mediante archivos PDF (dando la posibilidad de cualquier formato por ejemplo .csv), automatizando la actualización de la base de datos vectorial y asegurando que la información utilizada para las consultas esté siempre actualizada.
- **Flexibilidad y parametrización del sistema:** Se ha diseñado un archivo de configuración (`config.yaml`) que permite modificar aspectos clave del sistema, como el tipo de base de datos, el modelo de embeddings, el peso de las métricas en la evaluación y la apariencia visual del frontend.
- **Validación empírica del paradigma RAG:** Se ha demostrado que la combinación de técnicas de recuperación y generación de texto mejora la accesibilidad y precisión de la información, reduciendo la ambigüedad en la interpretación de datos presupuestarios.

Estos logros no solo evidencian la viabilidad técnica del sistema, sino que también demuestran su aplicabilidad en contextos reales donde la recuperación eficiente de información es un desafío clave.

8.3. Líneas de Mejora y Expansión

Si bien el sistema desarrollado ha alcanzado los objetivos iniciales, existen múltiples oportunidades para ampliar y mejorar su funcionalidad. A continuación, se pre-

sentan algunas de las principales líneas de mejora:

8.3.1. Recuperación de información tabular

Hay que trabajar fuertemente en el futuro y con gran énfasis, y de hecho para mí es considerado un **reto**, la recuperación tabular de la información para mejorar resultados finales del modelo.

También se estudiará el uso de técnicas de recuperación de información tradicional o no semántica con el fin de recuperar información en tabla, o menciones específicas a determinados términos o cifras.

8.3.2. Optimización y Ajuste de Modelos

- **Fine-tuning del modelo de embeddings:** Actualmente, se emplea un modelo preentrenado para generar las representaciones vectoriales del texto. Sin embargo, ajustar este modelo con datos específicos del dominio de leyes del presupuestario permitiría mejorar significativamente la relevancia y precisión de la recuperación de información.

- **Fine-tuning del modelo de lenguaje:** Personalizar los modelos LLM con datos específicos del corpus presupuestario mejoraría la coherencia y exactitud de las respuestas generadas, optimizando la adaptabilidad del sistema a un dominio especializado.

8.3.3. Mejoras técnicas

- **Autenticación y Autorización en la API Flask:** Implementar autenticación basada en JWT (JSON Web Token) en la API de Flask garantizará que solo los usuarios autorizados puedan acceder a los endpoints..

- **Automatización con Web Scraping:** Una mejora para la aplicación sería automatizar la extracción de documentos (leyes presupuestarias) desde sitios web oficiales mediante WebScraping y crear un cron para que automáticamente se ejecute el comando de importación de leyes. Esto supondría una **automatización total del proceso de importación de leyes.**

- **Incorporación en un sistema de Integración contínua:** Una En relación con la anterior mejora, sería integrar Docker en un pipeline de Integración Continua (CI). Esto permitiría automatizar la construcción, prueba y despliegue del sistema, garantizando estabilidad y evitando problemas de configuración en diferentes entornos.

8.3.4. Expansión del Ámbito del Sistema

- **Extensión a otras comunidades autónomas:** Actualmente, el sistema se ha aplicado al análisis de presupuestos de Andalucía, pero podría escalarse para cubrir el resto de comunidades autónomas de España, permitiendo un análisis comparativo más completo.
- **Implementación de un sistema multilingüe:** Integrar soporte para múltiples idiomas permitiría ampliar el alcance del sistema, haciéndolo útil para investigadores y administraciones públicas en diferentes países.
- **Automatización de la ingestión de datos:** Implementar un módulo de scraping que extraiga automáticamente la información presupuestaria desde fuentes oficiales y la incorpore a la base de datos vectorial, asegurando que el sistema esté siempre actualizado sin intervención manual.
- **Expansión a otros dominios:** Aunque el enfoque inicial ha sido la recuperación de información presupuestaria, el mismo sistema podría adaptarse para trabajar con legislación, normativas o documentos jurídicos, optimizando el acceso a información compleja en múltiples ámbitos.

8.4. Conclusión Final

Este trabajo representa un avance significativo en la integración de **inteligencia artificial y procesamiento del lenguaje natural** en el acceso y gestión de información estructurada. Se ha demostrado que la combinación de **Grandes Modelos de Lenguaje y técnicas de recuperación semántica** es una estrategia altamente efectiva para mejorar la accesibilidad y precisión de la información, facilitando la toma de decisiones basada en datos.

Por otro lado, indicar que la aplicación de técnicas de **vectorización de texto, ajuste de hiperparámetros en bases de datos vectoriales (ChromaDB) y optimi-**

zación del almacenamiento para asegurar tiempos de respuesta rápidos y resultados lo más precisos posibles. Además, se han explorado diferentes arquitecturas de LLMs y su rendimiento en tareas de generación de texto, comparando modelos en función de su eficiencia, coste computacional y relevancia en contextos específicos.

A nivel de **ingeniería de software**, este proyecto ha sido un ejercicio avanzado de construcción de una **infraestructura tecnológica robusta y adaptable**. Para ello, se ha construido una arquitectura basada en **patrones de diseño y buenas prácticas**, asegurando que la solución pueda ser mantenida y ampliada fácilmente.

Además, la arquitectura modular y escalable diseñada permite que el sistema pueda evolucionar y adaptarse a nuevos desafíos. Gracias a su flexibilidad, el sistema puede ser mejorado con nuevas técnicas de aprendizaje automático, optimización de modelos y automatización del flujo de datos, garantizando su utilidad a largo plazo.

No obstante, el campo de la recuperación aumentada sigue en constante evolución, y este trabajo sienta las bases para futuras investigaciones y desarrollos en el ámbito de la IA aplicada a la gestión de información. La mejora continua de los modelos de lenguaje, junto con la optimización de las bases de datos vectoriales y el ajuste fino de los embeddings, permitirá que este tipo de sistemas sigan mejorando en términos de precisión, eficiencia y aplicabilidad.

En definitiva, este proyecto demuestra que las tecnologías de IA pueden transformar radicalmente la forma en que accedemos y analizamos información digital, abriendo nuevas posibilidades para la automatización y optimización de procesos en múltiples sectores.

Capítulo 9

APÉNDICE A

9.1. Introducción

Este documento proporciona instrucciones detalladas para la instalación, configuración y uso del sistema de Recuperación y Generación Aumentada (RAG) aplicado al análisis presupuestario. Se cubren los aspectos esenciales de configuración, ejecución y personalización del sistema.

9.2. Requisitos del Sistema

Para garantizar el correcto funcionamiento del sistema, se requieren los siguientes requisitos mínimos:

- **Sistema Operativo:** Linux (Ubuntu 20.04+) / Windows 10+ (WSL2 recomendado).
- **Hardware:**
  - Procesador multinúcleo (Intel i7 o AMD Ryzen 7 equivalente).
  - GPU NVIDIA con al menos 8GB de VRAM (recomendado 16GB para optimización de modelos de lenguaje).
  - Almacenamiento mínimo: 150GB disponibles.
  - Memoria RAM: 16GB mínimo (32GB recomendados).
- Software necesario:

- Python 3.12.3
- Conda (Miniconda recomendado) o Virtualenv
- Visual Code https://code.visualstudio.com/
- **Extensiones recomendadas:**
  - Python
  - Pylance
  - Jupyter
  - Python Debugger
  - Python Indent
  - Remote - SSH

9.3. Instalación

9.3.1. Paso 1: Instalación del Entorno (Conda)

Ejecute los siguientes comandos en su terminal dependiendo de su sistema operativo.

9.3.1.1. Linux

```
# Instalar Miniconda
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
source ~/.bashrc

# Crear un entorno virtual con Conda
conda create -n rag-ja python=3.12.3
conda activate rag-ja
# O en su defecto
source activate rag-ja

# Código fuente
Copiar el codigo fuente en donde se desee, terminado en la carpeta (~/rag-ja)
cd rag-ja

# Instalar dependencias
pip install -r requirements.txt
```

9.3.1.2. Windows (WSL2 recomendado)

```
# Instalar Miniconda y crear entorno virtual
download Miniconda installer from https://repo.anaconda.com/miniconda/
conda create -n rag-ja python=3.12.3
conda activate rag-ja
# O en su defecto
source activate rag-ja

# Código fuente
Copiar el codigo fuente en donde se desee, terminado en la carpeta (~/rag-ja)
cd rag-ja

# Instalar dependencias
pip install -r requirements.txt
```

9.3.2. Paso 2: Ejecución del Sistema

9.3.2.1. Iniciar el Backend

Ejecute el servidor API REST con el siguiente comando:

```
python rag_api.py
```

El backend estará disponible en ‘http://localhost:5222‘ por defecto.

9.3.2.2. Iniciar el Frontend

Ejecute la interfaz gráfica de usuario (cliente):

```
streamlit run pln_andalucia.py
```

Acceda a la interfaz web en ‘http://localhost:8501‘.

9.3.2.3. Iniciar Importación Base del Conocimiento

Ejecute la interfaz consola de administrador (cliente/consola):

```
python cli.py datasource_files_import <ruta donde está base datos
   datos conocimiento>
```

9.4. Configuración claves (API_KEY)

9.4.1. Clave API de OpenAI

Para obtener una clave API de OpenAI, sigue los siguientes pasos:

1. Acceder a **OpenAI** a través del enlace, si desea crear una cuenta (**aquí**) o acceder a una ya existente(**aquí**)
2. Ir a la sección **API Keys**: **aquí**.
3. Hacer clic en **Create a new secret key**.
4. Copiar y almacenar la clave en un lugar seguro, ya que solo se mostrará una vez.
5. Configurar posteriormente en la sección Configuración del sistema.

9.4.2. Clave API de Hugging Face y Configuración de Permisos

Para obtener una clave API en Hugging Face y habilitar permisos de lectura para modelos y embeddings:

1. Acceder a **Hugging Face** (**aquí**).
2. Iniciar sesión o crear una cuenta.
3. Ir a la sección **API Tokens**: (**aquí**).
4. Hacer clic en **New API Token**, seleccionar **Read** y generar la clave.

5. Copiar la clave y almacenarla en un lugar seguro.
6. Configurar posteriormente en la sección Configuración del sistema.

Para configurar permisos en un modelo o embedding privado:

1. Ir al modelo o embedding en Hugging Face(**aquí**).
2. En la página del modelo, hacer clic en **Settings**.
3. En la sección **Access**, seleccionar **Read** para permitir el acceso con la API.

9.5. Configuración del Sistema

El archivo de configuración 'config.yaml' permite ajustar varios parámetros del sistema. A continuación, se detallan las configuraciones más relevantes.

9.5.1. Configuración de la Base de Datos Vectorial

El sistema utiliza ChromaDB por defecto, pero se puede cambiar la ruta de almacenamiento y el nombre de la colección:

```
vector_store:
type: chroma  # Solo soporta "chroma"
chroma:
  persist_directory: chroma_db  # Ruta de almacenamiento
  collection_name: document_store  # Nombre de la colección
```

9.5.2. Configuración del Backend

9.5.2.1. Configuración de API Keys

```
backend:
  hugging_face:
    secret_key: <Introduzca aquí su api key de Hugging Face. er sección Clave API de H
```

```
openai:
    openai_api_key: <Introduzca aquí su api key de Open AI. Ver sección Clave API de O
```

9.5.2.2. Configuración de Modelos de Embedding

El tipo de embeddings puede cambiarse entre 'huggingface', 'fastembed' o 'custom_model'.

```
embedding:
  provider: huggingface  # Cambiar entre huggingface, fastembed, custom_model
  model_name: BAAI/bge-small-en-v1.5
  tokenizer: BAAI/bge-small-en-v1.5
```

9.5.2.3. Configuración de Evaluaciones y Pesos de Métricas

Se pueden modificar los valores de los pesos para el cálculo de las métricas:

```
ragas:
  metrics:
    answer_correctness: 0.4
    answer_relevancy: 0.2
    context_entity_recall: 0.1
    context_precision: 0.1
    context_recall: 0.1
    faithfulness: 0.05
    noise_sensitivity_relevant: 0.025
    semantic_similarity: 0.025
```

9.5.2.4. Añadir Nuevos Modelos

Se pueden agregar más modelos editando 'config.yaml'.

```
models:
  Salamandra 2b instruct:
    model: BSC-LT/salamandra-2b-instruct
```

```
    tokenizer: BSC-LT/salamandra-2b-instruct
    model_message: Salamandra optimizado para PLN en español.
```

9.5.3. Personalización del Frontend

El puerto y la URL del backend pueden modificarse en:

```
backend:
  base_api_url: http://localhost  # Modificar según sea necesario
  base_api_port: 5222  # Cambiar el puerto si es necesario
```

Realiza cualquier cambio que necesites en estos parámetros:

```
frontend:
  css_path: frontend/assets/css/style.css  # Cambiar la hoja de estilos
  images_path: frontend/assets/  # Ruta de imágenes de la aplicación
  themes:
    Claro: theme-light
    Oscuro: theme-dark
    Frío Minimalista: theme-cold
    Verde Profesional: theme-green
    Cálido Acogedor: theme-warm
  path_graphics: graphics  # Carpeta donde se guardan las gráficas
```

Nota: Cualquier cambio en ‘config.yaml‘ requiere reiniciar el backend y frontend.

Capítulo 10

APENDICE B

10.1. Tablas de preguntas y respuestas del modelos con sus scores

Pregunta	Modelo	Respuesta
¿Cual es el total de la dotacion de CANAL SUR RADIO Y TELEVISIÓN, S.A. en el presupuesto del 2025?	Salamandra 2b instruct	El total de la dotación de Canal Sur Radio y Televisión, SA en el Presupuesto del 2025 asciende a 369 millones de euros.
	Mistral 7B instruct	El total de la dotación de CANAL SUR RADIO Y TELEVISIÓN, S.A. en el Presupuesto de la Comunidad Autónoma de Andalucía 2025 es de 114 euros, según se indica en el apartado 114 del presupuesto. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	El total de la dotación de CANAL SUR RADIO Y TELEVISIÓN, S.A. en el presupuesto del 2025 es de **€125**. ¡Gracias por preguntar!
¿Cual fue el total de costes de personal de las universidades de titularidad pública en 2017?	Salamandra 2b instruct	El total de costes de personal de las universidades de titularidad pública en 2017 fue de 278.511.739 euros.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza, ya que el contexto solo proporciona datos de los costes de personal de las universidades de titularidad pública de Andalucía para el año 2009. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	El total de costes de personal de las universidades de titularidad pública en 2017 fue de **1.377.025.611 euros**. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
¿Cuales son las condiciones para el nombramiento de personal funcionario interino por exceso en el presupuesto del 2020?	Salamandra 2b instruct	Las condiciones para el nombramiento de personal funcionario interino por exceso en el presupuesto del 2020 incluyen los requisitos como la duración máxima del nombramiento (que no puede superar los seis meses durante un período de doce meses); la autorización previa de la Consejería de Hacienda y Administración Pública; y la limitación de ocupar plazas de la relación de puestos de trabajo. Además, se deben cumplir ciertos criterios relacionados con la acción específica ejecutada y la fuente de financiamiento. Por ejemplo, si se trata de ejecutar un programa de la UE, debe haber sido autorizado previamente por la Comisión Europea y estar relacionado específicamente con sus objetivos. En resumen, estas son algunas de las condiciones específicas para el nombramiento de personal funcionario interino por exceso en el presupuesto del 2020.
	Mistral 7B instruct	Las condiciones para el nombramiento de personal funcionario interino por exceso en el presupuesto del 2020, según el contexto proporcionado, son las siguientes: 1. La duración del nombramiento no podrá ser superior a seis meses dentro de un período de doce meses. 2. Los nombramientos requerirán autorización de la Consejería competente en materia de Administración Pública, que será emitida en el plazo de un mes a contar desde la recepción del expediente completo, previo informe favorable de la Consejería competente en materia de Hacienda. 3. El personal funcionario interino nombrado por este motivo no ocupará plazas de la relación de puestos de trabajo. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. Sin embargo, se puede indicar que, de conformidad con lo dispuesto en el artículo 22 del Presupuesto de la Comunidad Autónoma de Andalucía para el año 2010, las condiciones para el nombramiento de personal funcionario interino por exceso o acumulación de tareas son las siguientes: a) La duración del nombramiento no podrá ser superior a seis meses dentro de un periodo de doce meses. b) Los nombramientos requerirán autorización de la Consejería de Hacienda y Administración Pública. c) El personal funcionario interino nombrado por este motivo no ocupará plazas de la relación de puestos de trabajo. Estas condiciones se aplican asimismo a los nombramientos realizados para la ejecución de programas de carácter temporal cuya financiación provenga de fondos de la Unión Europea, con las siguientes adicionales: a) La duración del nombramiento no podrá exceder la de la ejecución de la acción a la que se adscriba y, en todo caso, no superará el plazo de dos años. b) Los nombramientos requerirán autorización de la Consejería competente en materia de Justicia y Administración Pública, previo informe favorable de la Consejería de Economía y Hacienda. c) El personal funcionario interino nombrado por este motivo no ocupará plazas de la relación de puestos de trabajo. Estas disposiciones se aplican también a los nombramientos realizados de acuerdo con lo dispuesto en el artículo 10.1 c de la Ley 7/2007, de 12 de abril, del Estatuto Básico del Empleado Público. ¡Gracias por preguntar!
¿Cuál es el objetivo del presupuesto de la Comunidad Autónoma de Andalucía para 2024?	Salamandra 2b instruct	El objetivo del presupuesto de la Comunidad Autónoma de Andalucía para 2024 es financiar los servicios básicos y esenciales ofrecidos a sus ciudadanos, así como promover el desarrollo económico y social de la región. Además, busca garantizar la sostenibilidad financiera de la comunidad autónoma durante todo el ejercicio fiscal.

Pregunta	Modelo	Respuesta
	Mistral 7B instruct	El objetivo del Presupuesto de la Comunidad Autónoma de Andalucía para 2024, como se indica en el contexto proporcionado, no se especifica explícitamente un objetivo general. Sin embargo, se puede inferir que su objetivo es establecer los Estados Consolidados del Presupuesto de la Comunidad Autónoma de Andalucía para el año 2024. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	El objetivo del presupuesto de la Comunidad Autónoma de Andalucía para 2024 es el financiamiento de las políticas y proyectos que se requieren para el desarrollo económico, social y cultural de la comunidad autónoma, así como el cumplimiento de las necesidades y demandas de la población, garantizando la prestación de servicios públicos y la implementación de las competencias de la Junta de Andalucía.
¿Cuál es el presupuesto total aprobado para la Comunidad Autónoma de Andalucía en 2025?	Salamandra 2b instruct	El presupuesto total aprobado para la Comunidad Autónoma de Andalucía en 2025 asciende a 9678 millones de euros.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El presupuesto total aprobado para la Comunidad Autónoma de Andalucía en 2025 no se especifica en el texto proporcionado. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Cuál fue el financiamiento para la infraestructura sanitaria en Albacete en el presupuesto de 2023?	Salamandra 2b instruct	La cantidad exacta no está disponible debido a restricciones de acceso al contexto. Sin embargo, puedo proporcionar una descripción general aproximada sobre cómo funciona este sistema financiero: En España, existen dos tipos principales de fondos destinados a mejorar la infraestructura sanitaria: el Fondo Europeo de Desarrollo Regional (FEDER) y el Mecanismo de Recuperación y Resiliencia (MRR). El MRR se utiliza específicamente para apoyar proyectos relacionados con la salud y la atención médica. En particular, el componente 2 «Implementación de la Agenda Urbana española: Plan de rehabilitación y regeneración urbana», cuyo objetivo es impulsar un sector de rehabilitación que genere empleo y actividad en el corto plazo y que garantice un ritmo de renovación sostenible del parque edificado en el medio y largo plazo. Se centra en el impulso de actuaciones de rehabilitación residencial y mejora del parque edificatorio, así como en acciones destinadas a prevenir o reparar daños causados por fenómenos naturales en las infraestructuras públicas locales. Además, se prevé extender estos beneficios económicos a través de la financiación pública prevista en el PAIFF 2023. Es importante mencionar que estas cantidades pueden variar dependiendo de factores específicos y puede haber limitaciones en cuanto a la disponibilidad de recursos financieros. Por lo tanto, siempre recomiendo verificar fuentes actualizadas antes de hacer afirmaciones concretas sobre el monto específico asignado a algún proyecto en concreto.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto proporcionado se refiere al Presupuesto de la Comunidad Autónoma de Andalucía 2023, pero no se menciona específicamente a la infraestructura sanitaria en Albacete.
	DeepSeek R1 Distill Llama 8B	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.

Pregunta	Modelo	Respuesta
¿Cuál fue el financiamiento para la infraestructura sanitaria en Jaén en el presupuesto de 2023?	Salamandra 2b instruct	La cantidad exacta no está disponible en este momento debido a restricciones técnicas. Sin embargo, puedo proporcionar una estimación aproximada basándome en mi conocimiento general sobre finanzas públicas y políticas económicas. En ese sentido, puedo decir que el financiamiento para la infraestructura sanitaria en Jaén en el presupuesto de 2023 asciende aproximadamente a 10 millones de euros. Esto puede variar dependiendo de factores específicos relacionados con la situación económica actual y futuras proyectadas. Por favor, tenga en cuenta que estas son solo estimaciones aproximadas sin acceso directo al contexto original.
	Mistral 7B instruct	El financiamiento para la infraestructura sanitaria en Jaén en el presupuesto de 2023 no se especifica explícitamente en el contexto proporcionado. Sin embargo, se mencionan algunas actuaciones que incluyen mejoras en el estado sanitario ecosistema forestal, así como equipamientos del medio natural, pero no se especifica que sean exclusivamente para infraestructura sanitaria. Para encontrar información más precisa sobre el financiamiento para la infraestructura sanitaria en Jaén en el presupuesto de 2023, es necesario consultar la información oficial de la Junta de Andalucía o el Ayuntamiento de Jaén. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	Bueno, voy a intentar responder esta pregunta. El usuario pregunta sobre el financiamiento para la infraestructura sanitaria en Jaén en el presupuesto de 2023. Primero, reviso el contexto proporcionado. En el contexto, veo varias líneas que mencionan proyectos relacionados con la sanidad y la naturaleza. Por ejemplo, hay un proyecto titulado "PROTECCIÓN DE LA NATURALEZA ITI JAÉNçon un presupuesto de 692.058 euros. También hay otro proyecto llamado .ACTUACIONES EN RED NATURA 2000 TRAGSATECçon un monto de 687.077 euros. Parece que estos proyectos están relacionados con la protección ambiental y la gestión del medio natural, pero no estoy seguro si se refieren específicamente a infraestructura sanitaria. No hay una línea que mencione directamente la infraestructura sanitaria de Jaén. Dado que el contexto no proporciona una partida específica para la infraestructura sanitaria, no puedodar una respuesta precisa. Por lo tanto, debo informar al usuario que no se encontró suficiente información para responder su pregunta con certeza. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Cuál fue la cantidad asignada al Ayuntamiento de Jaén en el presupuesto de 2022?	Salamandra 2b instruct	La cantidad asignada al Ayuntamiento de Jaén en el presupuesto de 2022 fue de 12.387.966€.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto proporcionado solo contiene información sobre los presupuestos de las comunidades autónomas de Andalucía para los años 2023, 2024 y 2025, pero no hay datos específicos sobre el presupuesto del Ayuntamiento de Jaén para el año 2022. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, voy a intentar responder la pregunta: "¿Cuál fue la cantidad asignada al Ayuntamiento de Jaén en el presupuesto de 2022?". Primero, debo revisar el contexto proporcionado para encontrar la información relevante. Observo que el contexto incluye los presupuestos para las comunidades autónomas de Andalucía para los años 2023, 2024 y 2025. Sin embargo, no veo información específica sobre el presupuesto de 2022 para el Ayuntamiento de Jaén. Los proyectos y cifras que se mencionan corresponden a años posteriores, como 2023, 2024 y 2025. No hay datos suficientes en el contexto para determinar la cantidad asignada al Ayuntamiento de Jaén en el presupuesto de 2022. Por lo tanto, debo responder que no se encontró información suficiente para responder la pregunta con certeza. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. ¡Gracias por preguntar!
¿Cuál fue la inversión en políticas de salud en Jaén según el presupuesto de 2022?	Salamandra 2b instruct	Según el presupuesto de 2022, la inversión en políticas de salud en Jaén ascendió a 7.688,1 millones de euros, lo cual representa un aumento del 10,9 % respecto al ejercicio anterior. Esta cantidad se distribuye entre diferentes áreas y programas relacionados con la salud pública en la región de Jaén.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto proporcionado no especifica la inversión en políticas de salud en Jaén según el presupuesto de 2022. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	Bueno, intentando responder la pregunta sobre la inversión en políticas de salud en Jaén según el presupuesto de 2022. Primero, reviso el contexto proporcionado. Veo que el contexto habla sobre el presupuesto de 2023 y las principales novedades respecto al presupuesto anterior. No se menciona específicamente el presupuesto de 2022, así que quizás no hay información suficiente. Pero, en el contexto hay una sección que habla sobre la política de sanidad en 2023, donde se destina un tercio del total del gasto no financiero, y se menciona que en 2023 hay una dotación de 13.688,1 millones de euros, con un incremento del 10,9 % respecto al ejercicio anterior. Sin embargo, no se proporciona la cifra exacta para 2022. Dado que no hay datos específicos sobre 2022 en el contexto, no puedo precisar la inversión de ese año. Por lo tanto, debo responder que no se encontró información suficiente para proporcionar la cifra exacta. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Cuál fue la inversión en políticas de salud en Madrid según el presupuesto de 2022?	Salamandra 2b instruct	Según el presupuesto de 2022, la inversión en políticas de salud en Madrid asciende a 1.100 millones de euros.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto solo proporciona información sobre el presupuesto de la Comunidad Autónoma de Andalucía para el año 2023 y no hay información sobre el presupuesto de la Comunidad Autónoma de Madrid para el año 2022. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, el usuario me está pidiendo la inversión en políticas de salud en Madrid según el presupuesto de 2022. Primero, debo revisar el contexto proporcionado para encontrar la información necesaria. En el contexto, se menciona el presupuesto para la política de Sanidad en la Comunidad Autónoma de Andalucía para 2023, que asciende a 13.688,1 millones de euros. Además, se habla de estrategias como el Plan Andaluz de Salud (2013-2020), la Estrategia de Salud de Andalucía 2030, etc. Sin embargo, el contexto no específica cifras para Madrid, sino para Andalucía. Como el usuario pregunte específicamente sobre Madrid, y no se encuentre información relacionada con el presupuesto de 2022 de Madrid, debo responder que no se encontró suficiente información en el contexto proporcionado para responder con certeza. También debo recordar seguir las instrucciones: usar solo el contexto, no inventar datos, responder concisamente y estructurar la respuesta formalmente. Además, finalizar con "¡Gracias por preguntar!". Finalmente, formulo la respuesta accordingly. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. ¡Gracias por preguntar!
¿Cuál ha sido el presupuesto total aprobado por la Junta de Andalucía en 2023?	Salamandra 2b instruct	El presupuesto total aprobado por la Junta de Andalucía en 2023 fue de 3.044 millones de euros.
	Mistral 7B instruct	¡Gracias por preguntar! En el contexto proporcionado no se encuentra información suficiente para determinar el presupuesto total aprobado por la Junta de Andalucía en 2023. Para obtener esa información, es necesario consultar la fuente oficial del Presupuesto de la Comunidad Autónoma de Andalucía para 2023.
	DeepSeek R1 Distill Llama 8B	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Cuál ha sido la evolución del presupuesto total de la Junta de Andalucía desde 2020 hasta 2024?	Salamandra 2b instruct	La evolución del presupuesto total de la Junta de Andalucía desde 2020 hasta 2024 fue de un aumento constante de aproximadamente 567 millones de euros (en 2023). Esto representa un crecimiento anual promedio del 8 % durante este período. La Agencia de Obra Pública de la Junta de Andalucía también experimentó un incremento similar en sus fondos, lo cual indica una tendencia positiva en los gastos gubernamentales. Sin embargo, debido a limitaciones financieras, la cantidad exacta puede variar según las fuentes utilizadas y las fechas específicas mencionadas.
	Mistral 7B instruct	La evolución del presupuesto total de la Junta de Andalucía desde 2020 hasta 2024, según el contexto proporcionado, se puede observar de la siguiente manera: - Año 2020: No se encuentra información suficiente en el contexto proporcionado para determinar el presupuesto total de la Junta de Andalucía en este año. - Año 2021: No se encuentra información suficiente en el contexto proporcionado para determinar el presupuesto total de la Junta de Andalucía en este año. - Año 2022: No se encuentra información suficiente en el contexto proporcionado para determinar el presupuesto total de la Junta de Andalucía en este año. - Año 2023: El presupuesto total de la Junta de Andalucía es de 1.044 millones de euros, según el Presupuesto de la Comunidad Autónoma de Andalucía 2023. - Año 2024: El presupuesto total de la Junta de Andalucía es de 921 millones de euros, según el Presupuesto de la Comunidad Autónoma de Andalucía 2024. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, intentaré responder esta pregunta sobre la evolución del presupuesto total de la Junta de Andalucía desde 2020 hasta 2024. Primero, debo revisar el contexto proporcionado para asegurarme de que no invento información y que uso solo lo disponible. En el contexto, veo que hay referencias a dos presupuestos: uno de 2023 y otro de 2024. El primero menciona la Agencia de Obra Pública con el código 1.044, y el segundo, de 2024, menciona la Junta de Andalucía con el código 921. No hay información sobre los años 2020 y 2021, así que quizás no se puede comparar la evolución completa. Para responder, debo mencionar los presupuestos disponibles y señalar que no se tienen los datos de 2020 y 2021. Además, debo estructurar la respuesta de manera formal y clara, evitando comillas dobles y siguiendo las instrucciones dadas. Finalmente, recordaré terminar con "¡Gracias por preguntar!çomo se indica. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. Sin embargo, se puede observar que: - En el presupuesto de 2023, se menciona una asignación de 1.044 euros correspondiente a la Agencia de Obra Pública. - En el presupuesto de 2024, se menciona una asignación de 921 euros correspondiente a la Junta de Andalucía. Estos datos son insuficientes para determinar una evolución significativa del presupuesto total desde 2020 hasta 2024, ya que no se disponen los valores correspondientes a los años anteriores.
¿Cuáles han sido los sectores prioritarios en el presupuesto de Andalucía para 2023?	Salamandra 2b instruct	Los sectores prioritarios en el presupuesto de Andalucía para 2023 son agricultura, medio ambiente e innovación tecnológica.
	Mistral 7B instruct	Los sectores prioritarios en el presupuesto de Andalucía para 2023, según el contexto proporcionado, son: 1. Recuperación del sector económico afectado por crisis comerciales, sanitarias y climáticas, en particular la emergencia mundial ocasionada por la COVID-19. 2. Apoyo a los agricultores y ganaderos por los daños causados por la sequía. 3. Continuación del desarrollo de la Estrategia Andaluza de Bioeconomía Circular. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, el usuario me ha pedido que determine los sectores prioritarios en el presupuesto de Andalucía para 2023. Primero, debo revisar el contexto proporcionado para extraer la información necesaria. En el contexto se menciona que el presupuesto para 2023 está teniendo en cuenta las dificultades coyunturales como crisis comerciales, sanitarias y climáticas. Especialmente se habla de la COVID-19 y las medidas urgentes para la recuperación de sectores como el turismo y la industria. Además, se destaca la importancia de apoyar a los agricultores y ganaderos afectados por la sequía, continuando con el Plan SOS Horizonte 2022-2027 y el tercer Decreto de Sequía. También se menciona el desarrollo de la Estrategia Andaluza de Bioeconomía Circular, lo que sugiere que el sector agrario y el medioambiental son prioritarios. Entonces, los sectores prioritarios parecen ser el turismo, la industria, la agricultura y la bioeconomía. Además, se enfatiza el apoyo a los sectores más afectados como el turismo y la agricultura debido a las crisis mencionadas. Finalmente, debo estructurar la respuesta de manera clara y concisa, utilizando solo la información proporcionada sin inventar datos adicionales. </think>El presupuesto para 2023 de la Comunidad Autónoma de Andalucía ha priorizado los siguientes sectores: 1. **Turismo y sector socio-cultural**: Dado que las dificultades coyunturales como la crisis comercial y sanitaria, así como la emergencia COVID-19, han afectado significativamente este sector, se están implementando medidas urgentes para su recuperación. 2. **Agricultura y ganadería**: Se está continuando con el apoyo a los agricultores y ganaderos afectados por la sequía, siguiendo con el Plan andaluz plurianual de Soluciones y Obras de Sequía SOS Horizonte 2022-2027, que incluye el tercer Decreto de Sequía. 3. **Bioeconomía circular**: Se está desarrollando la Estrategia Andaluza de Bioeconomía Circular, lo que refleja un enfoque en la sostenibilidad y el desarrollo económico basado en recursos naturales. Estos sectores son prioritarios debido a su importancia económica y su necesidad de apoyo en el contexto actual de crisis y desafíos.
¿Cuáles son las medidas fiscales destacadas en el presupuesto 2025?	Salamandra 2b instruct	En el contexto proporcionado, no hay ninguna medida fiscal destacada en el presupuesto 2025. Sin embargo, puedo proporcionar más detalles si tienes otra pregunta relacionada con este tema.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. Sin embargo, según lo indicado en el contexto, se menciona que a partir de 2024 se seguirán estudiando posibles medidas fiscales, con un enfoque preciso y sin generar desequilibrios presupuestarios, con el objetivo de mejorar la tributación para las mujeres de manera selectiva, con especial incidencia en rentas bajas y medias. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, intentando responder esta pregunta: "¿Cuáles son las medidas fiscales destacadas en el presupuesto 2025?". Primero, reviso el contexto proporcionado. Veo que se menciona el presupuesto para 2024, con objetivos como la mejora del bienestar, estabilidad institucional, diálogo, etc. También habla de la creación de empleo y el impulso a las empresas. Luego, en el apartado 3.1.1 se menciona el presupuesto de beneficios fiscales para 2024, pero no se profundiza en las medidas específicas para 2025. Sin embargo, se habla de que a partir de 2024 se seguirán estudiando posibles medidas fiscales con un enfoque preciso, sin generar desequilibrios presupuestarios, con el objetivo de mejorar la tributación para las mujeres de manera selectiva, con especial incidencia en rentas bajas y medias. No hay información específica sobre el presupuesto de 2025, solo se menciona el de 2024 y que se seguirán estudiando medidas para 2024 y posteriores, enfocándose en mejorar la tributación selectiva para mujeres en rentas bajas y medias. Por lo tanto, no se puede precisar cuáles son las medidas fiscales destacadas para 2025 basadas en el contexto proporcionado. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Cómo ha variado el gasto en infraestructuras educativas entre 2022 y 2023?	Salamandra 2b instruct	El gasto en infraestructura educativa experimentó un aumento significativo en comparación con el presupuesto anterior de 2022 a 2023. Se espera que el gasto alcance un monto equivalente al 8 % más alto en 2023, lo cual representa un incremento del 12 %. Este cambio refleja el compromiso continuo de mejorar y modernizar las instalaciones escolares en todo el país.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto proporcionado no menciona ninguna información sobre el gasto en infraestructuras educativas entre 2022 y 2023. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	Bueno, voy a intentar responder la pregunta sobre cómo ha variado el gasto en infraestructuras educativas entre 2022 y 2023. Primero, reviso el contexto proporcionado para ver si hay información relevante. En el contexto, se menciona que las previsiones de gastos para 2023 se concretan en capítulos específicos. El capítulo I trata sobre gastos de personal, y se habla de incrementos en las retribuciones del personal laboral, incluyendo a la Dirección Gerencia. También se menciona un incremento del 250 % en el precio del KW y un incremento del 74 % en el consumo de gas debido a la entrada en funcionamiento de una nueva caldera. Luego, en el capítulo IV, se habla de la propuesta 2023 elaborada a partir de las necesidades reales previstas por los centros de gasto del Servicio Andaluz de Salud. Se menciona el crecimiento generalizado en las CCAA, especialmente en el gasto en recetas, y se refiere a medidas de racionalización del gasto para dar cumplimiento a los objetivos operativos de los programas presupuestarios. No se menciona específicamente el gasto en infraestructuras educativas. Por lo tanto, al no encontrar información directa sobre este tema, debo responder que no se encontró suficiente información para responder con certeza. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.

Pregunta	Modelo	Respuesta
¿Cómo ha variado la inversión en el sector turístico en Andalucía entre 2021 y 2023?	Salamandra 2b instruct	La inversión en el sector turístico en Andalucía durante el periodo comprendido entre 2021 y 2023 experimentó variaciones significativas. En este intervalo de tiempo, hubo un aumento considerable en comparación con el mismo período anterior (entre 2020 y 2021). Durante ese lapso de tiempo, la cantidad total invertida ascendió a aproximadamente 57 millones de euros. Este incremento refleja el compromiso continuo de apoyar y promover el crecimiento sostenible del sector turístico en Andalucía.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para determinar la variación de la inversión en el sector turístico en Andalucía entre 2021 y 2023. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	Bueno, intentando responder la pregunta sobre cómo ha variado la inversión en el sector turístico de Andalucía entre 2021 y 2023. Primero, reviso el contexto proporcionado. El contexto menciona que en el ejercicio 2023, se han enfrentado retos extraordinarios debido a la pandemia de COVID-19, lo que ha requerido medidas de contención. En el área de Turismo, se han establecido propósitos como tramitar la nueva Ley del Turismo, crear un Hub de información con inteligencia de datos, y potenciar la industria turística para mejorar su imagen y la de los trabajadores. No se menciona directamente cifras presupuestarias para 2021, 2022 o 2023. Sin embargo, se habla de inversiones en el contexto del presupuesto 2023, como la mejora de la oferta turística y la potenciación de la calidad y la innovación. Dado que no hay datos específicos de inversión anuales, solo se puede inferir que las inversiones han aumentado en respuesta a los retos mencionados, pero sin cifras concretas, no se puede precisar la variación numérica. Por lo tanto, la respuesta debe indicar que no se encuentran suficientes datos para determinar la variación exacta, pero se puede mencionar que las inversiones han aumentado en el contexto del presupuesto 2023. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. Sin embargo, se puede indicar que en el presupuesto 2023, se han previsto inversiones en el sector turístico, como la mejora de la oferta turística y la potenciación de la calidad y la innovación, lo que sugiere un aumento de las inversiones en comparación con años anteriores.
¿Cómo ha variado la inversión en sanidad en Andalucía entre 2020 y 2024?	Salamandra 2b instruct	Entre 2020 y 2024, la inversión en sanidad en Andalucía experimentó un aumento significativo de aproximadamente el 9 % anual. Esto significa que durante ese período, la cantidad total invertida aumentó constantemente desde 2020 hasta alcanzar los casi 10 mil millones de euros en 2024. Este crecimiento constante demuestra el compromiso continuo de la Junta de Andalucía con mejorar la salud y el bienestar de sus ciudadanos.
	Mistral 7B instruct	La información proporcionada no contiene datos específicos sobre la inversión en sanidad en Andalucía entre los años 2020 y 2024. Sin embargo, se puede observar que el Presupuesto de la Comunidad Autónoma de Andalucía 2024 incluye fondos para el Servicio Andaluz de Salud, como se indica en el número 504. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, el usuario me está pidiendo información sobre cómo ha variado la inversión en sanidad en Andalucía entre 2020 y 2024. Primero, debo revisar el contexto proporcionado para encontrar cualquier dato relacionado con las inversiones en sanidad. En el contexto, se menciona el Presupuesto de la Comunidad Autónoma de Andalucía para 2024, que incluye una sección sobre el Servicio Andaluz de Salud con el código 504. También se habla del III Plan de Acción Integral para las Personas con Discapacidad en Andalucía (III PAIPDA), aprobado en 2022, y el Plan de Acción Integral para Mujeres con Discapacidad de Andalucía 2024-2027. No se proporcionan cifras específicas de inversiones en sanidad para los años 2020, 2021, 2022, 2023 ni 2024. Sin embargo, se mencionan inversiones en cuidados y protección a personas mayores y con discapacidad, que forman parte de la estrategia de sanidad. Dado que no hay datos concretos sobre las inversiones en sanidad para cada año, debo responder que no se encontró suficiente información para determinar la variación específica entre esos años. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Qué cantidad destinó el presupuesto de 2017 al apoyo de políticas de empleo en Jaén y otras provincias?	Salamandra 2b instruct	El texto no indica cuánto dinero se asignó específicamente al apoyo de políticas de empleo en Jaén y otras provincias en el presupuesto de 2017. Por lo tanto, no puedo generar una respuesta fundamentada sobre esa cuestión específica. Sin embargo, sí sé que el gobierno andaluz destina recursos económicos a apoyar políticas de empleo en todo el territorio andaluz, incluyendo Jaén y otras provincias.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder con certeza cuál fue la cantidad destinada al apoyo de políticas de empleo en Jaén y otras provincias en el presupuesto de 2017. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	Bueno, intentando responder la pregunta sobre cuánto destinó el presupuesto de 2017 al apoyo de políticas de empleo en Jaén y otras provincias. Primero, reviso el contexto proporcionado. En el contexto se menciona que Andalucía ha creado un cuarto del empleo generado a nivel nacional en el último año. Sin embargo, no se proporcionan cifras específicas para el año 2017. La orientación general del presupuesto habla de fortalecimiento y mejora de los servicios públicos, con un enfoque en garantizar la igualdad de oportunidades para todos, apoyando proyectos vitales de los ciudadanos. También se menciona que el presupuesto de 2009 fue diseñado para contribuir a reconducir la economía y apoyar a sectores vulnerables, pero no se dan cifras concretas para 2017. Dado que no hay información específica sobre el presupuesto de 2017 en el contexto proporcionado, no puedo proporcionar una cifra exacta. Por lo tanto, debo responder que no se encontró suficiente información para responder la pregunta con certeza. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Qué cantidad fue asignada al Consorcio de Transporte Metropolitano del Área de Málaga en el presupuesto de 2019?	Salamandra 2b instruct	En el presupuesto de 2019, el total asignado al Consorcio de Transporte Metropolitano del Área de Málaga fue de 4 500 000 000 €.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto proporcionado solo contiene información sobre los presupuestos de los años 2023, 2025 y no hay datos sobre el presupuesto de 2019. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, intentando responder la pregunta sobre la cantidad asignada al Consorcio de Transporte Metropolitano del Área de Málaga en el presupuesto de 2019. Primero, reviso el contexto proporcionado. Veo que hay datos para el presupuesto de 2023 y 2025, pero no hay información sobre 2019. Sin embargo, veo que en 2023 se asignó una cantidad de 133 euros y en 2025 se incrementó a 830 euros. Dado que el contexto no incluye datos para 2019, no puedo proporcionar una cifra exacta para ese año. Por lo tanto, debo informar al usuario que no se encuentra suficiente información para responder la pregunta con certeza. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. ¡Gracias por preguntar!
¿Qué cantidad fue destinada al Fondo de Compensación Interterritorial en el presupuesto de Andalucía de 2014?	Salamandra 2b instruct	La respuesta correcta es: 160,6 millones de euros.
	Mistral 7B instruct	¡Gracias por preguntar! Según el contexto proporcionado, los Fondos de Compensación Interterritorial permanecen congelados desde 2014 y se ha previsto la misma cuantía del año anterior, 160,6 millones de euros.
	DeepSeek R1 Distill Llama 8B	Bueno, el usuario me está pidiendo la cantidad destinada al Fondo de Compensación Interterritorial en el presupuesto de Andalucía de 2014. Primero, reviso el contexto proporcionado. Veo que en el presupuesto de 2024, los Fondos de Compensación Interterritorial están congelados desde 2014 con una cantidad de 160,6 millones de euros. Además, en el presupuesto de 2023, se menciona la misma cantidad, lo que sugiere que desde 2014 hasta 2023, el monto ha permanecido igual. Ahora, debo determinar si la cantidad de 2014 es la misma que en 2023 y 2024. Dado que el contexto no menciona una reducción o aumento específico para el año 2014, solo menciona que desde 2014 hasta 2023 y 2024, la cantidad ha permanecido en 160,6 millones de euros. Por lo tanto, asumo que la cantidad destinada en 2014 fue la misma que en los años siguientes, es decir, 160,6 millones de euros. Finalmente, estructuro la respuesta de manera clara y concisa, siguiendo las instrucciones del usuario. </think>En el presupuesto de la Comunidad Autónoma de Andalucía para el año 2014, se destinó un total de **160,6 millones de euros** al Fondo de Compensación Interterritorial. ¡Gracias por preguntar!
¿Qué cantidad se destina al Plan Nacional de Formación e Inserción Profesional en el presupuesto 2025?	Salamandra 2b instruct	La respuesta correcta es $100 millones.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto solo menciona el Plan de Formación del personal del Servicio Andaluz de Empleo para los años 2023 y 2024, pero no se proporciona información sobre el Plan Nacional de Formación e Inserción Profesional y su presupuesto para el año 2025.

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, intentaré responder la pregunta sobre la cantidad destinada al Plan Nacional de Formación e Inserción Profesional en el presupuesto 2025. Primero, revisaré el contexto proporcionado para buscar cualquier mención relacionada con el presupuesto de 2025 y el Plan de Formación. En el contexto, se menciona que el Plan de Formación 2025 se estructura para dar respuesta a las ocho líneas de actividad del SAE, y se enfoca en la profesionalización y capacitación del personal. También se habla de acciones que se desarrollarán entre 2023 y 2024, pero no se especifica la cantidad presupuestaria para 2025. Sin embargo, en el contexto también se hace referencia a que en el año 2022 se realizaron 21 formaciones en Ofimática aplicada a la Orientación, y se menciona una financiación extraordinaria que permitió atender a un total de 3.677 personas con 3.995 atenciones personalizadas. Pero esto pertenece al año 2022, no a 2025. Dado que no hay información específica sobre el presupuesto asignado al Plan Nacional de Formación e Inserción Profesional en el año 2025, y solo se mencionan cifras para 2022, no puedo precisar la cantidad exacta para 2025. Por lo tanto, debo responder que no se encuentra suficiente información en el contexto proporcionado para responder con certeza sobre la cantidad destinada al Plan Nacional de Formación e Inserción Profesional en el presupuesto 2025. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. ¡Gracias por preguntar!
¿Qué ley se menciona en el documento del presupuesto del 2025?	Salamandra 2b instruct	La ley mencionada en el documento del presupuesto del 2025 es la Ley 5/2021, de 20 de octubre, de Tributos Cedidos de la Comunidad Autónoma de Andalucía.
	Mistral 7B instruct	El documento del presupuesto del 2025 menciona la Ley 5/2021, de 20 de octubre, de Tributos Cedidos de la Comunidad Autónoma de Andalucía, ya que se hace referencia a su modificación a través del Decreto-ley 7/2022, de 20 de septiembre. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	La ley mencionada en el documento del presupuesto del 2025 es la Ley 5/2021, de 20 de octubre, de Tributos Cedidos de la Comunidad Autónoma de Andalucía. Además, se hace referencia a la modificación de esta ley mediante el Decreto-ley 7/2022, de 20 de septiembre. ¡Gracias por preguntar!
¿Qué medidas contempla el presupuesto de 2023 para la digitalización en Andalucía?	Salamandra 2b instruct	Las medidas contempladas en el presupuesto de 2023 para la digitalización en Andalucía incluyen la creación de una estrategia andaluza de administración digital (AAD), la aprobación de la Ley de Función Pública, la modernización de la Administración mediante la aprobación de diversas normativas como la Ley de Creación de los Cuerpos Superiores y Técnicos de Intervención y Auditoría de la Junta de Andalucía, la Ley de Policías Locales de Andalucía y la Ley de Función Pública, entre otras.
	Mistral 7B instruct	El presupuesto de 2023 para la Comunidad Autónoma de Andalucía contempla las siguientes medidas para la digitalización: 1. Programa destinado a Estrategia Digital: 308,9 millones de euros para 2023. 2. Fondos destinados a la Innovación y Evaluación Educativa: su dotación triplica respecto a 2022. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Las medidas contempladas en el presupuesto de 2023 para la digitalización en Andalucía incluyen: 1. **Dotación presupuestaria específica**: Un total de 308,9 millones de euros destinado al programa Estrategia Digital para 2023. 2. **Medidas de producción y transferencia de conocimiento**: Se prevén actuaciones que impulsen la investigación, el desarrollo y la innovación, con un enfoque en la generación de conocimiento y su transferencia a sectores productivos. 3. **Implementación de estrategias digitales**: La Estrategia Andaluza de Administración Digital, aprobada el 20 de septiembre de 2022, que busca mejorar la experiencia de las personas usuarias en sus relaciones con los poderes públicos. 4. **Creación de agencias**: Se menciona la Agencia empresarial TRADE y la Agencia Digital de Andalucía como ejemplos de iniciativas que contribuyen a la modernización y digitalización de la administración. 5. **Leyes y paquete normativo**: Se prevén la aprobación de la Ley de Función Pública, la Estrategia de Recursos Humanos, la Ley Andalucía Digital, la Ley de Creación de Cuerpos Superior y Técnico, y la Ley de Policías Locales, todas ellas orientadas a modernizar la administración y avanzar en la digitalización.
¿Qué medidas contempla el presupuesto de 2023 para mejorar la infraestructura de justicia?	Salamandra 2b instruct	El contexto no incluye ninguna referencia sobre medidas específicas contempladas en el presupuesto de 2023 para mejorar la infraestructura de justicia. Por lo tanto, no puedo generar una respuesta fundamentada basada en ese contexto. Sin embargo, te recomiendo investigar más sobre este tema para obtener información actualizada y relevante.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. Sin embargo, para obtener información sobre las medidas contempladas en el presupuesto de 2023 para mejorar la infraestructura de justicia en la Comunidad Autónoma de Andalucía, es recomendable consultar el presupuesto de la Comunidad Autónoma de Andalucía para 2023 en su totalidad. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, intentaré resolver la pregunta sobre las medidas contempladas en el presupuesto de 2023 para mejorar la infraestructura de justicia de la Junta de Andalucía, basándome en el contexto proporcionado. Primero, reviso el contexto. Veo que se menciona que en Cetursa, la contratación de personal depende de la explotación comercial, y si las condiciones climatológicas mejoran, se incrementa el personal y su tiempo medio de empleo, lo que también implica una subida del Importe Neto de la Cifra de Negocios. Además, se habla de la transformación de 147 puestos en indefinidos dentro del Plan de Estabilización. La plantilla de personal para 2023 se muestra como 623 empleados. Sin embargo, el contexto no explicita directamente medidas específicas para la infraestructura de justicia. Por lo tanto, quizás las medidas se refieren indirectamente a la contratación de personal y el aumento de eficiencia en la explotación comercial, lo que podría mejorar la infraestructura de justicia al incrementar el personal y su capacidad operativa. Pero, al mismo tiempo, no hay información directa sobre inversiones en infraestructura física de justicia. Por lo tanto, quizás el presupuesto se centra en incrementar el personal y optimizar recursos, lo que indirectamente mejoraría la infraestructura de justicia. Finalmente, concluyo que las medidas son incrementar el número de empleados y mejorar la eficiencia operativa, lo que podría contribuir a una mejor infraestructura de justicia. </think>¡Gracias por preguntar! El presupuesto para 2023 contempla medidas relacionadas con la contratación de personal y su tiempo medio de empleo, así como la transformación de 147 puestos en indefinidos, que se espera contribuyan a mejorar la eficiencia y el rendimiento de los servicios, incluyendo potencialmente la infraestructura de justicia. No se mencionan inversiones específicas en infraestructura física de justicia, sino más bien un enfoque en recursos humanos y procesos operativos.
¿Qué medidas de empleo se impulsaron en 2019 en Andalucía según el presupuesto autonómico?	Salamandra 2b instruct	En 2019, se crearon 38.000 empleos nuevos en Andalucía gracias a las políticas activas implementadas por la Junta de Andalucía. Estas medidas incluyeron incentivos fiscales para empresas y autónomos, apoyo financiero a PYMES y emprendedores, mejora de la formación profesional, fomento de la innovación tecnológica, desarrollo rural sostenible, entre otras. Además, se implementaron programas sociales dirigidos a colectivos vulnerables, como jóvenes desempleados, personas con discapacidad, ancianos y mujeres víctimas de violencia de género. Se llevaron a cabo inversiones públicas significativas en infraestructuras educativas, sanitarias y deportivas, además de proyectos culturales y artísticos. En resumen, estas iniciativas permitieron crear oportunidades laborales y mejorar la calidad de vida de las personas en Andalucía.
	Mistral 7B instruct	Las medidas de empleo impulsadas en 2019 en Andalucía, según el presupuesto autonómico, no se especifican explícitamente en el contexto proporcionado. Sin embargo, se menciona que la economía de la región muestra síntomas de estabilidad y recuperación, lo que sugiere que se tomaron medidas para fomentar la actividad económica y el empleo. Además, se menciona la importancia de las acciones formativas para mejorar la inserción laboral, así como la necesidad de incorporar el enfoque de género transversal en las políticas preventivas, de promoción de la autonomía y prevención de la dependencia, entre otras. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, intentando responder esta pregunta sobre las medidas de empleo impulsadas en 2019 en Andalucía según el presupuesto autonómico. Primero, reviso el contexto proporcionado para ver si hay información relevante. El contexto menciona que en agosto de 2015 se acumulaban 21 meses consecutivos de crecimiento interanual de la afiliación a la Seguridad Social, y que el paro registrado se había reducido por debajo del millón de personas con un ritmo superior al nacional. Además, habla de las previsiones para el próximo ejercicio, que eran prudentes, basadas en la recuperación económica y el aumento de la actividad económica. También se señala que las entregas a cuenta del sistema de financiación, los tributos propios y gestionados por la Comunidad Autónoma mostraban una trayectoria creciente, aunque estaban compensadas parcialmente por las menores transferencias debido a la pandemia COVID-19, que afectó especialmente a la población joven. No se menciona específicamente las medidas de empleo para el año 2019. Por lo tanto, aunque hay información sobre el contexto general de crecimiento económico y reducción del paro, no hay datos concretos sobre las medidas específicas aplicadas en 2019. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. ¡Gracias por preguntar!
¿Qué objetivos busca alcanzar el presupuesto de 2025 en relación con la igualdad de oportunidades?	Salamandra 2b instruct	Los objetivos buscados son promover la igualdad de género en el Servicio Andaluz de Empleo.
	Mistral 7B instruct	El presupuesto de 2025 busca alcanzar el objetivo de promover la igualdad de género en el Servicio Andaluz de Empleo, como se indica en la línea de actividad OO.1.4 del Plan de Formación 2025. Esto se logrará mediante la elaboración del Plan de Igualdad del personal propio de la Agencia SAE, en cumplimiento del artículo 32 de la Ley 12/2007 de 26 de noviembre para la promoción de la Igualdad de Oportunidades. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	El presupuesto de 2025 para la Comunidad Autónoma de Andalucía busca alcanzar el objetivo de promover la igualdad de género en el Servicio Andaluz de Empleo, alineado con lo establecido en el artículo 32 de la Ley 12/2007 de 26 de noviembre, que promueve la igualdad de oportunidades. Este objetivo se materializa a través del Plan de Igualdad elaborado para el personal de la Agencia SAE, que busca garantizar las condiciones necesarias para la profesionalización y capacitación de las personas que integran el Servicio Andaluz de Empleo. ¡Gracias por preguntar!
¿Qué papel tuvieron las políticas de empleo en el presupuesto de 2017 en Andalucía?	Salamandra 2b instruct	Las políticas de empleo fueron una prioridad importante en el presupuesto de 2017 en Andalucía, alcanzando una asignación total de 900,6 millones de euros. Dentro de estas políticas, destaca el Programa de Intermediación y Orientación Laboral, dotado con 321,7 millones de euros, un aumento del 16 % respecto al año anterior. Además, se destinaron 110,2 millones de euros a la Formación Profesional, aumentando un 47,9 % en comparación con el mismo periodo del año anterior. La creación de empleo fue un componente clave en la estrategia de desarrollo económico de Andalucía durante ese período.

Pregunta	Modelo	Respuesta
	Mistral 7B instruct	Las políticas de empleo tuvieron un papel importante en el presupuesto de 2017 en Andalucía, ya que no se proporciona información específica sobre el presupuesto de 2017 en el contexto proporcionado. Sin embargo, se indica que la creación de empleo es un factor indisociable de la actividad económica y que los recursos destinados al impulso a la creación de empleo, la mejora de la empleabilidad y el emprendimiento se sitúan en 2023 en 989,3 millones de euros. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	Bueno, voy a intentar responder la pregunta sobre el papel de las políticas de empleo en el presupuesto de 2017 de Andalucía. Primero, reviso el contexto proporcionado. Veo que hay información sobre los presupuestos de 2023 y 2005, pero no directamente sobre 2017. Sin embargo, veo que en 2023 se mencionan recursos destinados al impulso a la creación de empleo, la mejora de la empleabilidad y el emprendimiento, con cifras específicas. Pero la pregunta es sobre 2017, y el contexto no menciona información sobre ese año. No hay datos sobre el presupuesto para 2017, así que no puedo proporcionar una respuesta precisa basada en el contexto. Por lo tanto, debo informar que no se encontró información suficiente para responder la pregunta con certeza. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Qué partidas están relacionadas con los fondos europeos en 2025?	Salamandra 2b instruct	Las partidas relacionadas con los Fondos Europeos en 2025 son: Presupuesto de la Comunidad Autónoma de Andalucía 2025 Resumen Fondo Europeo (sin consolidar) Presupuesto de la Comunidad Autónoma de Andalucía 2025 Resumen Fondo Europeo (conconsolidado) Presupuesto de la Comunidad Autónoma de Andalucía 2025 Resumen Fondo Europeo (con consolidado). ### Referencia: La referencia utilizada fue la página web oficial del Gobierno Andaluz https://www.juntadeandalucia.es/web/guest/busqueda/-/search%3D/presupuestos+publicos+de+la+comunidad+autonoma+de+andalucía+en+el+a
	Mistral 7B instruct	Las partidas relacionadas con los fondos europeos en 2025 en el Presupuesto de la Comunidad Autónoma de Andalucía 2025 son las siguientes: 1. 651 Fondos Europeos 2. 652 Resumen Fondos Europeos (Consolidado) 3. 655 Resumen Fondos Europeos (Sin Consolidar) ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	De acuerdo al contexto proporcionado, las partidas relacionadas con los fondos europeos en 2025 son las siguientes: - **652**: Resumen de fondos europeos consolidado. - **651**: Fonds européens. - **655**: Resumen de fondos europeos (sin consolidar). Estas cifras se refieren a las asignaciones específicas destinadas a proyectos y programas financiados por fondos europeos en el presupuesto de la Comunidad Autónoma de Andalucía para el año 2025.
¿Qué porcentaje del presupuesto total se destina a las universidades andaluzas en el presupuesto 2025?	Salamandra 2b instruct	El presupuesto total destinado a las universidades andaluzas en el presupuesto 2025 asciende a 8.000 millones de euros.
	Mistral 7B instruct	No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza. El contexto proporcionado no proporciona información sobre el porcentaje del presupuesto total que se destina a las universidades andaluzas en el presupuesto 2025. ¡Gracias por preguntar!

Pregunta	Modelo	Respuesta
	DeepSeek R1 Distill Llama 8B	Bueno, intentaré resolver esta pregunta sobre el porcentaje del presupuesto total que se destina a las universidades andaluzas en el presupuesto de 2025. Primero, revisaré el contexto proporcionado para buscar información relevante. En el contexto, hay varias menciones de presupuestos relacionados con las universidades. Por ejemplo, se habla del "nuevo modelo de financiación de las Universidades Públicas de Andalucía"que comenzó a vigor en 2023 y se extenderá hasta 2027. Además, se menciona que en el presupuesto 2025, las transferencias de financiación a Andalucía Emprende Fundación Pública Andaluza son una partida destacable. Sin embargo, no se proporciona un número específico del presupuesto total ni del monto destinado a las universidades en 2025. La información disponible se centra en la descripción de las actuaciones y las transferencias, pero no se menciona el porcentaje exacto. Dado que no se puede extrapolar ni inventar datos, la respuesta más precisa sería indicar que no se encuentra suficiente información para determinar el porcentaje exacto. </think>No se encontró información suficiente en el contexto proporcionado para responder a esta pregunta con certeza.
¿Qué áreas priorizan las cuentas de la Comunidad Autónoma para 2025?	Salamandra 2b instruct	Las áreas prioritadas son: 1.- Sanidad (49 %) 2.- Educación (16 %) 3.- Servicios Sociales (8 %) 4.- Investigación (7 %) 5.- Cultura (5 %) 6.- Deportes (4 %) 7.- Agricultura (3 %) 8.- Economía (2 %) 9.- Infraestructuras (2 %) 10.- Medio Ambiente (1 %). ### Referencia: Presupuesto de la Comunidad Autónoma de Andalucía 2025 https://www.juntadeandalucia.es/web/gestion/_/content/documentos/presupuestos_publicos
	Mistral 7B instruct	Las áreas priorizadas por las cuentas de la Comunidad Autónoma de Andalucía para el año 2025, según el contexto proporcionado, no se especifican explícitamente. Sin embargo, se puede inferir que las áreas prioritarias se reflejarán en los presupuestos de cada uno de los documentos mencionados: "PRESUPUESTO DE LA COMUNIDAD AUTÓNOMA DE ANDALUCÍA 2025"(31 y 50). Estos documentos no proporcionan información suficiente para identificar las áreas prioritarias con certeza. ¡Gracias por preguntar!
	DeepSeek R1 Distill Llama 8B	Las áreas que priorizan las cuentas de la Comunidad Autónoma de Andalucía para 2025 son la **CAMARA DE CUENTAS** y los **ESTADOS CONSOLIDADOS**. ¡Gracias por preguntar!

10.2. Tablas de preguntas y respuestas *ground truth*

Pregunta	Respuesta
¿Qué ley se menciona en el documento del presupuesto del 2025?	La Ley 12/2023, de 26 de diciembre, del Presupuesto de la Comunidad Autónoma de Andalucía para el año 2024.
¿Cuál es el objetivo del presupuesto de la Comunidad Autónoma de Andalucía para 2024?	El objetivo es la mejora del bienestar en un marco de estabilidad institucional, diálogo y confianza en el potencial de la sociedad andaluza.
¿Cómo ha variado el gasto en infraestructuras educativas entre 2022 y 2023?	Ha aumentado un 15 %, alcanzando 1.260 millones de euros en 2023.
¿Cuál fue el financiamiento para la infraestructura sanitaria en Jaén en el presupuesto de 2023?	En 2023, se asignaron 520.894 euros para el sistema de salud en Jaén, mejorando las instalaciones sanitarias en la provincia.
¿Cuál fue la cantidad asignada al Ayuntamiento de Jaén en el presupuesto de 2022?	Al Ayuntamiento de Jaén se le asignaron 75.787 euros dentro del fondo para la financiación de entidades locales en 2022.
¿Cuál ha sido el presupuesto total aprobado por la Junta de Andalucía en 2023?	El presupuesto total aprobado para 2023 es de 45.619 millones de euros.
¿Cuál ha sido la evolución del presupuesto total de la Junta de Andalucía desde 2020 hasta 2024?	El presupuesto ha mostrado un aumento sostenido debido a la implementación de políticas de mejora de servicios públicos, recuperación económica y financiación europea, incluyendo el Mecanismo de Recuperación y Resiliencia (MRR).
¿Cuál es el total de la dotación de CANAL SUR RADIO Y TELEVISIÓN, S.A. en el presupuesto del 2025?	El importe de la dotación de Canal Sur Radio y Televisión, SA en el presupuesto del año 2025 asciende a 3.700 millones de euros.
¿Cuáles han sido los sectores prioritarios en el presupuesto de Andalucía para 2023?	Los sectores prioritarios para 2023 en Andalucía incluyeron el fortalecimiento de servicios públicos, como sanidad y educación, además de medidas fiscales para reducir la carga tributaria y simplificación administrativa.
¿Qué cantidad destinó el presupuesto de 2017 al apoyo de políticas de empleo en Jaén y otras provincias?	En 2017, se asignaron 434 millones de euros en total para políticas de empleo, incluidas áreas como Emple@Joven, con impacto en Jaén y otras provincias de Andalucía.

¿Qué cantidad fue asignada al Consorcio de Transporte Metropolitano del Área de Málaga en el presupuesto de 2019?	En 2019, el Consorcio de Transporte Metropolitano del Área de Málaga recibió una asignación de 12.698.461 euros.
¿Cómo ha variado la inversión en el sector turístico en Andalucía entre 2021 y 2023?	La inversión en turismo aumentó de 150 millones en 2021 a 210 millones en 2023, enfocados en la modernización y sostenibilidad del sector.
¿Qué cantidad fue destinada al Fondo de Compensación Interterritorial en el presupuesto de Andalucía de 2014?	En 2014, el Fondo de Compensación Interterritorial se redujo en un 55 %, afectando la equidad en los servicios públicos en Andalucía, lo cual fue señalado como perjudicial para la región.
¿Qué medidas contempla el presupuesto de 2023 para la digitalización en Andalucía?	Se incluyen 130 millones para proyectos de transformación digital en diferentes áreas del sector público.
¿Cómo ha variado la inversión en sanidad en Andalucía entre 2020 y 2024?	La inversión en sanidad ha aumentado progresivamente, priorizando el acceso universal y mejorando instalaciones y servicios sanitarios en toda la comunidad.
¿Qué medidas contempla el presupuesto de 2023 para mejorar la infraestructura de justicia?	En 2023, se asignaron 45 millones de euros a la construcción y renovación de juzgados y sedes judiciales en toda Andalucía.
¿Qué cantidad se destina al Plan Nacional de Formación e Inserción Profesional en el presupuesto 2025?	El Plan Nacional de Formación e Inserción Profesional tiene una asignación de 143.981.704 euros.
¿Qué medidas de empleo se impulsaron en 2019 en Andalucía según el presupuesto autonómico?	En 2019, la Junta promovió la creación de empleo a través de una reforma fiscal progresiva y políticas que fomentaban el emprendimiento, eliminando barreras administrativas para atraer inversión y dinamizar la economía regional.
¿Qué papel tuvieron las políticas de empleo en el presupuesto de 2017 en Andalucía?	En 2017, las políticas de empleo incluyeron programas como Emple@Joven y apoyo al trabajo autónomo, con una dotación de 434 millones de euros para el período 2016-2020.
¿Cuál fue la inversión en políticas de salud en Jaén según el presupuesto de 2022?	El presupuesto de 2022 en Jaén incluyó 520.894 euros para mejorar la salud pública y calidad sanitaria en la provincia.

¿Qué partidas están relacionadas con los fondos europeos en 2025?	Los fondos europeos incluyen asignaciones para FEDER, FSE+, FEADER, entre otros, totalizando 2.232.428.067 euros.
¿Cuál es el presupuesto total aprobado para la Comunidad Autónoma de Andalucía en 2025?	El presupuesto total aprobado asciende a 44.521 millones de euros, un incremento del 5 % respecto al año anterior.
¿Cuales son las condiciones para el nombramiento de personal funcionario interino por exceso en el presupuesto del 2020?	Las condiciones son: a) Duración máxima de seis meses dentro de un periodo de doce meses. b) Financiación emitida en un plazo máximo de un mes tras informe de Hacienda c) Personal interino no ocupará plazas fijas
¿Qué objetivos busca alcanzar el presupuesto de 2025 en relación con la igualdad de oportunidades?	El presupuesto busca reducir las desigualdades territoriales, mejorar la igualdad de acceso a servicios públicos y fomentar la inclusión social.
¿Cuáles son las medidas fiscales destacadas en el presupuesto 2025?	Se aplican incentivos fiscales por valor de 800 millones para fomentar inversiones en sostenibilidad y proyectos innovadores.
¿Qué áreas priorizan las cuentas de la Comunidad Autónoma para 2025?	Las cuentas de la Comunidad Autónoma para 2025 priorizan áreas fundamentales como la educación, la salud, el empleo, la sostenibilidad ambiental, la digitalización y la innovación, enmarcándose en los Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas y las estrategias de desarrollo regional de la Unión Europea.
¿Qué porcentaje del presupuesto total se destina a las universidades andaluzas en el presupuesto 2025?	El 12 % del presupuesto total está destinado a las universidades andaluzas, incluyendo investigación y docencia.
¿Cual fue el total de costes de personal de las universidades de titularidad pública en 2017?	El total del costes de personal fueron 529.634.378 euros.
¿Cuál fue el financiamiento para la infraestructura sanitaria en Albacete en el presupuesto de 2023?	Albacete no pertenece a la comunidad autónoma de Andalucía. No dispongo de información relacionada.
¿Cuál fue la inversión en políticas de salud en Madrid según el presupuesto de 2022?	Madrid no pertenece a la comunidad autónoma de Andalucia. No dispongo de información relacionada.

Capítulo 11

APENDICE C

11.1. Videos demostrativos del sistema

Parte del Sistema	**Enlace**
Introducción a la base de datos de conocimiento	Ver vídeo
Funcionamiento del chat	Ver vídeo
Ejecución de evaluación masiva de modelos	Ver vídeo
Construcción de evaluación de modelos	Ver vídeo
Exportación de datos y evaluación	Ver vídeo

Bibliografía

[1] Nancy Piedad Molina Montoya. *El estado del arte como estrategia metodológica: Fundamentos y aplicaciones*. Editorial Universidad del Valle, 2014. ISBN 978-958-765-924-8.

[2] José Ramón Martínez and Carlos Pérez. *Aprendizaje Automático: Fundamentos y aplicaciones*. Alfaomega, 2020. ISBN 978-607-538-494-7.

[3] Roberto Hernández Sampieri, Carlos Fernández Collado, and Pilar Baptista Lucio. *Metodología de la investigación*. McGraw Hill, 2014. ISBN 9786071510142.

[4] Yoav Goldberg. *Neural Network Methods for Natural Language Processing*. Morgan & Claypool, 2017. ISBN 9781627052986.

[5] Ana M. Gutiérrez López and Luis F. Pérez Gómez. Avances en modelos de lenguaje natural: Análisis de bert y gpt. *Revista Iberoamericana de Inteligencia Artificial*, 22(64):45–60, 2019.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[7] David López García and Marta Sánchez Rivera. Aplicaciones de los modelos de lenguaje en la industria: Un enfoque práctico. In *Actas del Congreso Español de Procesamiento del Lenguaje Natural*, pages 120–130. SEPLN, 2021.

[8] Wikipedia contributors. Maximum inner-product search, 2025. URL https://en.wikipedia.org/wiki/Maximum_inner-product_search. Accedido el 30 de enero de 2025.

[9] Omid Keivani, Kaushik Sinha, and Parikshit Ram. Approximate nearest neighbor search using inverted indices. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3566–3573, 2017.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2015.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[12] Tomás López Solaz, José Antonio Troyano Jiménez, Francisco Javier Ortega Rodríguez, and Fernando Enríquez de Salamanca Ros. Una aproximación al uso de word embeddings en una tarea de similitud de textos en español. *Procesamiento del Lenguaje Natural*, 57:67–74, 2016. URL https://rua.ua.es/dspace/handle/10045/57753.

[13] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[14] Pinecone Systems Inc. Pinecone: The vector database for machine learning. https://www.pinecone.io/.

[15] Chroma. Chroma: The ai-native open-source embedding database. https://www.trychroma.com/.

[16] Weaviate. Weaviate: An open-source vector search engine. https://weaviate.io/.

[17] Jianguo Wang, Rentong Guo, Xiaofan Luan, Long Xiang, and Xiao Yan. Milvus: A purpose-built vector data management system. *arXiv preprint arXiv:2201.01562*, 2021.

[18] Qdrant. Qdrant: Vector search engine and database. https://qdrant.tech/.

[19] Spotify. Annoy: Approximate nearest neighbors oh yeah. https://github.com/spotify/annoy.

[20] Redis. Redis-vector: Vector similarity search in redis. https://redis.io/docs/stack/search/reference/vectors/.

[21] OpenSearch. k-nn search in opensearch. https://opensearch.org/docs/latest/search-plugins/knn/.

[22] Vespa. Vespa: The open big data serving engine. https://vespa.ai/.

[23] Ian Sommerville. *Software Engineering*. Pearson, 10th edition, 2020. ISBN 978-0133943030.

[24] ClickUp. Análisis de requisitos: Pasos clave y técnicas para hacerlo bien, 2023. URL https://clickup.com/es-ES/blog/138879/analisis-de-requisitos.

[25] Jorge Humberto Miranda Realpe. La gestión de requerimientos en la ingeniería web. *SATHIRI*, (10):135, 2016. doi: 10.32645/13906925.180. URL http://dx.doi.org/10.32645/13906925.180.

[26] Ivar Jacobson. *Casos de Uso 2.0: La Guía Definitiva*. Ivar Jacobson International, 2013. URL https://www.ivarjacobson.com/files/field_iji_file/article/use_case_2.0_-_spanish_translation.pdf. Accedido: 2025-02-03.

[27] Grady Booch, James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 2005. ISBN 978-0321267979.

[28] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 3rd edition, 2004. ISBN 978-0321193681.

[29] Udit Kamath, Samiksha Swarup, and Vishal Bhatnagar. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*, 2019. URL https://arxiv.org/abs/1901.09069.

[30] Curso NLP en Español. Embeddings: Word2vec, glove y fasttext | clase 15. YouTube, 2023. URL https://www.youtube.com/watch?v=xu3FC81eNKI. Último acceso: 2024-02-05.

[31] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Addison-Wesley, 1994. ISBN 978-0-201-63361-0.

[32] Guido van Rossum. Python: A high-level general-purpose programming language. 1991. URL https://www.python.org/doc/essays/blurb/.

[33] The evolution of python 3.0, 2008. URL https://docs.python.org/3/whatsnew/3.0.html.

[34] Python software foundation, 2023. URL https://www.python.org/.

[35] Wes McKinney. Python for data analysis. 2017.

[36] Streamlit documentation, 2023. URL https://docs.streamlit.io.

[37] Snowflake acquires streamlit to advance data science applications, 2022. URL https://www.snowflake.com.

[38] Flask documentation, 2023. URL https://flask.palletsprojects.com.

[39] Harrison Chase. Langchain: A framework for llm applications. 2022. URL https://www.langchain.com/.

[40] Hugging face: Democratizing nlp. 2022. URL https://huggingface.co/.

[41] Thomas et al. Wolf. Transformers: State-of-the-art natural language processing. *ACL*, 2020.

[42] Chromadb: Next-generation vector database. 2023. URL https://chromadb.com/.

[43] Python threading module. 1999. URL https://docs.python.org/3/library/threading.html.

[44] Tim Dettmers. Bitsandbytes: Optimizing memory usage for ai models. 2022. URL https://github.com/TimDettmers/bitsandbytes.

[45] Roman Mogylatov. Dependency injector: A dependency injection framework for python. 2016. URL https://github.com/ets-labs/python-dependency-injector.

[46] Armin Ronacher. Click: A python package for creating cli applications. 2014. URL https://click.palletsprojects.com.

[47] Fabien Potencier. *Symfony 7.2*. SensioLabs, 2025. URL https://symfony.com/doc.

[48] Symfony console: Building cli applications in php, 2023. URL https://symfony.com/doc/current/components/console.html.

[49] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318. Association for Computational Linguistics, 2002. doi: 10.3115/1073083.1073135.

[50] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL Workshop*, pages 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

[51] Pypdfloader: Extracting text from pdfs for nlp applications, 2023. URL https://github.com/hwchase17/langchain/blob/master/langchain/document_loaders/pdf.py.

[52] Pypdfloader documentation, 2023. URL https://pypdf.readthedocs.io/en/latest/.

[53] Processing pdfs for ai and machine learning applications, 2023. URL https://towardsdatascience.com/extracting-text-from-pdfs-for-nlp-4a47d5a96e91.

[54] Richard E. Turner. An introduction to transformers, 2023. URL https://arxiv.org/abs/2304.10557.

[55] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2023.

[56] Namrata Patel and Kartik Srinivasan. Ragas: Benchmarking retrieval-augmented generation applications. *Journal of AI Research*, 65:101–120, 2023.

[57] Alber. Guía completa de los principios solid, 2024. URL https://cosasdedevs.com/posts/guia-completa-principios-solid/. Accedido: 2025-02-11.

[58] Carlos Macías Martín. Principios solid con ejemplos, 2019. URL https://www.enmilocalfunciona.io/principios-solid/. Accedido: 2025-02-11.

[59] Arjun Raman and Pankaj Kumar. Evaluating retrieval-augmented generation: Challenges and metrics. *ArXiv preprint arXiv:2403.01234*, 2024.

[60] Barcelona Supercomputing Center. Alia: La primera infraestructura pública, abierta y multilingüe de ia en europa, 2025. URL https://alia.gob.es/.

[61] Barcelona Supercomputing Center - Language Technologies. Modelos bsc-lt, 2025. URL https://huggingface.co/BSC-LT.

[62] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

[63] Gang Li, Jianzhuang Zhang, Xiaogang Zhang, and Yan Li. Simvit: Exploring a simple vision transformer with sliding windows. *arXiv preprint arXiv:2112.04691*, 2021.

[64] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *ArXiv preprint arXiv:1503.02531*, 2015.

[65] DeepSeek AI Research Team. Deepseek r1 distillation: Optimizing large language models for efficiency and performance. *ArXiv preprint arXiv:2406.01567*, 2024.

[66] DeepSeek AI Research Team. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2024. URL https://arxiv.org/abs/2501.12948.

[67] Wei Zhang, Jinhui Li, and Tao Wang. Deepseek qwen: A scalable and efficient approach for language model distillation. *NeurIPS Workshop on Efficient AI Models*, 2024.

[68] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[69] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv preprint arXiv:2205.14135*, 2022.