Integración de la Arquitectura de Generación Aumentada con Recuperación para Guías Clínicas: Desarrollo de un Chatbot de Apoyo para Reumatólogos

Daniel Camacho Montaño

Machine Learning

Màster en Bioestadística y Bioinformática

Nombre del director/a de TF:
Alfredo Madrid García

Nombre del/de la PRA:
Agnès Pérez Millan

4 de Junio de 2025

Universitat Oberta
de Catalunya

Ficha Del Trabajo Final

Título del trabajo:	Integración de la Arquitectura de Generación Aumentada con Recuperación para Guías Clínicas: Desarrollo de un Chatbot de Apoyo para Reumatólogos
Nombre del autor/a:	Daniel Camacho Montaño
Nombre del director/a de TF:	Alfredo Madrid García
Nombre del/de la PRA:	Agnès Pérez Millan
Fecha de entrega:	4 de Junio de 2025
Titulación o programa:	Màster en Bioestadística y Bioinformática
Área del trabajo final:	Machine Learning
Idioma del trabajo:	Castellano
Palabras clave:	Inteligencia artificial, chatbot, guías clínicas, reumatología, sistema de ayuda a la decisión

Resumen del trabajo

Este Trabajo de Fin de Máster presenta una evaluación comparativa del rendimiento de la arquitectura Retrieval-Augmented Generation (RAG) aplicada a guías clínicas en reumatología. Con el objetivo de analizar cómo varía la calidad de las respuestas generadas en función del volumen de información recuperada y del modelo de lenguaje utilizado, se implementaron tres variantes: un modelo base puramente generativo (Generative), un modelo RAG con fragmentos de 500 caracteres (RAG500) y otro con fragmentos de 1000 caracteres (RAG1000). Para cada configuración, se emplearon tres LLM: GPT-3.5 Turbo, GPT-4o mini y GPT-4.1 mini.

El corpus documental se construyó a partir de guías clínicas del American College of Rheumatology, procesadas mediante técnicas de extracción de texto (GROBID y LlamaParse), segmentación (RecursiveCharacterTextSplitter) y vectorización semántica (text-embedding-3-large), integrando herramientas como LangChain y LangSmith. Las respuestas fueron evaluadas con seis métricas: fidelidad, relevancia, precisión, concisión, completitud parcial y total, usando 340 preguntas clínicas generadas por un modelo externo (Gemini 1.5 Pro).

Los resultados muestran mejoras significativas de RAG frente al modelo generativo. En el tercer experimento, RAG500 y RAG1000 fueron claramente preferidos ($p < 0.001$). Además, RAG1000 superó a RAG500 en fidelidad ($p = 0.001$), relevancia ($p = 0.0013$), precisión ($p = 0.0011$) y completitud ($p < 0.001$), aunque con ligera pérdida de concisión. Se aplicaron pruebas de Wilcoxon y binomial para validar estas diferencias.

El estudio aporta evidencia sobre la eficacia de RAG para mejorar la generación de información clínica estructurada, destacando su potencial como sistema de apoyo a la toma de decisiones.

Abstract

This Master's Thesis presents a comparative evaluation of the performance of the Retrieval-Augmented Generation (RAG) architecture applied to clinical guidelines in rheumatology. Aimed at analyzing how the quality of generated responses varies depending on the volume of retrieved information and the language model used, three variants were implemented: a purely generative baseline model (Generative), a RAG model with 500-character chunks (RAG500), and another with 1000-character chunks (RAG1000). For each configuration, three LLMs were employed: GPT-3.5 Turbo, GPT-4o mini, and GPT-4.1 mini.

The document corpus was built from clinical guidelines by the American College of Rheumatology, processed through text extraction techniques (GROBID and LlamaParse), segmentation (RecursiveCharacterTextSplitter), and semantic vectorization (text-embedding-3-large), integrating tools such as LangChain and LangSmith. Responses were evaluated using six metrics: fidelity, relevance, accuracy, conciseness, partial completeness, and full completeness, based on 340 clinical questions generated by an external model (Gemini 1.5 Pro).

The results show significant improvements of RAG over the generative model. In the third experiment, RAG500 and RAG1000 were clearly preferred (p ¡ 0.001). Moreover, RAG1000 outperformed RAG500 in fidelity (p = 0.001), relevance (p = 0.0013), accuracy (p = 0.0011), and completeness (p ¡ 0.001), though with a slight loss in conciseness. Wilcoxon and binomial tests were applied to validate these differences.

This study provides evidence of RAG's effectiveness in enhancing the generation of structured clinical information, highlighting its potential as a decision-support system.

Índice general

Índice de figuras

Índice de cuadros

Siglas

AASLD American Association for the Study of Liver Diseases

ACR American College of Rheumatology

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

CCEG Competencia en Compromiso Ético y Global

EASL European Association for the Study of the Liver

ERC Enfermedades Renales Crónicas

ERM Enfermedades Reumáticas y Musculoesqueléticas

EULAR European Alliance of Associations for Rheumatology

FAISS Facebook AI Similarity Search

GDPR General Data Protection Regulation

GPT Generative Pre-trained Transformer

GPU Graphics Processing Unit

GROBID GeneRation Of BIbliographic Data

HIPAA Health Insurance Portability and Accountability Act

HTML HyperText Markup Language

IA Inteligencia Artificial

JSON JavaScript Object Notation

LlaMA Large Language Model Meta AI

LLM Large Language Model

MAGDA Multi-Agent Guideline-Driven Diagnostic Assistant

ML Machine Learning

MTEB Massive Text Embedding Benchmark

NDCG Normalized Discounted Cumulative Gain

NLM Neural Language Models

NLP Natural Language Processing

ODS Objetivos de Desarrollo Sostenible

PDF Portable Document Format

PLM Pre-trained Language Models

RAG Retrieval-Augmented Generation

RLHF Reinforcement Learning from Human Feedback

SFT Supervised Fine-Tuning

SLM Statistical Language Model

STS Semantic Textual Similarity

TEI Text Encoding Initiative

XML Extensible Markup Language

Capítulo 1

Introducción

1.1. Contexto y justificación del trabajo

Las **Enfermedades Reumáticas y Musculoesqueléticas (ERM)** engloban un amplio espectro de patologías crónicas que afectan principalmente al aparato locomotor, aunque también pueden comprometer otros órganos y sistemas, afectando comúnmente las articulaciones de individuos de todas las edades y géneros, pero también pueden afectar los músculos, órganos internos y otros tejidos. La elevada prevalencia, el dolor y las complicaciones asociadas las sitúan entre las principales causas de deterioro en la calidad de vida a nivel global. La complejidad clínica, sumada a la variabilidad en las manifestaciones y en las respuestas al tratamiento, subraya la necesidad de contar con herramientas avanzadas. que faciliten la toma de decisiones, teniendo en cuenta la última evidencia disponible de las mayores asociaciones, European Alliance of Associations for Rheumatology (EULAR), y American College of Rheumatology (ACR)

En este contexto, la Inteligencia Artificial (IA) y los **Modelos de Lenguaje de Gran Tamaño** (Large Language Model (LLM)) han revolucionado la forma en que se procesa y genera información a partir de datos textuales complejos. Herramientas comerciales como ChatGPT han demostrado una notable capacidad para comprender y generar texto de manera coherente y versátil, facilitando su adopción en prácticamente todos los sectores. Esta revolución responde a la creciente necesidad de sistemas eficientes capaces de procesar grandes volúmenes de información y proporcionar respuestas rápidas, optimizando el flujo de trabajo y mejorando la toma de decisiones en diversas disciplinas.

El reciente avance de los LLM ha impulsado el desarrollo de soluciones inteligentes para la generación y recuperación de información en múltiples áreas. En el ámbito médico, donde el acceso a información precisa, actualizada y basada en evidencia es crucial, estas herramientas prometen transformar la toma de decisiones clínicas, la educación médica y la investigación. Sin embargo, los modelos fundacionales presentan limitaciones importantes, como la tendencia a generar respuestas incompletas o **alucinaciones**, lo que puede comprometer la precisión en entornos donde la fiabilidad es fundamental (1). Además, es importante considerar que los LLM fundacionales adquieren su conocimiento mediante un preentrenamiento computacionalmente costoso que, si bien es eficaz en tareas generales, no permite incorporar información nueva y actualizada sin llevar a cabo un nuevo entrenamiento, lo que hace que su conocimiento permanezca estático.

Para abordar estas limitaciones, la técnica ***Retrieval-Augmented Generation*** **(RAG)** combina la generación del lenguaje con la recuperación de documentos relevantes, usados como fuentes externas de conocimiento. Esta técnica permite al modelo acceder a una base de conocimiento externa al entrenamiento, como las guías de prácticas clínicas oficiales y verificadas, asegurando que las respuestas generadas sean más precisas y confiables, reduciendo las alucinaciones y proporcionando una respuesta más fundamentada.

Las guías de práctica clínica, a su vez, son documentos normativos elaborados por instituciones de salud y sociedades científicas que contienen las últimas recomendaciones e indicaciones basadas en una evidencia demostrada. Por lo tanto, integrar estas fuentes de información en una arquitectura RAG permitiría mejorar la calidad de las respuestas en aplicaciones como asistentes médicos virtuales, permitiendo dar apoyo en decisiones clínicas (2; 3).

El uso de RAG en guías médicas permitiría resolver necesidades críticas en el ámbito médico, ya que se reducirían los errores en la generación de texto por alucinaciones al integrar un sistema de recuperación de documentos (4). También permitiría a los profesionales de la salud e investigadores un acceso rápido y contextualizado a información basada en evidencia y referenciada propiciando una mejora en la toma de decisiones clínicas. Así, el uso de esta tecnología tiene el potencial de facilitar una práctica médica más dinámica y actualizada, en la que los profesionales pueden disponer de recomendaciones respaldadas por evidencia sin dependere de su propia capacidad de actualización constante.

1.2. Objetivos del trabajo

El objetivo principal de este trabajo de fin de máster es la creación de un sistema RAG que incluya guías de práctica clínica de reumatología, mejorando el cuidado asistencial.

Para alcanzar este objetivo general, se plantean los siguientes objetivos:

1.2.1. Objetivos Primarios

1. **Integración de guías clínicas de reumatología en un modelo de lenguaje basado en una arquitectura RAG**

2. **Evaluación del rendimiento del sistema RAG en comparación con un modelo de lenguaje fundacional (LLM)**: análisis comparativo de la precisión, eficiencia y calidad de las respuestas generadas por el modelo RAG frente a un modelo de lenguaje fundacional.

3. **Desarrollo de una interfaz visual**: implementación de una interfaz gráfica intuitiva que integre el modelo RAG optimizado, proporcionando a profesionales médicos y pacientes una herramienta automatizada para el diagnóstico y la consulta de guías médicas. Esta interfaz permitirá respuestas rápidas y orientadas al usuario, mejorando el acceso a la información y reduciendo el tiempo de acción.

1.2.2. Objetivos Secundarios

1.1 **Objetivos secundarios asociados al objetivo 1**

1.1 **Identificación y preparación de documentos clínicos sobre patologías reumáticas:** localización de fuentes médicas relevantes, recopilación de documentos Portable Document Format (PDF) y aplicación de técnicas de extracción y limpieza del texto para el spoterior uso en una arquitectura RAG.

1.2 **Evaluación de técnicas de procesamiento de texto**: análisis y selección de métodos de parseo, segmentación (chunking) y bases de datos vectoriales para el tratamiento de los documentos de las guías clínicas.

2.2 **Objetivos secundarios asociados al objetivo 2**

2.1 **Revisión del estado del arte de RAG en el ámbito médico**: revisión y análisis de las últimas implementaciones de modelos RAG aplicados a guías médicas, evaluando sus capacidades y limitaciones en este contexto.

2.2 **Comparación de diferentes arquitecturas RAG**: evaluación de modelos actuales en términos de métricas de rendimiento, precisión, coherencia y capacidad de adaptación en la generación de texto.

3.3 **Objetivos secundarios asociados al objetivo 3**

3.1 **Diseño y desarrollo de la interfaz gráfica:** creación de una interfaz intuitiva y amigable que permita la interacción con el modelo RAG optimizado.

3.2 **Integración de funcionalidades de consulta**: implementación de mecanismos que permitan realizar consultas, recibir respuestas, así como obtener las fuentes documentales utilizadas.

3.3 **Pruebas de usabilidad y validación**: realización de pruebas como usuario para evaluar la experiencia, rapidez y eficacia en el acceso a la información proporcionada por la interfaz.

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

Esta sección analiza los posibles impactos, tanto positivos como negativos, derivados de esta tesis, considerando las tres dimensiones establecidas por la Competencia en Compromiso Ético y Global (CCEG): sostenibilidad, dimensión socioética y diversidad. El análisis se enmarca dentro de los Objetivos de Desarrollo Sostenible (ODS), garantizando un enfoque basado en la responsabilidad y la ética académica.

1.3.1. Sostenibilidad y medio ambiente

La creciente revolución de la IA ha transformado la industria médica, ofreciendo mejoras significativas en diferentes áreas, como la precisión diagnóstica, la eficiencia y los resultados de los pacientes (5). Sin embargo, estas tecnologías requieren de un gran consumo de energía, y, consecuentemente, una alta producción de CO2 ambiental, llegando, en ocasiones, superar el consumo ahorrado (5). De hecho, un estudio reciente concluyó que para el entrenamiento de

un solo modelo de IA, se generaron cerca de 285.000kg de CO2, siendo una producción mayor que la de 5 automóviles en toda su vida útil (6).

Además, el funcionamiento de los modelos requieren de agua para enfriar los centros de datos y generar electricidad para proveerlos de energía, aumentando drásticamente el consumo de agua en los últimos años. Se plantea que, para 2027, la demanda de agua relacionada al uso de IA será de 6.600 millones de metros cúbicos.(6). Durante el entrenamiento de uno de los modelos Generative Pre-trained Transformer (GPT), concretamente GPT-3, se estima que se consumieron alrededor de 700.000 litros de agua dulce, lo que representa una contribución significativa a la huella hídrica asociada al uso de la inteligencia artificial (7).

Asimismo, la creciente demanda unidades de procesamiento gráfico (Graphics Processing Unit (GPU)) conlleva un impacto ambiental significativo, ya que su producción requiere la extracción de metales como el litio, el cobalto y el níquel (8). Estos materiales son esenciales para la fabricación de otros componentes electrónicos avanzados, pero su obtención implica procesos de minería intensiva que pueden provocar desforestación, contaminación y pérdida de biodiversidad.

Por otro lado, los modelos RAG presentan una alternativa más sostenible, ya que no requieren un reentrenamiento constante para mantenerse actualizados. Los modelos actuales requieren un entrenamiento costoso y computacionalmente intensivo, lo que dificulta su actualización frecuente. En cambio, RAG permite acceder a informaciónactualizada sin la necesidad de desarrollar y entrenar nuevos modelos específicos de LLM, lo que se traduce en una reducción significativa del impacto ambiental asociado a estos procesos. Al disminuir la demanda de recursos computacionales, como las GPU, también se reduce la necesidad de extraer materiales críticos como el litio, el cobalto y el níquel, mitigando así las graves consecuencias ambientales derivadas de la minería intensiva, como la deforestación, la contaminación y la pérdida de biodiversidad.

1.3.2. Impacto socio-ético de los modelos RAG

Actualmente, el uso de la IA en el ámbito de la salud se centra en la obtención de resultados, mientras que su aplicación en el diagnóstico y las consultas médicas basadas en guías médicas está en desarrollo. De hecho, el uso de modelos LLM y los asistentes virtuales no especializados presentan una baja precisión, ya que no siempre se basan en documentos verificados (9).

Sin embargo, la implementación de recuperación en la generación de respuestas permitiría el acceso a información actualizada en tiempo real. Además, al proporcionar la visualización de las fuentes de utilizadas mejorarían la transparencia en la toma de decisiones médicas y reducirían el riesgo de desinformación (10) y (11).

Finalmente, los asistentes virtuales basados en IA requieren de una infraestructura potente, no siempre disponible en todos los hospitales, especialmente en regiones con menos recursos. En este sentido, aunque un modelo RAG depende de un LLM, este puede ser alojado en la nube, mientras que la base de datos vectorial y los sistemas de recuperación de información pueden mantenerse en servidores locales. Esta arquitectura híbrida facilita el despliegue en áreas rurales donde el acceso a médicos especializados es limitado, permitiendo la provisión de diagnósticos precisos y oportunos sin necesidad de infraestructura computacional avanzada en el sitio. De esta manera, contribuiría a reducir la brecha digital existente, permitiendo que centros de salud con menos recursos también se beneficien de los avances en inteligencia artificial médica (12).

Además, un modelo RAG con memoria podría facilitar el acceso a la IA por parte de usuarios con distintos niveles de conocimiento, adaptándose a sus necesidades específicas. Esto sería posible mediante el uso de técnicas de *prompt engineering*, ajustando la formulación de las respuestas en función del contexto y la situación particular del usuario. De esta manera, se democratizaría su uso más allá del ámbito profesional, haciéndolo accesible también a pacientes y cuidadores.

1.3.3. Diversidad y derechos humanos

En el trabajo, se van a usar guías clínicas elaboradas por comités científicos y sociedades científicas, caracterizadas por su rigor y un gran sentido de la diversidad, caracterizadas por su rigor, en las cuales se presupone que el panel de elaboración es diverso. Por lo tanto, el uso de estas guías mediante RAG puede contribuir a reducir el sesgo, asegurando que la información proporcionada esté basada en el consenso de expertos y representando adecuadamente a distintos grupos y contextos clínicos (13).

Este enfoque facilita el acceso equitativo a información médica confiable, mejorando la atención en comunidades marginadas. Además, al adaptar sus recomendaciones en función del usuario, el sistema permite que información médica compleja sea comprensible para cualquier persona (13).

La arquitectura RAG no manejará datos confidenciales de pacientes ni historiales clínicos reales, limitándose a utilizar información procedente de fuentes validadas y guías clínicas públicas, y su implementación se ajusta plenamente a las normativas vigentes en materia de protección de datos, como el General Data Protection Regulation (GDPR) o la Health Insurance Portability and Accountability Act (HIPAA), garantizando la privacidad y seguridad de la información (14; 15).

1.4. Enfoque y metodología seguida

Los pasos realizados para establecer el modelo RAG fueron los siguientes:

1. **Estudio de la literatura:** Se iniciará con un estudio del estado del arte sobre LLM y los modelos RAG desarrollados en el ámbito médico.

2. **Preparación del dataset:** Se obtendrán las guías médicas en formato PDF, se extraerá la información (parseo), se fragmentarán (chunking), se realizará una transformación vectorial (embedding) y se creará una base de datos vectorial.

3. **Selección del modelo:** Se estudiarán los diferentes modelos de LLM como generadores de respuesta y los diferentes *retrievers* disponibles comercialmente. Se evaluará su capacidad de recuperar información a partir del vectorstore, la precisión y rendimiento.

4. **Diseño de la arquitectura RAG:** Con los modelos seleccionados, se formulará la arquitectura RAG adecuada para el proceso de los datos. El proceso se enfocará en mejorar las respuestas de LLM fundacionales.

5. **Evaluación del modelo:** Se estudiará el rendimiento del modelo mediante métricas de precisión, solicitando a un LLM externo al sistema para que determine el rendimiento de los modelos. Se solicitará la evaluación de métricas, así como se determinará si existe una superioridad de la respuesta proporcionada por un modelo sobre el resto, calculando si existe una mayor calidad general en las respuestas generadas por un modelo sobre otro.

6. **Interfaz gráfica web:** La arquitectura finalmente diseñada se integrará a una interfaz web (local) que permitirá la interacción usuario-asistente virtual, haciendo uso de gradio.

1.5. Planificación del trabajo

Se han tenido en cuenta las fechas de entrega de las PEC para la distribución de trabajo en cuatro bloques. En la figura Figura 1.1 se muestra la distribución del tiempo de las tareas a realizar en cada bloque, descritas a continuación:

1.5.1. Distribución de tareas

Para lograr los objetivos de este trabajo final de máster, se definieron las siguientes tareas:

1. **Investigación y revisión de literatura (3 semanas)**

   El objetivo es realizar una profunda revisión de la literatura, enfocada en los modelos LLM, la arquitectura RAG y sus aplicaciones previas en consultas médicas y guías clínicas.

   - Identificación de los modelos LLM candidatos y sus aplicaciones previas.
   - Revisión de la arquitectura RAG, sus componentes y sus aplicaciones.
   - Resumen de los hallazgos de la investigación.

2. **Recopilación de documentos y procesamiento (2 semanas)**

   El objetivo es recopilar guías clínicas en formato PDF y procesarlas para posteriormente crear una base de datos vectorial a partir de la cual se recupere contenido relevante para fundamentar la respuesta del LLM. Se estudiarán los diferentes métodos de parseo disponibles y se realizarán pruebas para determinar el más eficiente para la conversión del formato PDF en texto plano

   Debido a la complejidad de la transformación de imágenes en texto plano, solo se vectorizará el contenido textual, dejando de lado las figuras. Sin embargo, debido a la importancia de la información contenida en las tablas, sí que se incluirá dicha información, convirtiendo a formato tabular para evaluar si esto mejora la precisión del modelo.

   - Data collection: Recopilación de guías médicas verificadas
   - Data processing: Identificación, aplicación y evaluación de los métodos de parseo, segmentación (*chunking*) y vectorización.

3. **Selección del modelo y análisis preliminar (2 semanas)**

   Se evaluarán diferentes arquitecturas RAG, combinando diversos métodos de recuperación (*retrievers*) y generación (*generators*) para seleccionar la mejor configuración para la arquitectura RAG del trabajo final de máster.

   - Realización de pruebas iniciales para evaluar la idoneidad de cada modelo según los datos disponibles.
   - Comparación de los modelos basados en métricas de rendimiento.
   - Documentación del proceso de selección y justificar la elección del modelo final.

4. **Arquitectura RAG (3 semanas)**

   Se implementará la arquitectura seleccionada para procesar la información del vectorstore y generar respuestas relevantes basadas en las guías médicas. Se ajustarán los parámetros para tratar de optimizar la relevancia, configurando las estrategias de indexación y búsqueda, así como los parámetros iniciales del modelo.

   - Selección del modelo base para la recuperación de información y la generación de respuestas.
   - Indexación de los documentos utilizando técnicas de *embedding*.
   - Especialización del modelo con datos médicos específicos para maximizar la relevancia clínica.

5. **Evaluación del modelo y optimización (3 semanas)**

   Se analizarán métricas de rendimiento, evaluando la precisión, concisión, relevancia, fidelidad y completitud. Se realizarán análisis del efecto del preprocesamiento, rendimiento del LLM e incremento de la potencia estadística. Se optimizará el modelo ajustando los componentes del sistema necesarios.

   - Generación de las preguntas (10 por patología) a partir de las guías clínicas.
   - Generación de las respuestas resultantes de cada modelo (Generative, RAG500 y RAG1000).
   - Evaluación de las métricas de rendimiento de cada modelo mediante el LLM evaluador.
   - Comparación de los modelos 2 a 2, evaluando la calidad de las respuestas.
   - Análisis del rendimiento del modelo y aplicar modificaciones.
   - Análisis complementarios tras modificar el sistema, evaluando las capacidades del LLM y la potencia estadística.

6. **Creación de una interfaz web basada en el mejor modelo (2 semanas)**

   Se diseñará una interfaz gráfica web con herramientas de prototipado de Gradio que integre el mejor modelo de arquitectura RAG, permitiendo la interacción del usuario con el sistema.

- Diseño de una interfaz web.
- Integración del modelo dentro de la interfaz.
- Evaluación de la funcionalidad de la interfaz.

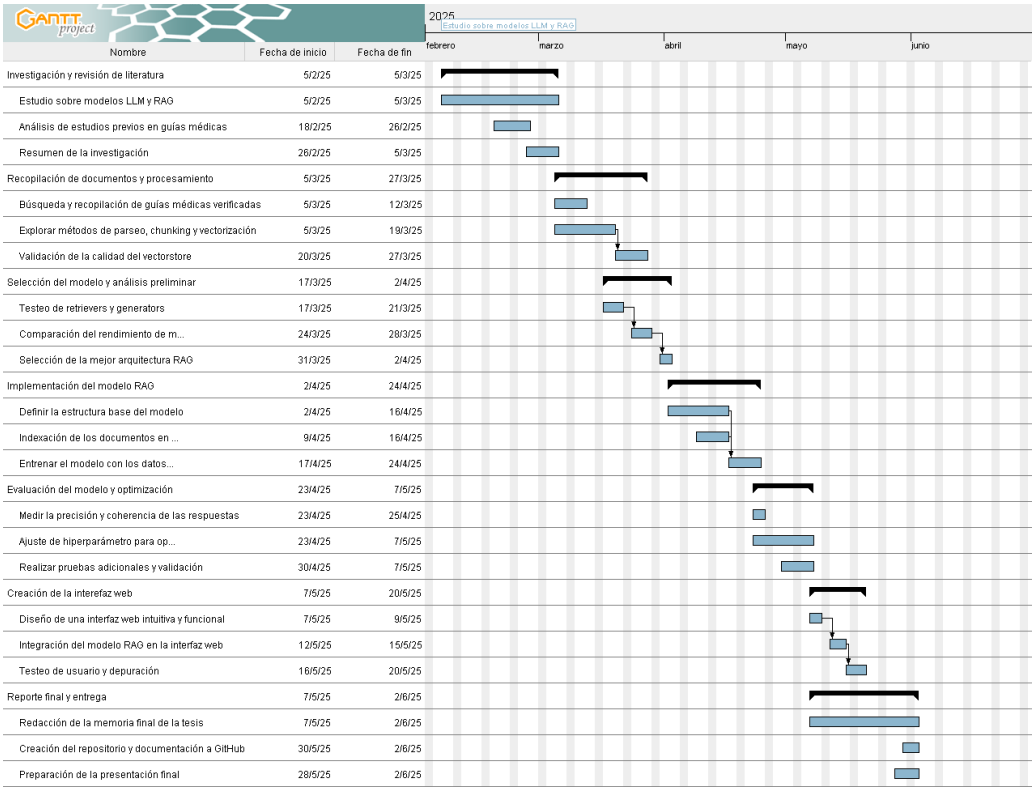
7. **Reporte final (3 semanas)**

   En esta fase se redactará la memoria final del trabajo final de máster, se diseñará y grabará la presentación, y se creará un repositorio en GitHub con el código empleado.

   - Redacción de la memoria final.
   - Creación del repositorio en GitHub.
   - Preparación y grabación de la presentación.
   - Realización de la entrega final de la tesis.

1.5.2. Diagrama de Gantt

A continuación, la Figura 1.1 presenta un Diagrama de Gantt, en el que se plantea la planificación semanal del trabajo



Nombre	Fecha de inicio	Fecha de fin
Investigación y revisión de literatura	5/2/25	5/3/25
Estudio sobre modelos LLM y RAG	5/2/25	5/3/25
Análisis de estudios previos en guías médicas	18/2/25	26/2/25
Resumen de la investigación	26/2/25	5/3/25
Recopilación de documentos y procesamiento	5/3/25	27/3/25
Búsqueda y recopilación de guías médicas verificadas	5/3/25	12/3/25
Explorar métodos de parseo, chunking y vectorización	5/3/25	19/3/25
Validación de la calidad del vectorstore	20/3/25	27/3/25
Selección del modelo y análisis preliminar	17/3/25	2/4/25
Testeo de retrievers y generators	17/3/25	21/3/25
Comparación del rendimiento de m...	24/3/25	28/3/25
Selección de la mejor arquitectura RAG	31/3/25	2/4/25
Implementación del modelo RAG	2/4/25	24/4/25
Definir la estructura base del modelo	2/4/25	16/4/25
Indexación de los documentos en ...	9/4/25	16/4/25
Entrenar el modelo con los datos...	17/4/25	24/4/25
Evaluación del modelo y optimización	23/4/25	7/5/25
Medir la precisión y coherencia de las respuestas	23/4/25	25/4/25
Ajuste de hiperparámetro para op...	23/4/25	7/5/25
Realizar pruebas adicionales y validación	30/4/25	7/5/25
Creación de la interefaz web	7/5/25	20/5/25
Diseño de una interfaz web intuitiva y funcional	7/5/25	9/5/25
Integración del modelo RAG en la interfaz web	12/5/25	15/5/25
Testeo de usuario y depuración	16/5/25	20/5/25
Reporte final y entrega	7/5/25	2/6/25
Redacción de la memoria final de la tesis	7/5/25	2/6/25
Creación del repositorio y documentación a GitHub	30/5/25	2/6/25
Preparación de la presentación final	28/5/25	2/6/25

Figura 1.1: Planificación temporal del trabajo final de máster.
El diagrama muestra la planificación semanal de las tareas del TFM, desde la revisión bibliográfica inicial hasta la entrega de la memoria y la presentación. Se determinan tres fases principales: recopilación de datos, desarrollo del sistema RAG y evaluación y análisis.

1.5.3. Análisis de riesgos

Antes de la elaboración del proyecto, se han contemplado los posibles riesgos que pueden surgir y causar un impacto negativo en la elaboración del trabajo, determinando el nivel de riesgo y las posibles alternativas para mitigarlos.

1. **Problemas en la creación del repositorio**

   Impacto: Alto

   El proceso de parseo es esencial para implementar un modelo RAG, ya que garantiza la extracción y estructuración óptima del texto de los documentos PDF originales. Sin embargo, este paso puede presentar problemas como la presencia de contenido visual (imágenes), formato desorganizado y pérdida de información.

   Para mitigar estos riesgos, se establecen herramientas de parseo alternativas y paralelas para estudiar las diferencias y escoger la herramienta más adecuada. También se han seleccionado guías médicas con la estructura lo más similar y simple posible para evitar diferencias entre documentos que alteren el proceso.

2. **Diversidad entre guías médicas**

   Impacto: Medio

   Las guías médicas no fueron diseñadas específicamente para ser procesadas por modelos de análisis de datos, sino que su propósito es servir como referencia y apoyo para profesionales de la salud. Aunque muchas están reconocidas por la ACR, cada una presenta una estructura particular, lo que dificulta su tratamiento automatizado.

   Por ello, se deben considerar las posibles limitaciones al utilizarlas en un entorno computacional. Además, cabe destacar que algunas guías clínicas pueden estar protegidas por derechos de autor, por lo que su uso en este trabajo se ha realizado exclusivamente con fines académicos, sin ningún propósito comercial.

3. **Dificultades en la vectorización**

   Impacto: Medio

   Los modelos RAG recuperan la información basándose en la proximidad vectorial, por lo que la correcta vectorización de la información impacta directamente en la calidad de la recuperación de documentos. La conversión del contenido de los documentos PDF a un formato estructurado es fundamental para una representación adecuada de los datos. Una transformación deficiente puede inducir errores en la interpretación del texto.

   Para optimizar este proceso, se realiza una segmentación eficiente de los documentos en fragmentos de tamaño óptimo (chunks) para preservar el contexto en la recuperación de información. Asimismo, se evalúan diferentes enfoques de vectorización en paralelo para determinar la metodología más eficaz.

4. **Dificultades en el modelo RAG**

   Impacto: Medio

El modelo RAG enfrenta desafíos de ajuste fino, interpretabilidad y eficiencia en la recuperación de documentos. Estos riesgos pueden clasificarse en dos factores. Primero, al tratarse de una arquitectura relativamente nueva, existen pocos ejemplos de referencia que permitan guiar el desarrollo. Segundo, su demanda computacional requiere de GPU potentes, limitando las opciones de implementación, pero no resultan significativamente perjudiciales. Se debe tener en cuenta que, si la fase de recuperación no es efectiva, el modelo RAG generará respuestas basadas en documentos irrelevantes.

Para abordar estos problemas, se evaluaron distintas plataformas y alternativas, realizando pruebas en diferentes infraestructuras que se ajusten a las necesidades del proyecto. Además, se optimiza el modelo mediante ajustes de los componentes del sistema para mejorar la eficiencia computacional. Para mitigar la demanda computacional, una solución fue utilizar APIs de modelos en la nube, evitando la necesidad de desplegar infraestructuras locales avanzadas, facilitando la implementación en entornos con recursos limitados.

5. **Coste económico**

   Impacto: Medio

   En todos los LLM disponibles mediante API, como GPT-4 o Claude3, existe un coste asociado, que corresponde al tipo de modelo usado y el número de tokens requeridos, tanto en el prompt de entrada (input) como en la respuesta del LLM (output), limitando la complejidad del modelo.

   Se comprobaron alternativas para establecer la más económica, que optimice el coste y mantenga la calidad del resultado, eligiendo modelos con un coste por token menor, como GPT-3.5 Turbo o GPT-4o mini.

6. **Evolución constante del ámbito médico**

   Impacto: Medio

   El entorno médico y clínico está en constante evolución, creando y mejorando nuevas guías clínicas consantemente, por lo que es crucial que el sistema propuesto tenga la misma capacidad de evolucionar y acceder a información actualizada en tiempo real.

   Como medida de mitigación, se trabajaron exclusivamente las guías más recientes disponibles hasta la fecha de marzo de 2025, asegurando que el sistema se base en información actualizada. Además, se prioriza la extracción de datos únicamente de fuentes fiables, evitando el uso de información no verificada.

7. **Evaluación del sistema**

   Impacto: Medio

   El modelo de arquitectura RAG proporcionará respuestas generadas por el LLM basándose en las guías médicas. Para evaluar la precisión y relevancia de estas respuestas, se requiere de la validación de especialistas médicos en enfermedades reumáticas y musculoesqueléticas.

   Con el fin de reducir esta dependencia, se propone el uso de un LLM externo capaz de comparar y evaluar automáticamente las diferencias entre las respuestas, facilitando así el proceso de revisión y mejorando la eficiencia del análisis.

8. **Tiempo de redacción de la memoria final**

   Impacto: Bajo

   El tiempo de entrega del TFM puede verse afectado por posibles retrasos en el desarrollo y validación del modelo, comprometiendo la calidad del informe final. Este riesgo podría derivar en una documentación incompleta o una falta de análisis en profundidad.

   Para mitigar este riesgo, se establecen cronogramas con hitos claros, revisiones periódicas del avance y la realización paralela de tareas clave para optimizar los recursos disponibles. Del mismo modo, se optó por la redacción de la memoria final a lo largo del trabajo, avanzando en paralelo.

1.6. Breve resumen de los productos obtenidos

Al finalizar el Trabajo Final de Máster, se obtuvo uno de los primeros modelos RAG basado en guías médicas verificadas que se podrá emplear mediante la plataforma web en formato de asistente virtual.

Se obtuvo:

- **Un conjunto de datos** conformado por las guías clínicas reumatológicas con los cuales construir la base de datos vectorial.
- **Una base de datos vectorial** con los documentos procesados.
- **Un conjunto de preguntas** empleadas para la evaluación de los modelos.
- **Una interfaz web** que emplee el modelo RAG, basándose en las guías médicas para interactuar con los usuarios de forma amigable teniendo en cuenta la experiencia del usuario.
- **Un repositorio de GitHub** con el código empleado y los obtenibles resultantes, accesible a través del siguiente enlace: https://github.com/dcamacmon/DCM_ARQUITECTURA-RAG/
- **Una memoria** con la evolución del proyecto.
- **Una presentación virtual** con la tesis final.

1.7. Breve descripción de los capítulos de la memoria

La memoria consta de los siguientes capítulos:

Capítulo 2: Marco teórico

Este capítulo contiene una descripción detallada del funcionamiento de la arquitectura RAG, desglosada en las tres fases principales: recuperación de información, aumento del contexto y generación de la respuesta. Se explica cómo los sistemas RAG utilizan bases de datos vectoriales para determinar los fragmentos de información relevantes mediante embeddings, cómo se integran en el prompt del modelo y cómo el LLM genera respuestas a partir de ese contexto.

Además, se detallan aspectos clave de la arquitectura RAG, como la ingeniería de prompts, el uso de rerankers y las implicaciones inherentes. También aborda los posibles tipos de RAG en función del tipo de datos, ilustrando su expansión a entornos multimodales, finalizando con los retos actuales de esta arquitectura, como la dependencia de los datos, la carga computacional necesaria o los riesgos de seguridad que presentan.

Capítulo 3: Estado del arte

Este capítulo presenta una revisión del estado del arte en la aplicación de arquitecturas RAG en el ámbito clínico en diferentes especialidades. Se presentan estudios recientes, desde cardiología hasta nefrología, que evidenciaron empíricamente la mejora en la generación de respuestas clínicas a partir de la recuperación de información clínica fundamentada. Se detallan múltiples enfoques técnicos, como el uso de modelos de embedding especializados, la integración de bases de datos vectoriales y el uso de frameworks especializados.

También se abordan las mejoras obtenidas en métricas como la precisión y la reducción de alucinaciones, así como los desafíos relacionados con la calidad del preprocesamiento de la documentación o la necesidad de actualizaciones constantes de información.

Capítulo 4: Materiales y métodos

Durante este capítulo, se describe el diseño metodológico seguido para comparar un modelo generativo estándar (GPT-3.5 Turbo) con dos arquitecturas RAG desarrolladas a partir de las guías clínicas reumatológicas proporcionadas por la ACR: RAG500 (chunks de 500 caracteres) y RAG1000 (chunks de 1000 caracteres). Se detalla el preprocesamiento de las guías clínicas (parseo, limpieza, fragmentación y vectorización), el uso de frameworks especializados (LangChain) y técnicas de reranker.

La evaluación se basó en 6 métricas (fidelidad, relevancia, precisión, completitud relativa, completitud total y concisión), aplicadas a 170 preguntas clínicas generadas por un LLM y evaluadas por un modelo externo (Gemini AI 1.5 pro). Se realizaron comparaciones estadísticas 2 a 2 entre modelos, utilizando las pruebas de Binomial y Wilcoxon. Además, se llevaron a cabo dos análisis complementarios, evaluando el impacto del LLM generador (empleando GPT-4o-mini y GPT-4.1 mini) y el efecto del tamaño muestral (ampliando a 340 preguntas).

Finalmente, se desarrolló una interfaz web con Gradio para desplegar el modelo RAG1000 con funcionalidades como memoria conversacional y trazabilidad.

Capítulo 5: Resultados

En este capítulo se describen y analizan los resultados derivados de la evaluación de los tres modelos (el generativo base y los dos modelos RAG). La evaluación combina análisis

cuantitativos basados en métricas de calidad de respuesta y análisis cualitativos centrados en la preferencia entre modelos a partir de comparaciones pareadas.

Se presentan los resultados estadísticos obtenidos de la prueba binomial y la prueba de wilcoxon en las comparaciones 2 a 2 de los modelos en los tres LLM evaluados.

Capítulo 6: Discusión

Durante este capítulo se evidencia el potencial de RAG para generar respuestas clínicas más precisas, completas y alineadas con la consulta clínica, en comparación con los modelos generativos puros. Se observó que la combinación de un buen preprocesamiento, el tamaño de chunks de 1000 caracteres y la recuperación de información desde guías clínicas permitió minimizar las alucinaciones y mantener la coherencia clínica. Además, el uso de modelos más potentes no eliminó la potencia de RAG, reforzando el papel como complemento esencial para entornos médicos.

Sin embargo, al centrarse únicamente en reumatología, con un número limitado de documentos y sin evaluación humana experta, se redujo la aplicabilidad directa en la práctica clínica general. También se utilizó un entorno simulado (preguntas generadas por un LLM), sin considerar casos reales ni variables clínicas más complejas.

A pesar de estas limitaciones, los resultados sugieren aplicaciones prometedoras, como asistencia en atención primaria o urgencias en la toma de decisiones clínicas.

Capítulo 2

Marco teórico

2.1. Introducción a los LLM

Los recientes avances en el campo de la inteligencia artificial (IA) ha permitido el desarrollo de LLM como LlaMA, ChatGPT o Claude; algoritmos potentes capaces de sintetizar grandes cantidades de datos y realizando una amplia variedad de tareas en respuesta a comandos humanos.

Los LLM están diseñados principalmente para procesar los datos en formato de texto, pero hay modelos multimodales que trabajan con otros tipos de datos, como audio o video. Los LLM más conocidos comercialmente, como GPT-4 de OpenAI, Large Language Model Meta AI (LlaMA) de Meta AI o Claude de Anthropic son modelos que pueden realizar tareas y generar un texto coherente, trabajando solo con entradas y salidas textuales.

Los LLM, como GPT-4, funcionan mediante una combinación de Deep Learning, entrenamiento masivo con grandes volúmenes de texto y generación de texto basada en probabilidades. Específicamente, los LLM están basados en redes neuronales profundas, específicamente en la arquitectura Transformers introducida en 2017 (16). Dependiendo de su enfoque, pueden emplear modelos autorregresivos (como GPT), que generan texto prediciendo la siguiente palabra en función del contexto previo, asignando probabilidades a cada posible término y eligiendo el más probable. También pueden utilizar modelos de autoencoding (como Bidirectional Encoder Representations from Transformers (BERT)), que aprenden representaciones de texto enmascarando palabras dentro de una oración y tratando de predecirlas, lo que permite una mejor comprensión del contexto bidireccional (17).

2.1.1. Historia de los LLM

Los LLM sirven como piedra angular en el procesamiento del lenguaje natural (NLP, Natural Language Processing). De hecho, el lenguaje es la herramienta que permite expresar pensamientos y la comunicación entre dos entidades. Sin embargo, las maquinas carecen de la capacidad intrínseca de comprender y comunicarse en un lenguaje humano, por lo que requieren de algoritmos de IA para disponer de esta capacidad.

El Natural Language Processing (NLP) abarca una amplia gama de enfoques que incluyen modelos estadísticos, probabilísticos y modelos de lenguaje (18). Inicialmente, los modelos de procesamiento de lenguaje se basaban en técnicas estadísticas como los modelos de Markov

ocultos (HMM) o los Modelos de Lenguaje Estadísticos (Statistical Language Model (SLM)), que calculaban la probabilidad de ocurrencia de palabras o secuencias en un corpus de datos. Con el avance del Machine Learning (ML), surgieron los modelos basados en redes neuronales (como Word2Vec o GloVe), y posteriormente los modelos de LLM, basados en la arquitectura Transformers, que han revolucionado la generación y comprensión del lenguaje natural.

De forma general, los modelos de lenguaje han sufrido diferentes evoluciones que han permitido el desarrollo de los modelos actuales, los LLM. En la Figura 2.1, se puede observar los modelos más relevantes de cada tipo de LM, desde el nacimiento de los más básicos (N-grams) hasta los modelos de última generación basados en la arquitectura Transformer.

Si bien GPT-4 ha sido uno de los avances más significativos en este campo, actualmente existen otros modelos de alto rendimiento, como GPT-4o, Claude 3.5 Sonnet o Gemini 2.5 entre otros. Estos modelos continúan mejorando en eficiencia, capacidad de razonamiento y generación de texto más precisa y contextualizada.

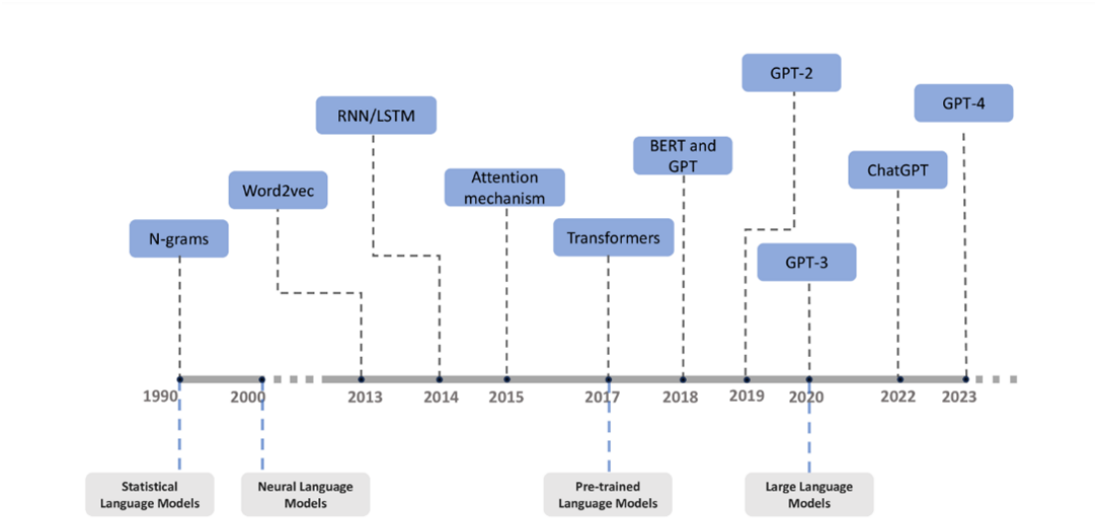


Figura 2.1: **Historia y evolución de los distintos modelos de procesamiento de lenguaje natural.** La figura ilustra la cronología de los avances clave en procesamiento del lenguaje natural, desde los primeros modelos estadísticos hasta los actuales modelos de lenguaje a gran escala (LLM). Se destacan la aparición de Transformers y modelos como BERT o GPT, mostrando una evolución hacia arquitecturas cada vez más eficientes y complejas. Extraída de (18).

A continuación, se presentan las principales categorías de modelos de lenguaje, destacando sus características, ventajas y limitaciones:

- **Statistical Language Models (SLMs)**: Son modelos matemáticos que emplean las propiedades contextualmente relevantes del lenguaje natural desde una perspectiva estadística probabilística. Su esencia radica en determinar la probabilidad de que una oración aparezca dentro de un contexto a partir de las probabilidades condicionales consecutivas de las unidades lingüísticas simples (palabras o letras) (19). Su mayor limitación radica en que, al tener solo en cuenta las dos primeras palabras precedentes a cada término, se pierde significativamente su precisión y contexto.

- **Neural Language Models (NLM)**: Emplean redes neuronales para predecir las probabilidades de palabras posteriores dentro de secuencias lingüísticas, y son capaces de comprender el concepto de las palabras mediante vectorización y su ubicación. Es decir, convierten las palabras a lenguaje binario, se transforman en vectores con una ubicación fija, de manera que pueden simular las relaciones entre palabras a través de los ángulos entre los vectores. Su mayor limitación es que disponen de un contexto limitado y, pese a generar un texto coherente, no poseen una comprensión real del significado, llegando a generar respuestas sintácticamente correctas pero sin sentido real (20; 21).

- **Pre-trained Language Models (PLM)**: Estos modelos se someten a un entrenamiento inicial con un gran volumen de texto no etiquetado, permitiendo captar estructuras lingüísticas fundamentales, como la sintaxis o la semántica, para posteriormente, realizar un segundo entrenamiento con conjuntos de datos más pequeños para "afinar" el conocimiento (*fine tuning*), de manera que se pueden realizar tareas más específicas sin perder calidad. Sin embargo, al depender del pre-entrenamiento, no se puede actualizar dinámicamente con nueva información sin ser reentrenados, además están enfocados en tareas más específicas en las que son entrenadas.

- **Large Language Models (LLM)**: Son los modelos más modernos. Se entrenan con corpus de texto con decenas de miles de millones de parámetros para comprender los comandos humanos. Igual que los PLM, pasan por un pre-entrenamiento inicial, para posteriormente ser alineados con los valores humanos en lugar de especializarse en un dominio específico como los PLMs. Destacan por su adaptabilidad y capacidad de aprovechar el contexto, pero su crecimiento depende del número de parámetros y datos de entrenamiento.

2.1.2. LLM: fundamentos, estado actual y familias

En los últimos años, ha habido diferentes factores que han permitido un rápido avance en los LLM. La disponibilidad de grandes volúmenes de texto de diferentes fuentes ha permitido mejorar su capacidad para generalizar y realizar múltiples tareas. Además, el crecimiento exponencial de la potencia computacional ha permitido entrenar modelos cada vez más grandes, ya que el desarrollo de hardware especializado como GPU ha acelerado su crecimiento (GPT-3 habría tardado 355 años en entrenarse en una sola GPU, pero con 1024Xa100 GPUs solo tardó 34 días) (18). El desarrollo de la arquitectura Transformer superó enfoques anteriores como redes neuronales recurrentes y convolucionales, optimizando la atención, el preentrenamiento y la eficiencia computacional (22).

Tomando como ejemplo a GPT-3, se pueden observar cuatro grandes principios que siguen los LLM para generar sus respuestas:

- **Entrada y codificación**: Reciben secuencias de palabras (*tokens*), convirtiendo cada palabra a un vector numérico (one-hot) dentro de un vocabulario de términos.

- **Embedding**: Para mejorar la eficiencia, los vectores *one-hot* se transforman en representaciones más compactas.

- **Codificación posicional**: Al carecer de estructura secuencial, se agregan codificaciones posicionales usando funciones trigonométricas para capturar el orden de las palabras en una oración.

- **Matriz de entrada**: La combinación de embeddings y codificaciones posicionales generará la matriz de entrada, que se procesará en las capas de transformadores.

Los atributos y comportamientos de los LLM están profundamente vinculados con los procesos del entrenamiento, caracterizado por tres etapas principales: preentrenamiento, ajuste fino supervisado (Supervised Fine-Tuning (SFT)) y aprendizaje de refuerzo por feedback humano (Reinforcement Learning from Human Feedback (RLHF)) (23). Estas fases permiten que los modelos no solo adquieren una comprensión básica del lenguaje, sino que también mejoren su capacidad para generar respuestas precisas y alineadas con los criterios humanos de calidad.

- **Pre-entrenamiento**: Permite adquirir conocimientos y habilidades fundamentales, entrenando mediante la predicción autorregresiva de tokens dentro de secuencias de texto. De esta manera, desarrollan una comprensión profunda de la sintaxis y habilidades de razonamiento, estableciendo una base sólida para su posterior ajuste y refinamiento (24).

- **SFT**: Entrena el modelo con un conjunto de datos anotado compuesto por pares de instrucciones y respuestas, mejorando la capacidad de los modelos para seguir instrucciones específicas, ofreciendo respuestas más precisas y útiles. Se ha confirmado la eficacia de este paso para un rendimiento adecuado en tareas no vistas previamente, mostrando capacidad de generalización (25; 26).

- **RLHF**: Se utiliza la evaluación de respuestas por parte de humanos para optimizar el comportamiento del modelo, para maximizar la recompensa determinada, usando algoritmos de aprendizaje por refuerzo. Esto permite que el modelo genere respuestas más alineadas con las expectativas humanas, mejorando la calidad de las respuestas y reduciendo la generación de contenido sesgado (27).

2.1.3. Limitaciones de los LLM

Los modelos de LLM han demostrado ventajas significativas frente a otros modelos de lenguaje. Sin embargo, a pesar de su eficiencia en aplicaciones generales, los modelos disponibles comercialmente presentan limitaciones en el ámbito clínico y médico, llegando a generar, en ocasiones, *alucinaciones*, es decir, respuestas con referencias incompletas o incluso completamente inventadas. Esta falta de precisión compromete la fiabilidad de los resultados, lo que, en un contexto de diagnóstico médico, puede tener consecuencias graves (2).

A pesar de su capacidad de generar texto coherente y realizar tareas complejas, los LLM tienen deficiencias estructurarles, como la falta de comprensión real, ya que no disponen de un razonamiento ni conocimiento como un humano, sino que identifican patrones en datos, por lo que pueden llegar a generar información inconsistente (28; 29).

Sin embargo, se deben reconocer las desventajas que acompañan estos modelos.

- **Sesgo**: Las decisiones pueden presentar sesgos en contra de poblaciones o grupos más marginados, ya que durante el entrenamiento con datos extensos y no estructurados, se pueden

absorber representaciones erróneas y comportamientos excluyentes. Esta problemática se mitiga eliminando los datos claramente sesgados del conjunto de entrenamiento, sin embargo esta solución no es completamente efectiva, ya que puede afectar la capacidad del modelo para generalizar correctamente y reducir su efectividad en ciertas tareas (30). El sesgo no solo proviene de los datos de entrenamiento, sino también de su funcionamiento interno. Por ejemplo, el modelo puede generar el siguiente token basándose en sus propias predicciones previas, lo que puede dar lugar a errores acumulativos (sesgo de exposición), o cuando prioriza el conocimiento aprendido durante el entrenamiento sobre la información proporcionada en la entrada (sesgo de conocimiento paramétrico) (31).

- **Seguridad y alucinaciones**: Los LLM tienen diversas aplicaciones en industrias como el ámbito médico, legal o de finanzas, por lo que la veracidad de la información es especialmente relevante. En ocasiones, pueden generar resultados que difieren del contexto proporcionado o del conocimiento factual, denominados *alucinaciones* (31).

- **Privacidad**: Los modelos de lenguaje a gran escala pueden implicar riesgos para la privacidad, especialmente si se entrenan con datos sensibles o sin un adecuado control de filtrado. La exposición de información personal o confidencial puede representar un problema crítico en su implementación (32). Para mitigar estos riesgos, se implementan guardrails, que incluyen técnicas como la anonimización de datos, la detección y eliminación de información sensible en las respuestas del modelo, el uso de mecanismos de acceso restringido y auditorías periódicas del sistema. Estas estrategias buscan garantizar que los modelos cumplan normas de privacidad y eviten la divulgación de datos confidenciales (33; 34).

De hecho, las alucinaciones en los LLM fundacionales como ChatGPT o Gemini pueden parecer mostrar capacidades de razonamiento muy elevadas, por lo que pueden pasar desapercibidas y dar lugar a resultados falsos o sin fundamento. Esto puede generar desconfianza en su aplicación en ámbitos críticos, como el legal o el médico, donde la precisión y la fiabilidad de la información son fundamentales (35).

2.2. Sistemas RAG

Tanto los LLM como las IA han demostrado un gran potencial para mejorar la eficiencia de la interacción de los médicos y los pacientes con los sistemas de atención médica (36). No obstante, debido a que toda la información procede de los datos proporcionados por el entrenamiento, esta información no se actualiza regularmente, pudiendo inducir a errores.

Para abordar esta problemática, la arquitectura RAG ha ganado popularidad gracias a su capacidad de integrar información externa verificable en tiempo real, en lugar de depender exclusivamente del conocimiento interno del modelo. De este modo, se incorpora el contenido de fuentes específicas y actualizadas, clave en entornos clínicos y médicos donde la evidencia científica evoluciona rápida y constantemente.

Estudios recientes han demostrado que, con un adecuado preprocesamiento documental, este tipo de sistemas puede alcanzar niveles de precisión cercanos al 99 % en tareas especializadas (37).

2.2.1. Arquitectura de RAG

La arquitectura Retrieval-Augmented Generation combina la generación de un texto mediante un LLM con la recuperación de información externa para mejorar la calidad y la precisión de las respuestas generadas. Esta arquitectura se basa en dos etapas principales: la búsqueda y recuperación de información relevante en una base de datos (*retriever*) y posteriormente se emplea la información recuperada para generar una respuesta más precisa y contextualizada, accediendo a información que el LLM no disponía durante su entrenamiento, el módulo *generator*. De este modo, el enfoque RAG permite al modelo LLM una mayor versatilidad y eficiencia, especialmente en tareas que requieren de conocimiento específico y actualizado sin necesidad de un re-entrenamiento (38).

En general, una arquitectura RAG inicia con la consulta de entrada, la cual pasa al módulo recuperador que accederá al vectorstore para buscar los k documentos más relevantes. Posteriormente, se procesarán junto con la consulta al módulo generador como contexto para que produzca una salida condicionada. La Figura 2.2 muestra el workflow genérico de las arquitecturas RAG.

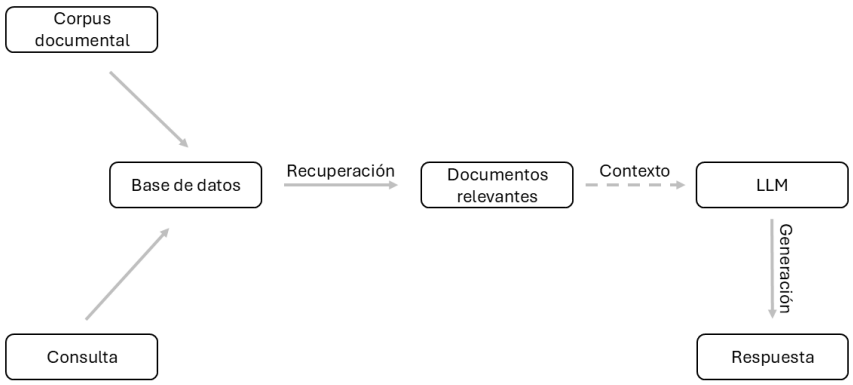


Figura 2.2: **Esquema de una arquitectura RAG.**
El módulo recuperador selecciona información relevante de una base de datos generada a partir del corpus documental. Esta información, evaluada según su relevancia frente a la consulta, es proporcionada al modelo LLM para generar la respuesta final.

Framework

Durante el desarrollo de sistemas de RAG, el uso de frameworks especializados facilita la integración entre los modelos de lenguaje y la bases de datos vectoriales, optimizando la eficiencia y precisión en la recuperación y generación de respuestas. Actualmente existen múltiples herramientas especializadas, destacando LangChain, LlamaIndex y Haystack. Estos frameworks proporcionan una infraestructura que permite la implementación de técnicas avanzadas cómo

la búsqueda semántica, re-ranking y generación de texto en base a un contexto, imprescindibles en un modelo de RAG.

La elección de uno de estos frameworks dependerá de sus características como la escalabilidad, compatibilidad con los LLM y la flexibilidad con la personalización del workflow diseñado.

- **LangChain (39)**: Está diseñado para facilitar la integración de LLM con fuentes de datos externas, bases de datos vectoriales y herramientas adicionales. Permite personalizar cada componente del *pipeline*, además de integrar herramientas nativas como la memoria conversacional, haciéndola una gran opción en modelos de RAG. Su estructura básica se centra en los *retrievers* (módulo de recuperación basada en embeddings), *chains* (flujos de trabajo), memoria y agentes (que ejecutan workflows dinámicos con toma de decisiones).
- **LlamaIndex (40)**: Está especializado en la indexación y recuperación eficiente de información desde documentos estructurados (y semiestructurados), optimizando la integración con modelos de LLM para generar respuestas basadas en información. Es compatible con múltiples LLM y fuentes de datos. Su arquitectura se basa en cargadores de documentos (*Document Loaders*), Indexadores (estructurando la información de bases vectoriales) y *Query Engine* (el motor de búsqueda semántica).
- **Haystack (41)**: Es un framework de código abierto que permite crear aplicaciones LLM y canales generativos con recuperación mejorada y sistema de búsqueda en grandes colecciones de documentos. Su estructura modular permite combinar diferentes motores de búsqueda, con pipelines híbridos e integración con bases de datos vectoriales. Su arquitectura se basa en *Document Stores* (almacenajes basados en SQL-NoSQL), *Retrievers* (módulo de recuperación), *Readers* (Modelo de NLP para la extracción) y el *Generator* (Modelo de LLM para la generación de respuesta).

Pre-procesados: parseo, chunking y vectorstore

Para el eficiente procesamiento de datos en la recuperación de la información, las arquitecturas RAG se fundamentan en dos componentes clave: la recuperación y la generación. Para que el modelo pueda acceder a los documentos más relevantes, se debe crear un repositorio de los mismos, requiriendo un preprocesamiento adecuado. Esto implica convertir la información desde un formato no estructurado, como los archivos en formato PDF de las guías clínicas a accesible para su análisis computacional. Este proceso implicará la extracción de texto de los documentos, su normalización y limpieza, así como la transformación de la información textual en vectores numéricos mediante técnicas de embeddings. Finalmente, estos embeddings se almacenarán en una base de datos vectorial (Vectorstore), que permitirá gestionar grandes volúmenes de información de forma eficiente, facilitando la búsqueda, comparación y análisis semántico, optimizando la recuperación de documentos relevantes y mejorando el rendimiento general del sistema.

Parseo: Extracción del texto

El primer paso en el proceso de preprocesamiento de documentos es el análisis de los datos o parseo, que implica extraer y organizar la información desde fuentes no estructuradas (como

los archivos PDF de las guías clínicas reumatológicas) para convertir los datos en un formato adecuado para el procesamiento computacional de la información. Este proceso es crítico, ya que debido a la complejidad del formato PDF y su carencia de una estructura semántica clara, lo hace más difícil de manejar en comparación con otros formatos como Extensible Markup Language (XML) o HyperText Markup Language (HTML), que tienen una jerarquía definida en los datos.

Existen múltiples herramientas que permiten extraer el contenido de documentos en formatos diferentes, como XML o Markdown. El formato PDF en específico es complejo de extraer, ya que carece de una estructura semántica clara como otros formatos como XML o HTML, que definen los datos con una estructura jerárquica personalizada.

Los métodos de parseo se pueden clasificar en dos enfoques principales:

El **primer enfoque** sigue un conjunto de reglas predefinidas para extraer la información de los documentos, basándose en patrones o estructuras bien definidas dentro del documento. De esta manera, emplea expresiones regulares o delimitadores ya presentes en los documentos, siendo altamente preciso en documentos con formatos uniformes y bien estructurados, como XML. Este enfoque es rápido y eficiente, ya que no requiere de entrenamiento de modelos de ML, sin embargo, son dependientes de etiquetas de estructura bien definidas, lo que los hacen menos flexible ante variaciones de formato. Un ejemplo de este enfoque es *PDFminer*, que extrae texto de archivos PDF.

El **segundo enfoque**, emplea modelos de NLP y ML para interpretar y estructurar el contenido del documento sin seguir las reglas rígidamente. Esto permite manejar documentos con estructuras complejas, como los artículos científicos (con más de una columna, gráficos y tablas), de manera que extrae no solo el texto, sino que emplea técnicas avanzadas para extraer las tablas y las referencias del texto. Al emplear métodos más avanzados, regularmente son más costosos computacionalmente, como GeneRation Of BIbliographic Data (GROBID), que estructura artículos científicos en un formato XML y Text Encoding Initiative (TEI).

Actualmente existen múltiples herramientas comerciales disponibles, siendo capaces de extraer la información de los documentos PDF en diferentes formatos (42). Algunas herramientas son las siguientes:

- **PyMuPDF**: Se trata de una librería de Python ligera y rápida que permite la extracción de texto y estructuración de documentos. Permite navegar y extraer elementos específicos de las páginas del documento a través de las diferentes clases y métodos que ofrece el paquete. Los formatos de salida pueden ser desde texto plano sin estructura (TXT) hasta HTML o XML con información jerárquica.

- **Pdfplumber**: Es una librería de Python diseñada para la extracción avanzada de datos desde archivos PDF. Se especializa en la extracción de texto, tablas e imágenes enfocado en la precisión y la estructuración. Es especialmente práctico en documentos complejos, con múltiples columnas, tablas embebidas o textos con formatos irregulares.

- **GROBID**: Es una herramienta de código abierto diseñada para la extracción y estructuración automática de información en documentos científicos. A diferencia de los modelos anteriores, emplea técnicas de NLP y ML para transformar artículos académicos en formatos estructurados como XML y TEI. Además del texto, también permite la extracción de metadatos, como los títulos, referencias o palabras clave.

- **LlamaParse**: Emplea una combinación de IA, visión computacional y técnicas de NLP para extraer información de documentos complejos. Se destaca por su capacidad de manejar documentos con objetos embebidos como tablas, proporcionando una representación en formato Markdown.

A pesar de que existen herramientas altamente eficaces para el parseo, cada una presenta limitaciones que dificultan una extracción perfecta en todos los casos. Además, se complica aún más cuando se trabaja con documentos heterogéneos, como las guías clínicas, ya que pueden tener estructuras diferentes entre sí, como tablas, columnas múltiples o encabezados variados.

Por ello, es habitual seleccionar una herramienta principal en función del tipo de documento, y complementar el documento de salida mediante procesos de postprocesamiento u otras herramientas adicionales. Por ejemplo, GROBID es especialmente útil para estructurar documentos con múltiples columnas, como los artículos científicos, pero presenta limitaciones en la extracción precisa de tablas. En cambio, herramientas como LlamaParse destacan en el tratamiento de tablas embebidas, aunque presenta dificultades para separar correctamente columnas.

Pese a que hay herramientas significativamente eficaces en el parseo, todas presentas ciertas limitaciones que imposibilitan una única selección con eficacia absoluta. Si, además, añadimos un conjunto de documentos que no tienen la misma estructura en todos los documentos, la tarea de extracción puede hacerse aún más compleja con solo una herramienta de parseo. Por lo tanto, para optimizar dicha tarea, se debe realizar una combinación de dos herramientas que se complementen en sus características, como GROBID, que permite identificar entre columnas, pero dificulta la extracción de tablas, y LlamaParse, que permite la extracción de tablas pero tiene dificultades con la diferenciación de columnas.

Chunking, la fragmentación de documentos

El siguiente paso en el preprocesamiento de los datos es la fragmentación (*chunking*) que consiste en dividir el texto extraído en fragmentos (*chunks*) más pequeños, que pueden ser desde párrafos completos a oraciones o unidades lingüísticas más simples. Esta fragmentación permite una recuperación más precisa y específica, pues el modelo se puede enfocar en unidades de texto más manejables y relevantes y, al gestionar fragmentos más pequeños, se optimiza el modelo en términos de memoria, y permite una mejor paralelización de las tareas de procesamiento (43).

Se ha demostrado que los sistemas RAG que emplean LLM suelen generar respuestas inexactas al recuperar todo un documento, ya que se incorpora información innecesaria que introduce ruido en el sistema (44). Esto se debe a que la información excesiva distorsiona el resultado de dichos modelos, reduciendo la confiabilidad del sistema, sobre todo en tareas críticas como el asesoramiento clínico. En la Figura 2.3 se puede observar el rendimiento de dos modelos, detallando el efecto del chunking, demostrando la mejora del renidmiento de la recuperación y de la generación de la respuesta gracias a la fragmentación.

Los modelos RAG basados en fragmentos como ChunkRAG han demostrado una eficiencia mucho más elevada respecto a los modelos basados en recuperación de documentos completos, sobre todo en documentos largos y complejos (44; 45). Los modelos basados en chunks realizan una jerarquización de los fragmentos, de manera que puede considerar aquellos más relevantes para recuperar, dando lugar a un incremento de la precisión general del modelo.

Además de la jerarquización de los fragmentos, el sistema también será capaz de comparar fragmentos entre ellos por similitud, filtrando los resultados para evitar información repetida

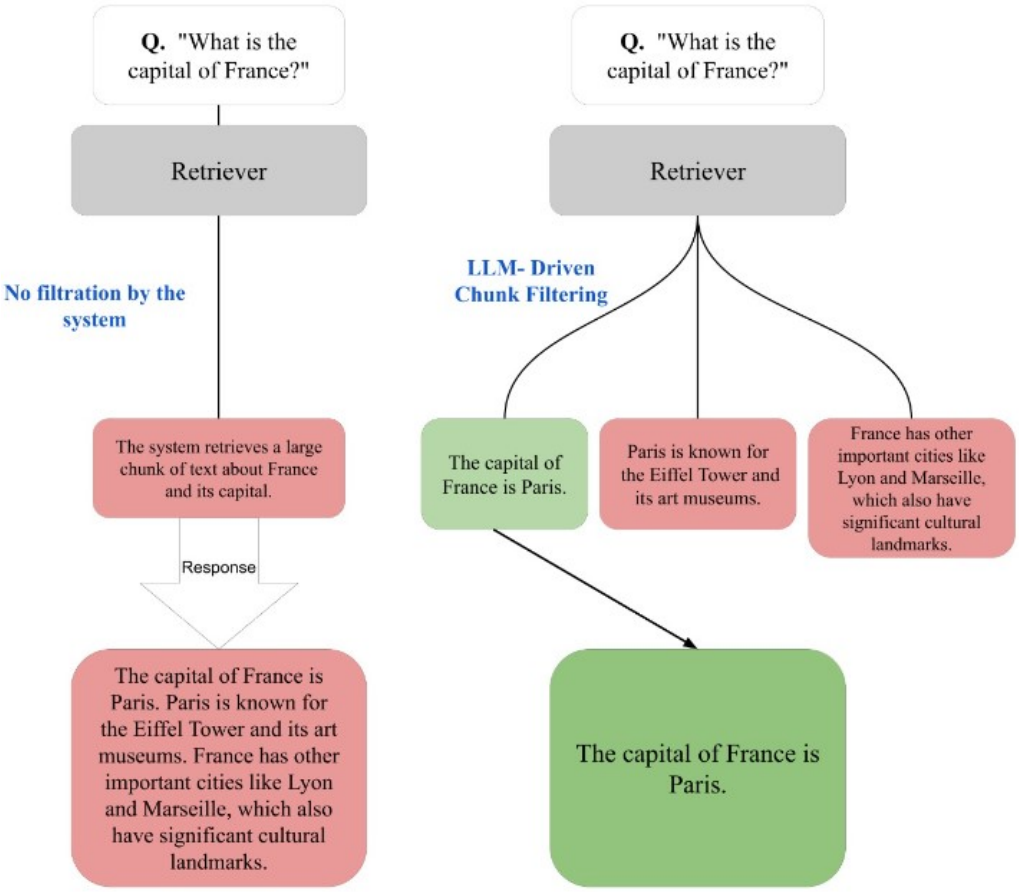


Figura 2.3: **Comparación del efecto del *chunking***.
La figura muestra cómo la longitud del contexto afecta a la generación de respuestas: en el primer flujo, al incluir demasiado contexto, la respuesta incorpora información no relacionada con la pregunta. En cambio, en el segundo flujo, al usar chunks más pequeños, la respuesta es más precisa y concisa. Extraído de (44).

o redundante que dificultan la capacidad del modelo para generar repuestas coherentes y únicas. Existen múltiples enfoques que analizan la similitud entre fragmentos, como el cálculo de similitud de coseno, que calcula el ángulo entre dos vectores distintos de cero en un espacio de producto interno, cuantificando la similitud entre dos vectores independientemente de su magnitud (46).

También existen modelos que, para optimizar la recuperación de fragmentos, emplean una búsqueda basada en caché. Este enfoque permite reducir el cálculo redundante en un 51 % en cargas de trabajo de producción reales, mejorando así la latencia y el rendimiento en tareas de inferencia (47). El almacenamiento de caché de claves y valores permite almacenar fragmentos de información previamente procesada para su reutilización, reduciendo la redundancia computacional. Sin embargo, este sistema enfrenta limitaciones en cuanto a la sobrecarga computacional en toma de decisiones en tiempo real, además de la baja flexibilidad ante diferentes modelos de lenguaje. Esto refleja la importancia del almacenamiento de memoria, pero también implica

una limitación en el almacenamiento de caché.

Se debe tener en cuenta que optimizar el tamaño de los fragmentos es esencial para la precisión y la recuperación eficiente, sin embargo, determinar el tamaño adecuado depende de la estructura de los datos y la densidad de información e incluso, en la misma base se pueden encontrar diferentes grados de granularidad (48). De hecho, otros factores como la misma pregunta pueden afectar al tamaño del chunk óptima, debido a que en función de la granularidad de la *query*, el tamaño del chunk puede variar (si se solicita una información más amplia, se prefiere una granularidad más gruesa). En conclusión, se debe encontrar un equilibrio en la granularidad, ya que una granularidad gruesa proporciona más información con menor precisión, mientras que la fina ofrece información completa a costa de la eficiencia.

Por lo tanto, las técnicas de chunking pueden clasificarse en:

- **Basado en tokens**: Este enfoque divide el texto en fragmentos de tamaño especificado según los recuentos de tokens sin procesar. Permite definir una ventana de solapamiento o superposicón (*overlap*) para respetar el contexto entre fragmentos (48). Aunque es eficiente, puede cortar frases o ideas en puntos arbitrarios, afectando la coherencia y el contexto del framento. En LangChain, se denomina `TokenTextSplitter`.

- **Basado en oraciones**: Este método se centra en dividir el texto en unidades más naturales, como oraciones completas, intentando mantener unidas las oraciones y los párrafos. De esta manera, se preserva mejor la coherencia y el contexto, mejorando la eficacia en la recuperación de información. Sin embargo, puede resultar menos eficiente si el texto no tiene una estructura clara o uniforme, generando fragmentos con tamaño muy diferente. En LangChain, se llama `SpacyTextSplitter`.

- **Recursivo**: La fragmentación recursiva divide el texto de manera iterativa hasta obtener el tamaño adecuado, intentando mantener el mayor tiempo posible párrafos completos para conservar la relación semántica. Es un enfoque bastante flexible, ya que permite establecer limitadores de tamaño y puede ajustarse a diferentes estructuras de texto, pero su efectividad depende de la estructura de los datos procesados. En LangChain, se denomina `RecursiveCharacterTextSplitter`.

En modelos RAG, la precisión de la recuperación depende en gran medida de los fragmentos de información. Por ello, es importante no solo el tamaño del chunk, si no que también es relevante que no se pierda el contexto dentro del mismo fragmento, por lo que **RecursiveCharacterTextSplitter** destaca entre el resto de técnicas.

Vectorización: embedding

Una incrustación (*embedding*) es una representación numérica densa de datos, generalmente en forma de un **vector**, en un espacio de alta dimensión. Su objetivo es transformar la información, como palabras, oraciones o conceptos en vectores que capturen las relaciones y características semánticas del contenido original, que pueden ser usados por sistemas computacionales (49). En el contexto del NLP, los embeddings convierten frases en vectores de números que permiten que el modelo entienda la similitud, el contexto y las relaciones entre los términos de manera más eficiente.

Los inicios de los modelos de embedding se remontan a los enfoques basados en frecuencia que permitían calcular la relevancia de términos en documentos, pero sin capturar las relaciones semánticas entre las palabras del contexto. Posteriormente, se introdujo la idea de representar palabras como vectores en un espacio continuo, de manera que términos con significado similar mantuvieran la similitud en las representaciones mediante distancia. Con la llegada de la arquitectura Transformers se introdujeron los embeddings contextuales, creando modelos como BERT y GPT, permitiéndoles adaptarse al contexto de cada palabra en una oración, permitiendo su incorporación en tareas de recuperación de información y el desarrollo de arquitecturas RAG.

Durante el proceso, los fragmentos de texto del corpus documental fueron transformados en vectores mediante un modelo de embedding y almacenados en una base de datos vectorial. Esta representación permite comparar y recuperar información de manera eficiente en función del contenido semántico. Cuando el usuario introduce una pregunta, esta también se convierte en vector usando el mismo modelo de embedding. A continuación, se calcula la similitud entre el vector de la pregunta y los vectores almacenados en la base de datos, utilizando métricas como la distancia del coseno o la distancia euclídea. Los fragmentos más cercanos en este espacio vectorial son considerados los más relevantes semánticamente, y se seleccionan como contexto para que el modelo generativo genere la respuesta.

De este modo, palabras estructuralmente diferentes como “perro” y “cachorro” serían palabras cercanas por su semántica, pese a que su escritura sea diferente.

Modelos como *OpenAI Embedding* convierten fragmentos de texto en vectores densos que pueden ser empleados para buscar y recuperar información relevante en bases de datos vectoriales. Esto permite mejorar la recuperación de documentos relevantes, de manera que el modelo recibe el contexto más relevante para generar respuestas informadas.

Los modelos de embeddings, pese a su efectividad presentan viarias limitaciones (50):

- **Dependencia de modelos de compresión grandes:** Estos métodos se basan en modelos de compresión grandes para lograr una efectividad, incrementando los costos computacionales y dificultando su implementación en entornos con recursos limitados.
- **Baja efectividad en la generación de respuestas:** Actualmente no aprovechan completamente el potencial de los LLM, ya que optimizan solo ciertos componentes del sistema, dejando el decodificador del modelo sin modificaciones.
- **Tasa de compresión fija:** Los métodos actuales no ofrecen diferentes tasas de compresión con respecto a la longitud del contexto de entrada, impidiendo optimizar el equilibrio entre velocidad de inferencia y la calidad de generación.
- **Limitación a un solo documento:** Los métodos efectivos actuales admiten en su mayoría la generación de respuestas a partir de un único documento, restringiendo la utilidad en tareas de análisis e integración de información de múltiples fuentes, como la generación de respuestas complejas basadas en diversos documentos.

Actualmente, existen múltiples modelos de embedding para realizar el proceso de embedding. La plataforma HuggingFace (51) proporciona una tabla de clasificación en Massive Text Embedding Benchmark (MTEB) Leaderboard, donde se comparan diferentes modelos de embedding en tareas del MTEB.

Esta clasificación ayuda a evaluar el rendimiento de los diferentes modelos de embedding en diferentes escenarios, permitiendo seleccionar el modelo más adecuado en función de las necesidades. La tabla proporciona las características y métricas evaluadas en diferentes contextos. En el contexto de RAG y la recuperación de información, las métricas más destacables serian:

- **Retrieval (Recuperación)**: Evalúa qué tan bien el embedding recupera información relevante.
- **Semantic Textual Similarity (STS)**: Mide qué tan bien el embedding entiende relaciones entre textos similares.
- **Re-ranking**: Ayuda a mejorar los documentos recuperados en RAG, priorizando ciertos documentos.
- **Bitext Mining**: Capacidad de identificar pares de textos equivalentes en distintos idiomas (para RAG multilingüe).

Se extrajeron las 4 mejores opciones en base a estos parámetros, teniendo en cuenta su compatibilidad con un buen rendimiento en RAG con LangChain. De este modo, se plantearon estas opciones:

- **Alibaba NLP (gte-Qwen2-7B-instruct)**: Presenta una gran recuperación y STS, además de que es open-source (gratuito). Soporta preguntas complejas en RAG, funcionando bien con retrieval y generación de contexto relevante. No tiene integración directa con LangChain.
- **Intfloat (e5-mistral-7b-instruct)**: Tiene un buen equilibrio entre retrieval y re-ranking, y está optimizado para tareas de NLP, pero tiene un consumo mayor que otros modelos, además de una integración menos directa con LangChain.
- **Gemini (gemini-embedding-exp-03-07)**: Se destaca tanto en re-ranking, retrieval y STS, optimizado para comprensión semántica profunda y tareas de recuperación de información. Tiene acceso restringido y falta de integración optimizada para LangChain.
- **OpenAI (text-embedding-3-large)**: Tiene una buena optimización para retrieval, STS y re-ranking. Es totalmente compatible con LangChain e incluso con el vectorstore de ChromaDB. Tiene una eficiencia y costo superior a otros modelos, sin requerir de una infraestructura propia.

Entre las opciones disponibles, OpenAI destaca entre ellas al disponer de integración nativa con LangChain y ChromaDB, simplificando su implementación. Además ofrece un rendimiento óptimo en los parámetros de retrieval, STS y re-ranking, garantizando una recuperación de información eficiente. También evita la necesidad de infraestructura propia para la inferencia, optimizando tanto en costos como eficiencia. Finalmente, su escalabilidad y persistencia asegura que el sistema, en caso de ser necesario, pueda crecer con el tiempo.

El resto de alternativas, pese a tener ventajas en ciertos aspectos, presentan también limitaciones clave en integración, costos o flexibilidad.

Base de datos vectoriales: vectorstores

Para almacenar y consultar embeddings, se emplean bases de datos vectoriales (*Vectrostores*), optimizadas para realizar búsquedas rápidas y eficientes, permitiendo recuperar los fragmentos de texto más relevantes en función de una consulta específica. Esto facilita una recuperación precisa y relevante de la información, mejorando la eficiencia en la generación de respuestas basadas en el contexto adecuado. Estas bases especializadas permiten almacenar datos de alta dimensión que no han podido ser caracterizados por sistemas de gestión de bases de datos tradicionales. Estos vectores pueden tener una cantidad variable de dimensiones, dependiendo de la granularidad de los datos, y representan matemáticamente las características o atributos de los fragmentos de texto (52).

Estas bases de datos especializadas presentan ciertas ventajas respecto a las bases de datos tradicionales:

- **Búsqueda y recuperación por similitud**: Pueden encontrar los datos más relevantes según distancia en el espacio vectorial, siendo especialmente práctica en aplicaciones de NLP, a diferencia de las bases de datos tradicionales, que no pueden captar el significado contextual o semántico en el texto.
- **Soporte para datos complejos y no estructurados**: Permiten almacenar y buscar datos con alta complejidad y granularidad, sin necesidad de estar estructurados como en las bases de datos tradicionales.
- **Escalabilidad y rendimiento**: Pueden manejar análisis y procesamiento de datos a gran escala y en tiempo real, esencial para aplicaciones de IA. Pueden distribuir la carga de trabajo y reducir la latencia en consultas a gran escala mediante técnicas avanzadas de indexación y paralelización.

Actualmente, los tres modelos de vectorstore más populares son ChromaDB, MongoDB (Atlas Vector Search) y Facebook AI Similarity Search (FAISS), cada una con sus características y limitaciones (53; 54).

- **MongoDB** soporta búsquedas vectoriales dentro de una base de datos documental, y permite combinar consultas estructuradas y vectoriales (55).
- **FAISS** es una librería optimizada para búsqueda rápida en colecciones de vectores, soportando una indexación eficiente y búsqueda aproximada (lo que reduce la latencia en consultas grandes) (56).
- **ChromaDB** es una base de datos optimizada para búsquedas por similitud, permitiendo almacenar y consultar metadatos adicionales. Tiene una integración nativa con el framework de LangChain, y se puede emplear en entornos locales y escalables en la nube (57).

Para determinar la base de datos vectorial adecuada, se debe tener en cuenta ciertos aspectos clave: la integración con el framework utilizado (LangChain), la posibilidad de almacenar metadatos y la facilidad para implementar el modelo y escalarlo. En la Tabla 2.1 se puede observar un resumen de las capacidades de cada modelo.

Respecto a la integración con LangChain, MongoDB no tiene una integración nativa con el framework (a diferencia de ChromaDB), así como FAISS, que pese a ser eficiente en búsquedas vectoriales, tampoco tiene un soporte nativo.

En cuestión de soporte para el **almacenamiento y metadatos**, MongoDB puede almacenar los metadatos en documentos JavaScript Object Notation (JSON) dentro de una base de datos documental, pero requiere de indexación específica para la búsqueda por similitud (algo que ChromaDB hace de forma automática). FAISS, por otro lado, no puede almacenar los metadatos dentro de la base de datos, requiriendo un almacenaje externo.

En términos de **facilidad de implementación**, MongoDB requiere el uso de Atlas Vector Search, una funcionalidad de pago, requiriendo de una implementación compleja de comprender. FAISS, al ser una librería optimizada en búsquedas, no es una base de datos persistente, por lo que para mantener los datos a largo plazo requiere de un almacenamiento externo.

Los embeddings, en un modelo RAG, se recuperarán mediante similitud con la pregunta del usuario. De este modo, uno de los factores clave en la selección de la base de datos vectorial será su optimización para **búsquedas por similitud**. ChromaDB, como se ha mencionado anteriormente, está diseñado específicamente para este propósito. MongoDB, pese a soportar estas búsquedas, su rendimiento es inferior ya que está diseñado como base de datos documental. FAISS, aunque es altamente eficiente en búsquedas vectoriales y está optimizado para grandes volúmenes de datos, carece de almacenamiento persistente.

Finalmente, debido a que las guías clínicas que se emplearán en este modelo pueden aumentar con el tiempo (con nuevas publicaciones, por ejemplo), el modelo seleccionado debe tener una **escalabilidad y flexibilidad** adecuada. MongoDB tiene una alta escalabilidad, pero depende del uso de Atlas Vector Search (de pago), limitando su uso en grandes volúmenes de datos. FAISS, pese a ser eficiente en búsquedas a gran escala, no está diseñado para ser una solución completa. ChromaDB, en cambio, se puede ejecutar tanto localmente como en la nube, con opciones de persistencia que permite escalar en función de las necesidades del modelo.

Cuadro 2.1: Comparativa de características entre ChromaDB, MongoDB y FAISS para su integración en arquitecturas RAG.

	ChromaDB	**MongoDB**	**FAISS**
Integración con LangChain	Integración nativa	No nativa	No nativa
Soporte para metadatos	Incluidos automáticamente	Requiere JSON y configuración adicional	No almacena metadatos (soporte externo)
Facilidad de implementación	Instalación directa	Requiere Atlas Vector Search (de pago) y configuración compleja	Técnica; requiere almacenamiento externo
Optimización para búsqueda por similitud	Diseñado para ello	No optimizada (base documental)	Optimizado para búsquedas vectoriales
Persistencia de datos	Sí (local o nube)	Sí (base documental)	No (almacenamiento externo)
Escalabilidad	Flexible y adaptable	Dependiente de Atlas Vector Search	Escalable en búsqueda, pero no solución completa

Arquitectura RAG

La arquitectura RAG está compuesta por 3 fases esenciales para la generación de respuestas mejoradas: la recuperación de información, el aumento del contexo y la generación de la respuesta. Cada elemento tiene un papel crucial en la mejora de la precisión y relevancia de las respuestas generadas por el modelo.

Recuperación de información: *retrieval* (R)

En esta fase, el *retriever* buscarán la información más relevante dentro de la base de datos vectorial en base a la consulta realizada, que contendrá los fragmentos de información y sus embeddings asociados. De este modo, la query realizada por el usuario sirve como input para el modelo de recuperación y buscará la información más relevante.

La información recuperada será la base para los posteriores procesos, por lo que se deben tener en cuenta las limitaciones que afecten a la calidad de la respuesta generada, por lo que debemos asegurar que el LLM generador de la respuesta se integre eficientemente con el módulo recuperador.

Durante la configuración del módulo de recuperación, el parámetro k define la cantidad de documentos que se recuperarán en el proceso, y su ajuste tiene un impacto directo en la calidad de las respuestas generadas. Recuperar demasiados documentos puede introducir información irrelevante, confundiendo al modelo, reduciendo la precisión y aumentando de forma innecesaria la carga computacional. Por el contrario, si se recuperan muy pocos documentos,

el modelo podría no contar con el contexto suficiente, lo que aumenta el riesgo de generar respuestas inexactas o alucinadas. Además, una recuperación limitada podría reflejar información sesgada, sin representar adecuadamente la diversidad del conocimiento disponible, dando lugar a respuestas parciales o incompletas.

Augmented (A)

En el contexto de RAG, el proceso de Augmented Retrieval juega un papel determinante en la mejora de la calidad de las respuestas generadas. En esta fase, una vez se han recuperado los fragmentos más relevantes del conjunto de datos, se emplearán estos fragmentos para proporcionar contexto adicional a la consulta realizada.

El contexto recuperado contiene información relevante y específica relacionada con la pregunta del usuario. Este contexto se introduce como parte del prompt en un modelo de LLM. Este prompt se construye con la consulta del usuario, combinándola con los fragmentos recuperados. El LLM generará una respuesta que no solo se basará en la consulta, sino que también considera el contexto proporcionado para intentar producir una respuesta más precisa, completa y relevante.

Este enfoque permite que el LLM aproveche el conocimiento externo, superando las limitaciones del modelo de generación de respuestas a partir de conocimiento proporcionado durante el entrenamiento del modelo. La clave de la Recuperación Aumentada, es que permite adaptar el contexto recuperado para que el modelo genere respuestas más informadas y específicas a preguntas complejas, mejorando la calidad de las respuestas en comparación con respuestas generadas sin contexto.

La ingeniería de prompt juega un papel crucial en la optimización de la recuperación aumentada. Al crear prompts de manera coherente con el contexto y con fidelidad al contenido recuperado se puede incrementar la precisión numérica y relevancia de las respuestas generadas, mejorando hasta un 5 %, que en contextos como el ámbito clínico, donde la exactitud de los datos es esencial, puede ser significativo (58).

Además, la ingeniería de prompt permite mitigar debilidades en los enfoques de recuperación, como la capacidad de la búsqueda vectorial de equilibrar la semántica y la exactitud, ya que ambos son aspectos fundamentales para generar respuestas relevante y completas.

Generador de respuestas: *generator* (G)

Finalmente, el LLM utilizará la información aumentada para generar una respuesta coherente y precisa. El módulo generador será el encargado de producir respuestas a partir de los documentos recuperados, sintetizando el conocimiento relevante y combinándolo con la capacidad generativa del modelo.

Dado que la generación se basa en documentos verificados, el riesgo de alucinaciones se reduce significativamente. Sin embargo, el módulo generador sigue enfrentando desafíos inherentes a la calidad de la información recuperada. La integración de fuentes diversas puede dar lugar a inconsistencias o conflictos de conocimiento, lo que requiere mecanismos adicionales para evaluar la fiabilidad y coherencia de los datos antes de generar una respuesta.

2.2.2. Tipos de RAG

Los LLM están diseñados principalmente para procesar los datos en formato de texto, pero hay modelos multimodales que trabajan con otros tipos de datos, como audio o vídeo. Del mismo modo, los modelos de RAG se basaron inicialmente en estudios de texto, pero han evolucionado para abarcar modalidades como audio, vídeo, incluso multimodal, permitiendo aplicaciones como el reconocimiento de voz o análisis de video (59).

Por ejemplo, los modelos de audio extienden los principios al procesamiento del habla, permitiendo la transcripción o los asistentes de voz inteligentes. Los datos se representan mediante embeddings de modelos preentrenados y se emplean tanto para la recuperación de la información como la generación de respuestas contextuales. A inicios de 2025, el estudio de **Fang et al. (2025)** (60) evaluó audios de enseñanza simulada con un modelo RAG local, obteniendo puntuaciones relativamente altas en precisión (4.13 sobre 5.00) y lenguaje (4.37 sobre 5.00). Este modelo presentó limitaciones técnicas derivadas del tipo de dato, ya que se detectaron faltas de precisión en el reconocimiento multilingüe.

Por otro lado, también existen modelos RAG que basan su recuperación en datos extraídos de imagenes, combinando transformadores de visión con la generación aumentada para mejorar la generación de respuestas a partir de imágenes, como se comprobó en el estudio de **Raminedi et al. (2024)** (61), en el que se emplearon Vision Transformers y variantes para procesar imágenes médicas y extraer información visual relevante, empleando GPT-2 como decodificador. Este estudio superó los modelos recurrentes en la generación de informes médicos, demostrando el potencial de los sistemas RAG en la mejora de eficiencia y precisión en el diagnóstico.

Unificando ambos tipos de datos, se puede asumir que la aplicación de RAG en video presenta un desafío significativo debido a la naturaleza multimodal del contenido audiovisual, necesitando de recuperación eficiente de fragmentos de vídeo, pero también generando respuestas basadas en los metadatos de los mismos. El estudio de **Tevissen et al. (2024)** (62) permite identificar las limitaciones más relevantes de un sistema RAG multimodal, siendo la más relevante que no existen métricas estandarizadas que midan la capacidad de los LLM para seleccionar segmentos de video adecuados a partir de descripciones textuales.

El estudio de **Kresevic et al. (2024)**(2) demostró que la calidad de los datos disponibles del sistema, mayor rendimiento muestra. Concretamente, este estudio realizó una transformación progresiva de los datos, desde una simple limpieza inicial de los datos en formato textual, a la conversión de imágenes a texto y tablas en formato tabular, obteniendo una precisión significativamente superior cuánto mayor calidad mostraba la conversión,

Se puede observar, entonces, que los modelos RAG están evolucionando para convertirse en herramientas multimodales capaces de manejar todo tipo de datos. Aunque existen desafíos técnicos y metodológicos, los avances en este campo prometen mejorar significativamente sectores como el diagnóstico médico.

2.2.3. Retos y límites de RAG, direcciones de futuro

La capacidad para combinar la recuperación de información y la generación de texto ha permitido avances significativos en tareas complejas, como la generación de informes a partir de imágenes, la creación de resumenes o la respuesta de preguntas. Sin embargo, pese a su creciente evolución y aplicabilidad, los sistemas RAG enfrentan varios retos y limitaciones

técnicas, quedando potencial por explotar.

En los sistemas RAG, las respuestas incompletas o erróneas pueden deberse a diversos factores dentro de su arquitectura, y no únicamente a la falta de contenido, lo que puede resultar en respuestas sin referencias sólidas que las respalden. En la Figura 2.4 se observan los posibles fallos en la recuperación de documentos. Uno de los fallos más comunes ocurre cuando el sistema no logra identificar información relevante en los documentos recuperados, a pesar de que esta esté presente, debido a errores en la clasificación que impiden destacar dicha información como significativa. Otro posible problema surge cuando, aun habiendo detectado la documentación adecuada, el modelo de lenguaje (LLM) no la incorpora correctamente en la respuesta, afectando negativamente la calidad de la información proporcionada. En situaciones donde se manejan numerosos documentos con contenido similar, la presencia de ruido en la información recuperada puede llevar a respuestas imprecisas o confusas. Asimismo, cuando la pregunta es demasiado general, el modelo puede generar respuestas que no abordan adecuadamente la necesidad de la consulta. Finalmente, también es posible que se generen respuestas incompletas que, aunque no sean incorrectas, omitan detalles relevantes presentes en el contexto, reduciendo así la utilidad de la información ofrecida (30).

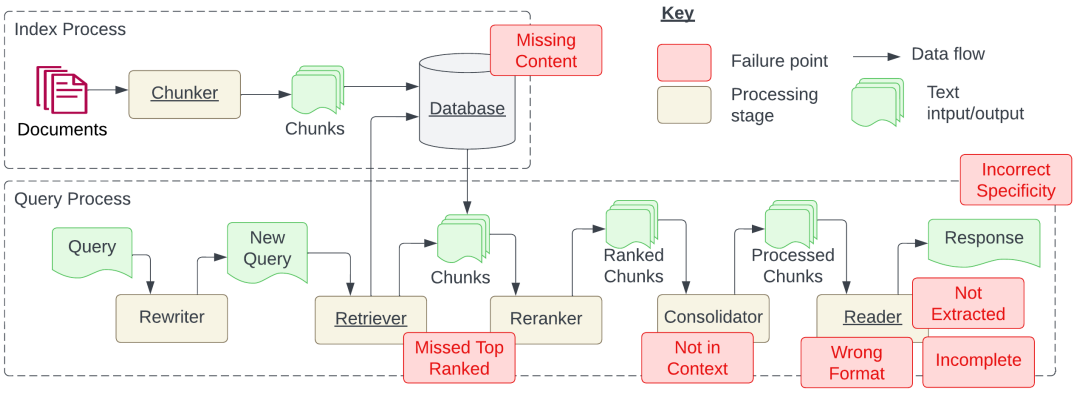


Figura 2.4: Workflow de arquitectura RAG.
La figura representa todos los puntos clave limitantes en un flujo de trabajo RAG, destacando en cuadros rojos las problemáticas que suponen, como contenido faltante en la base de datos o errores en el re-ranking del contexto recuperado. Extraído de (31).

A partir de estos errores, se derivan una serie de desafíos técnicos que impactan directamente en el rendimiento y la fiabilidad de los sistemas RAG:

- **Dependencia de la calidad de los documentos recuperados**: La precisión de los modelos RAG se basa, principalmente, en la calidad de los documentos recuperados, ya que en caso de incluir datos inexactos o incorrectos se puede producir una recuperación sesgada o alucinaciones, en que las respuestas generadas por el modelo se basan en referencias incompletas o completamente inventadas pero tomadas como reales. Esto compromete la fiabilidad de los resultados, influenciando a su aplicación práctica en ámbitos en que la precisión es fundamental (38). Del mismo modo, los modelos pueden *heredar* sesgos presentes en los documentos recuperados, siendo uno de los desafíos más críticos, ya que pese a que se han propuesto diferentes técnicas, como la recuperación justa, la completa eliminación del sesgo sigue siendo una tarea complicada.

- **Alta demanda computacional**: Uno de los principales retos de los modelos RAG radica en su alta demanda computacional, ya que requieren de dos módulos, la recuperación de información y la generación del texto. Solamente la latencia en la recuperación de documentos relevantes puede ser altamente costosa, afectando a la eficiencia del sistema si se manejan grandes cantidades de datos. Además, la creación de un repositorio de los vectores, los embeddings y la generación de respuesta impactan directamente en la capacidad del modelo RAG y la posibilidad de aplicarlo al mundo real.
- **Seguridad del sistema**: Las arquitecturas RAG presentan ciertas vulnerabilidades de seguridad, como los ataques de recuperación y los ataques generativos con el objetivo de manipular la información recuperada o la respuesta generada (63). Aun así, estos ataques requieren de condiciones específicas que pueden evitarse, como el acceso único a bases de datos locales (64).

Capítulo 3

Estado del arte

Las guías clínicas son herramientas fundamentales en la práctica médica, proporcionando recomendaciones basadas en evidencia para el diagnóstico y tratamiento de diversas patologías. Con el auge de la IA y el NLP, han surgido nuevas metodologías para mejorar el acceso y recuperación de información de estas guías.

En los últimos años, los modelos de RAG demostraron gran efectividad en la extracción y organización de conocimiento clínico, facilitando su uso en aplicaciones médicas y teniendo potencial de reducir el tiempo en la toma de decisiones por parte de los especialistas médicos.

Dada la necesidad de alta precisión en ámbitos sensibles como el diagnóstico médico, se han desarrollado variantes de RAG específicas para áreas como la biomedicina (65), hepatología (38; 2) o cirugía (66), con documentación médica, como guías médicas hepatológicas, pero no en patologías reumatológicas (2).

Un estudio reciente de febrero de 2025 realizado por **Vivani et al.** (67) integró bases de datos científicas, como PubMed y PMC, en un modelo de recuperación de información basado en RAG. En este enfoque, se empleó un módulo de recuperación en que se extrajeron fragmentos relevantes de PMC empleando representaciones TF-IDF y el modelo de **recuperación BM25**. Estos fragmentos se segmentarion y luego se transformaron en embeddings mediante el modelo **BioBERT**, calculando la similitud del coseno para determinar su relevancia. A continuación, los fragmentos recuperados, junto a la pregunta del usuario, se pasaron a un modelo de LLM para generar respuestas precisas, siguiendo instrucciones específicas para limitar la generación a la información previamente proporcionada por los fragmentos recuperados, mejorando la precisión fáctica a través de técnicas de *prompt engineering*. El estudio comparó varias configuraciones del modelo, mostrando que los modelos que incluyen BioBERT como modelo de embedding junto con modelos generativos, como GPT o LLaMA, lograron un rendimiento significativamente superior, obteniendo valores de Normalized Discounted Cumulative Gain (NDCG) notablemente más altos que las configuraciones sin el modelo generativo, destacando por su capacidad para recuperar documentos relevantes en función de la posición en la lista de resultados (68).

En el ámbito cardiovascular, **Adejumo et al. (2024)** (69) implementaron una arquitectura RAG con el objetivo de optimizar la evaluación del riesgo de ictus y guiar el tratamiento anticoagulante en pacientes con fibrilación auricular. Esta arquitectura, que combina la recuperación de información con el modelo de LLM **Llama3.1**, fue aplicada con un conjunto de **1000 notas clínicas**. Los resultados se validaron mediante la revisión manual de 200 notas por parte

de médicos expertos. Al comparar los resultados del modelo RAG con los datos estructurados, se observaron diferencias significativas en la capacidad de identificar los factores de riesgo. Por ejemplo, RAG fue capaz de identificar hipertensión en un 82.4 % frente al 26.2 % de los datos estructurados. Del mismo modo, otros factores clave fueron identificados con mayor precisión utilizando el modelo RAG. Este enfoque, al proporcionar una identificación más precisa de factores de riesg, tuvo un **impacto directo en la toma de decisiones**, particularmente en el cálculo de la puntuación CHA_2DS_2-VASc, que es crucial para evaluar el riesgo de accidente cerebrovascular en pacientes con fibrilación auricular.

En los servicios de urgencia y hospitales rurales, donde los profesionales clínicos pueden no contar con radiólogos cualificados para un análisis rápido de imágenes médicas, se presenta un desafío significativo para brindar una atención médica adecuada. Con el fin de mejorar la precisión diagnóstica, **Bani-Harouni et al. (2024)** (70) presentaron un enfoque denominado **Multi-Agent Guideline-Driven Diagnostic Assistant (MAGDA)**, integrando modelos de visión y lenguaje con guías clínicas de zero-shot para la clasificación de enfermedades raras a partir de imágenes médicas. Este enfoque se basa en la generación dinámica de modelos de visión-lenguaje sin necesidad de realizar ajustes finos (fine-tuning), facilitando su aplicación a enfermedades raras sin entrenamiento previo específico. Este **sistema multiagente** opera técnicas de razonamiento en cadena de pensamiento a través de tres componentes clave: cribado, diagnóstico y refinamiento. De este modo, se proporciona una justificación paso a paso hasta tomar la decisión diagnóstica, haciendo que el proceso sea transparente, crucial en el entorno clínico.

En medicina preoperatoria, **Ke et al., (2025)** (71) desarrollaron un modelo RAG basado en 35 **guías clínicas preoperatorias** como base de conocimiento para mejorar la precisión de las respuestas generadas por LLMs. Para ello, se emplearon herramientas como **LangChain** y **LlamaIndex** en el preprocesamiento de los documentos, fragmentándolos en chunks y almacenándolos en Pinecone como base de datos vectorial, utilizando la similitud coseno para la recuperación de información. El rendimiento se evaluó en 14 escenarios clínicos mediante la comparación de respuestas generadas por diferentes LLMs, llegando a lograr un aumento en la precisión al 91.4 % con GPT-4.0, superando el desempeño de modelos sin RAG. Además, el tiempo de respuesta se redujo de 10 minutos a 15-20 segundos, manteniendo una precisión comparable a la respuesta humana, demostrando el potencial de RAG para mejorar la eficiencia y confiabilidad de los sistemas de apoyo en el entorno médico.

Recientemente, el estudio de **Ge et al. (2024)** (38) desarrolló el modelo de LLM **LiVersa**, y evaluó la capacidad de los LLM en la interpretación de casos clínicos específicos, respondiendo preguntas sobre carcinoma hepatocelular y el Virus de la Hepatitis B. Para entrenar y optimizar LiVersa se utilizaron 30 documentos de la American Association for the Study of Liver Diseases (AASLD) incorporados mediante Azure OpenAI Cognitive Search, incluyendo guías clínicas, revisiones y estudios hepatológicos. El desempeño se evaluó comparando las respuestas por el modelo y las respuestas proporcionadas por médicos especializados, analizando tanto la exactitud de respuestas sí/no como respuestas detalladas. Mientras que las respuestas de tipo sí/no se respondieron correctamente, las preguntas detalladas mostraron 3 errores de 10 preguntas en las justificaciones, caracterizadas por la falta de documentación y por sesgos contextuales en la información proporcionada. Este estudio mostró las posibles **limitaciones en las arquitecturas RAG**, identificando la posibilidad de sesgo en la selección de documentos o la dependencia de la calidad de los documentos para proporcionar respuestas fundamentadas.

En el estudio de **Kresevic et al. (2024)** (72), se analizó el rendimiento de un modelo de LLM, específicamente **GPT-4 Turbo**, en la recuperación y evaluación de recomendaciones clínicas sobre el Virus de la Hepatitis C proporcionadas por la European Association for the Study of the Liver (EASL). En este estudio, se analizaron diferentes configuraciones experimentales, variando el reformateo y la estructura de los documentos y el uso de few-shot learning. Se plantearon 5 posibles **configuraciones**, con diferentes niveles de **procesamiento** de los documentos PDF, y se demostró que, a mayor nivel de procesamiento, mayor calidad mostraba la información extraída dentro de la documentación. De este modo, se observó que los procesos que **incrementaban la precisión del modelo** fueron la extracción de texto en formato TXT, requiriendo de una limpieza del contenido del mismo. Respecto a las imágenes y figuras, se extrajo la información contextual, mientras que en las tablas se transformaron en listas. Tras estas modificaciones, se **redujeron** drásticamente las **alucinaciones**, con un descenso de 57 a solamente 1 caso de alucinación, mientras que en las métricas cuantitativas de la similitud entre las respuestas generadas por el modelo de LLM y las respuestas proporcionadas por expertos humanos demostraron una clara mejora después de todo el preprocesado.

En el ámbito de la nefrología, **Miao et al. (2024)** (73) presentó un modelo de RAG enfocado en el manejo de la Enfermedades Renales Crónicas (ERC), empleando GPT-4 para recuperar información especializada en **guías KDIGO 2023**. Este sistema permitió respuestas más precisas, informadas con la evidencia clínica y adaptadas a la progresión de la ERC y su tratamiento farmacológico. El estudio demostró una mejora significativa en la precisión y especificidad en las respuestas en comparación al modelo sin RAG, obteniendo respuestas detalladas y alineadas con las últimas directrices de las guías clínicas, incluyendo tratamientos avanzados, a diferencia del modelo sin RAG, que solo proporcionaba respuestas más generales. Aún así, también se demostró la **dependencia a la información contextual** en la generación de las respuestas, requiriendo de integrar más guías clínicas especificas para subtipos de la ERC para abordar de forma más precisa situaciones específicas. En nefrología, donde los avances son constantes, la actualización automática de información resulta clave para evitar información obsoleta, por lo que RAG permite mejorar la precisión al extraer directamente fragmentos relevantes de artículos y guías clínicas, reduciendo el riesgo a alucinaciones.

Capítulo 4

Materiales y métodos

En este capítulo se detallan los procedimientos y herramientas utilizados para desarrollar el modelo de RAG aplicado a guías clínicas de reumatología. Se describen los pasos empleados en la adquisición y preprocesamiento de los documentos, las técnicas de fragmentación y la conversión de los archivos en formatos estructurados. Además, se especifican los algoritmos y modelos empleados para la recuperación de la información, junto a las métricas utilizadas para evaluar su rendimiento.

En la Figura 4.1 se muestra el procedimiento seguido: preprocesamiento, arquitectura RAG y evaluación.

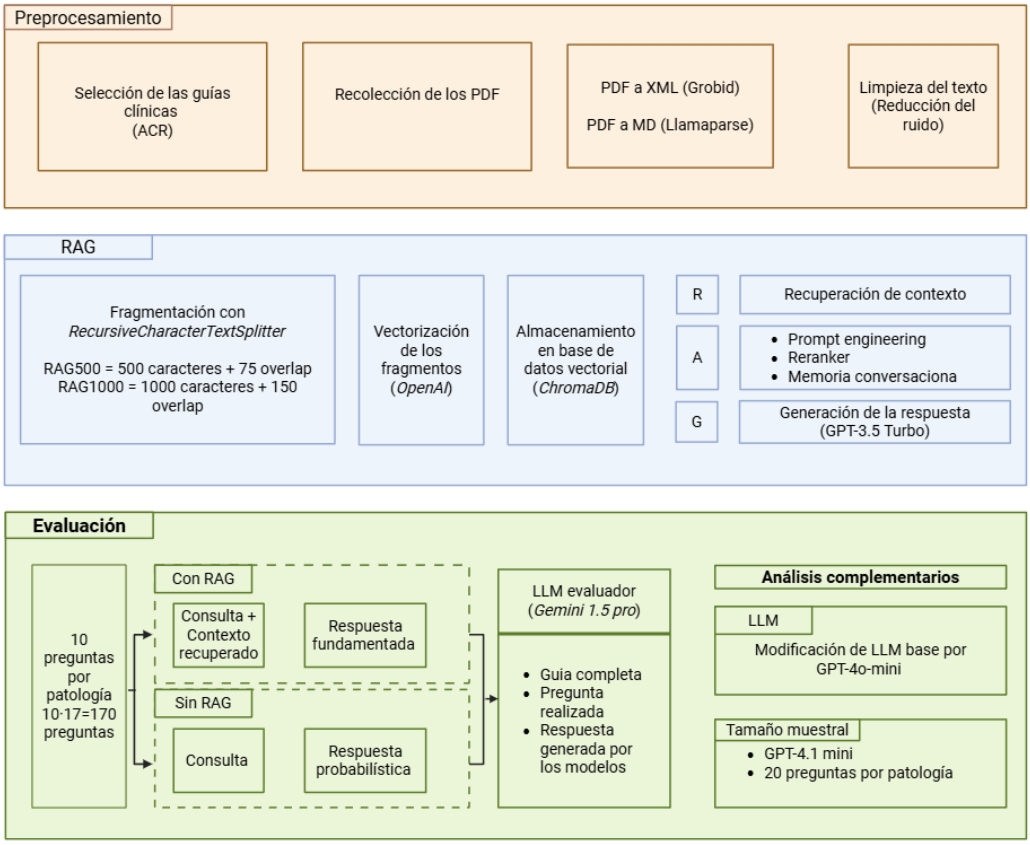


Figura 4.1: **Diagrama del flujo completo** del proceso desarrollado en el trabajo, dividido en tres fases principales: el Preprocesamiento (desde la recolección de las guías clínicas hasta la creación del vectorstore), el diseño de la Arquitectura RAG (determinando todas sus características) y la Evaluación de los modelos (realizando las comparaciones y diseñando modelos alternativos).

4.1. Materiales

En esta sección se presentan los datos y herramientas empleadas en el desarrollo del modelo. Se describe el conjunto de guías clínicas utilizadas, especificando su formato, origen y características, así como las herramientas de software y librerías utilizadas para el procesamiento de los documentos y la implementación del modelo.

4.1.1. Conjuntos de datos

Para la realización de este trabajo, se ha utilizado un conjunto de datos de documentos de guías clínicas en reumatología obtenidos de la ACR. Estos documentos engloban 17 enfermedades musculoesqueléticas, como artritis reumatoide o gota, presentando recomendaciones repartidas en 23 documentos en formato PDF.

Cada patología cuenta con un documento de recomendaciones clínicas detalladas, respaldadas por evidencia científica, junto con tablas y figuras que permiten entenderlas, en diferentes grados de complejidad.

Dado a que estas guías no siguen un formato homogéneo, requieren de un proceso de preprocesamiento inicial previo a la incorporación en el modelo RAG, implicando una extracción, una limpieza y normalización de la información. Se puede observar en Tabla 1 del Anexo 1 la información sobre las guías clínicas empleadas en el trabajo.

4.1.2. Herramientas y tecnologías

El desarrollo de este trabajo ha requerido de diversas herramientas y tecnologías para la implementación, procesamiento y almacenamiento de datos.

Respecto al lenguaje de programación y bibliotecas, se ha empleado el lenguaje de programación **Python** debido a su versatilidad y amplio abanico de bibliotecas especializadas, como pandas para la manipulación de datos o langchain para la integración del modelo con la arquitectura RAG.

Se ha empleado el modelo de lenguaje **GPT-3.5 Turbo** de OpenAI como LLM base para la recuperación aumentada de información. Se ha seleccionado este modelo por su capacidad para procesar grandes volúmenes de texto y generar respuestas coherentes en contexto de recuperación de contexto clínico. Además de su eficiencia económica, se destaca por una ventana de contexto de hasta 16.384 tokens, lo que permite incorporar fragmentos amplios de información recuperada, facilitando una mayor coherencia y fidelidad en las respuestas generadas a partir de las guías clínicas.

Para almacenar y gestionar la información extraída de los documentos se ha empleado **ChromaDB** junto con el modelo de embedding de Open AI (*text-embedding-3-large*), elegida por su eficiencia demostrada en la gestión de bases de datos vectoriales, facilitando la búsqueda semántica y la integración con modelos de lenguaje, además de ser sensible a cambios sutiles en el lenguaje, clave en textos complejos como las guías clínicas.

Tal como se introdujo anteriormente, existen múltiples frameworks especializados para el diseño de arquitecturas RAG, como LangChain, LlamaIndex o Haystack. Para este trabajo, se ha seleccionado **LangChain** por su flexibilidad, además de la integración nativa con ChromaDB como base vectorial y OpenAI GPT como LLM y su capacidad para gestionar interacciones complejas en sistemas de recuperación de información.

Los otros dos frameworks, pese a ser buenas opciones, fueron descartadas por diferentes motivos. LlamaIndex fue descartado ya que se centra en la integración de múltiples componentes en un workflow modular, pero su enfoque en la optimización de la indexación no es el adecuado para un modelo de RAG. Respecto a Haystack, se descartó porque su enfoque en pipelines híbridos y NLP tradicionales no es necesario para el modelo RAG, basado en embeddings y LLMs, donde LangChain permite una integración más directa.

Para la implementación de una interfaz web interactiva que permita mostrar el funcionamiento del modelo, se optó por utilizar **Gradio**. Gradio es una biblioteca de Python que permite crear interfaces web interactivas de manera sencilla, permitiendo cargar modelos locales o en la nube fácilmente. Además, permite personalizar el diseño y las salidas, así como ejecutar localmente, permitiendo mostrar el modelo RAG seleccionado de manera intuitiva.

4.2. Métodos

En esta sección se detallan los procedimientos de cada etapa del desarrollo del modelo, describiendo el flujo de trabajo aplicado en las etapas de preprocesamiento y la posterior implementación del modelo de recuperación de información y generación de respuesta. Asimismo, se explicarán los criterios de evaluación empleados para analizar el rendimiento del sistema.

4.2.1. Preprocesamiento de los documentos

Tal y como se ha mencionado anteriormente, las guías clínicas de ACR no mantienen una estructura uniforme, sino que contienen elementos complejos como tablas y figuras, además de constar, en ocasiones, en más de una columna de texto. Por ello, el preprocesamiento de los documentos es una fase crítica en la implementación del sistema RAG, ya que garantiza una homogeneización, estructuración y segmentación de la información contenida en las guías clínicas. Principalmente, se debe convertir el formato original de los documentos (PDF) en formatos más estructurados como XML o Markdown.

Para la extracción del contenido textual de los PDFs, se emplearon herramientas especializadas como **GROBID** y **LlamaParse**, que permitieron obtener una representación estructurada, separando secciones, títulos, párrafos e incluso tablas.

Parte de la complejidad de las guías clínicas corresponde a la información no contextual de los documentos, como las referencias del texto o los encabezados de la página. Esta información, pese a proporcionar información sobre la procedencia del texto, no proporciona un contexto sobra la información de las recomendaciones de las guías clínicas. Para mantener la mayor claridad posible dentro del texto de los documentos, se realizaron procesamientos de limpieza diferentes en cada documento considerando las capacidades de la técnica de parseo.

La herramienta **LlamaParse**, se ejecutó en un entorno en la nube mediante una clave API, en la que se obtuvo un documento Markdown como resultado del parseo. Esta herramienta es eficaz diferenciando y obteniendo las tablas en un formato Markdown, diferenciando entre las celdas tanto en filas como columnas. Sobre este formato, la limpieza se basó en una eliminación de todo contenido diferente a las tablas. El formato Markdown, pese a poseer cierta estructura, no se pudo *limpiar* eficazmente mediante código, por lo que se realizó una limpieza manual, asegurando la correcta estructura de las tablas, y estableciendo una diferenciación mediante secciones entre ellas.

Por otro lado, la herramienta de **GROBID**, pese a tener una alta capacidad en la diferenciación de columnas, tiene una carencia significativa en la extracción y estructuración de la información contenida en las tablas, sin poder diferenciar la información de celdas colindantes. Por ello, en los documentos XML/TEI obtenidos a partir de esta herramienta, se eliminaron todas las etiquetas asociadas a las tablas y figuras, así como la información que contenían. En los documentos XML/TEI, al tener una estructura más clara gracias a las diferentes etiquetas, se realizó una limpieza mediante código en que se realizaron las siguientes acciones:

- Eliminación de la sección de referencias bibliográficas.
- Limpieza de las referencias ubicadas en el texto (ubicadas gracias a la estructura de referenciado Vancouver)

- Eliminación de la sección de autores y reconocimientos.
- Eliminación de las tablas y figuras gracias a las etiquetas <table> y <figure>.

Las figuras, pese a la posibilidad de ser formateadas en una estructura tabular, no mantienen todo el contexto de la figura en ninguno de los dos formatos. Debido a que la información de las figuras es, en gran medida, la información contenida en el texto y las tablas, se determinó su eliminación.

Antes del chunking, para evitar la pérdida de información, se extrajo la información de ambos documentos y se combinaron obteniendo un documento en formato TXT por cada documento PDF original.

Con los documentos procesados, se aplicó la técnica de segmentación mediante **RecursiveCharacterTextSplitter**, ya descrita en el capítulo teórico. Esta herramienta permitió fragmentar el texto en chunks de tamaño controlado, ajustados a los límites de tokens del modelo de embeddings utilizado, preservando la coherencia semántica en cada segmento. Su uso resultó clave para evitar cortes abruptos en las frases y facilitar una vectorización eficiente de la información.

Cuando se fragmenta un documento, se corre el riesgo de romper el contexto semántico si se hace una división sin continuidad entre fragmentos. Por ejemplo, se podría romper una idea o concepto al tratarlo en dos fragmentos diferentes, pero sin continuidad entre ellos, requiriendo de una conexión entre fragmentos relacionados. La superposición o *overlap* es un parámetro clave para mantener el contexto durante el chunking, ya que permitirá al modelo encadenar los fragmentos. Con la información combinada, se especificaron los parámetros de fragmentación, con un tamaño de 500 caracteres y un overlap de 75 caracteres.

4.2.2. Representación y almacenamiento de la información

Con la información fragmentada, se procedió a la representación de la información contextual de cada fragmento en un formato numérico, convirtiendo los chunks a un formato vectorial de altas dimensiones, de manera que se mantuviera gran parte de la información contextual.

En esta fase, se empleó el modelo de embedding de OpenAI **text-embedding-large-3**, caracterizado por su relación costo-calidad, destacando entre el resto de opciones consideradas anteriormente.

Dado que los embeddings son representaciones numéricas de alta dimensión, requieren de un sistema de almacenamiento especializado que permita realizar búsquedas por similitud. Debido a su integración nativa tanto con el modelo de embedding como con el framework, se empleó el vectorstore de **ChromaDB**, una base de datos vectorial optimizada para la indexación y recuperación de embedding en entornos de recuperación aumentada de información (RAG).

ChromaDB se configura como un modo de almacenamiento persistente, de manera que los embeddings se pueden reutilizar sin necesidad de recalcularlos en cada consulta (se guardan en un directorio), evitando la necesidad de recrear la base de datos repetidamente. Cada vector generado se vinculó con los metadatos correspondientes, incluyendo el nombre original del documento, los identificadores DOI y Pubmed, así como el título del documento.

Durante el almacenamiento, se optimizan los parámetros de búsqueda para garantizar que los futuros tiempos de respuestas sean óptimos y rápidos. ChromaDB permite, gracias a su configuración, realizar consultas en tiempo real basadas en métricas de distancia (como coseno o

producto escalar), optimizando la precisión en la recuperación de fragmentos relevantes durante el proceso de recuperación.

4.2.3. Arquitectura RAG

Una vez finalizado el preprocesamiento de los documentos y la indexación en la base de datos vectorial, se procedió a implementar el sistema de RAG. Este sistema combina mecanismos de recuperación basados en similitud semántica con modelos de LLM generativos para responder preguntas abiertas mediante información contenida en las guías clínicas previamente procesadas.

El funcionamiento del sistema RAG se dividió en las siguientes fases:

- **Recuperación de fragmentos relevantes en base a una consulta:** A partir de la pregunta de entrada, el sistema convierte la consulta en un vector mediante el modelo de *embedding* de OpenAI (igual que en la creación del *Vectorstore*). Este vector será comparado con los vectores almacenados en la base de datos vectorial de ChromaDB para recuperar los fragmentos más similares semánticamente.

- **Re-ranking de los documentos recuperados:** Una vez recuperados los fragmentos más similares semánticamente, se aplicó una etapa de reordenamiento (*re-ranking*) con el objetivo de mejorar la relevancia contextual de los resultados. Para ello, se utilizó el modelo de lenguaje de *Cohere*, que permite evaluar la adecuación contextual de cada fragmento co nrespecto a la consulta original. Este paso es esencial para priorizar los fragmentos que, aunque sean similares semánticamente, presenten un mayor grado de utilidad específica para la consulta.

- **Memoria conversacional:** Para permitir interacciones más naturales y coherentes a lo largo de la conversación, se integró un sistema de memoria conversacional mediante *ConversationBufferMemory*. Este componente mantiene un historial de preguntas y respuestas previas, proporcionando contexto para generar respuestas más informadas y consistentes en diálogos sucesivos.

- **Trazabilidad con LangSmith:** Con el objetivo de facilitar el análisis, seguimiento y depuración del comportamiento del sistema, se incorporó la herramienta de LangSmith, ya que permite registrar el flujo de ejecución, visualizar las decisiones tomadas por el modelo y realizar un análisis detallado de cada paso en la cadena de razonamiento del modelo. Esto es especialmente práctico para la evaluación y optimización del sistema.

- **Generación de la respuesta:** Los fragmentos seleccionados, junto con el historial anterior, se proporcionan como contexto al modelo generativo de *GPT-3.5 Turbo*, que producirá una respuesta final basada tanto en la información recuperada como en la formulación de la pregunta original.

4.2.4. Creación de la arquitectura RAG

Para poder emplear las diferentes herramientas, se requieren distintas claves Application Programming Interface (API) para interactuar con ellas. Debido a la naturaleza sensible de las claves API, se decidió crear un documento `.env`, ya que permite gestionar las claves y

configuraciones sensibles de forma segura y desacoplada del código principal, facilitando la portabilidad del código (y crear un repositorio en Github). En dicho documento se requieren las claves API para OpenAI, que servirá como LLM de generación de respuestas para los dos modelos, para Cohere, que funcionará como reranker en el modelo RAG, y para LangSmith, que permitirá mantener la trazabilidad durante la ejecución de los modelos.

A continuación, dado que el vectorstore se creó en un directorio persistente, solo se debe cargar para ejecutar la recuperación de documentos. Esto permitirá acceder a los documentos más relevantes calculando su proximidad respecto a la consulta. Para acceder al vectorstore, se creó un script `cargar_vectorstore.py`, que permitía realizar la interacción entre la base vectorial previamente creada y el módulo de recuperación mediante una nueva función de recuperación.

Habitualmente, el mismo vectorstore funciona como recuperador de documentos, pasando los resultados directamente al LLM como contexto para la generación de respuestas. En este modelo, en cambio, se implementó un recuperador personalizado mediante un reranker de Cohere (`custom_retriever`). De este modo, el vectorstore de ChromaDB inicialmente recuperó "k=10" documentos, que luego son procesados por el reranker, filtrando únicamente aquellos 5 documentos que son más relevantes en función de su utilidad en base al contexto de la consulta.

Posteriormente, se emplea un prompt personalizado, que utiliza el historial previo, el contexto recuperado y la pregunta del usuario como entrada para el LLM, acompañado de instrucciones específicas para actuar como un asistente clínico, empleando exclusivamente la información proporcionada. Los prompts empleados se pueden consultar en la Tabla 2 del Anexo 2. Para evitar posibles alucinaciones, se especifica que, en caso de no disponer de suficiente información, el asistente debe indicarlo explícitamente (*"If you do not have enough information, say that you don't know. Do not make up information."*). Del mismo modo, es crucial que el LLM tenga un rol definido al cual adecuarse, generando respuestas más relevantes y precisas.

Finalmente, se generó una cadena RAG (`rag_chain`) que gestiona todo el flujo de trabajo: desde la recuperación de los documentos iniciales mediante el recuperador personalizado, pasando por la memoria conversacional que mantiene el contexto de la conversación, hasta la incorporación del prompt personalizado. Además, para asegurar la trazabilidad de las respuestas en base al contexto recuperado, se configuró la devolución de los documentos fuente junto con las respuestas generadas. Todo este proceso es gestionado por *LangChainTracer*, asegurando la monitorización de cada etapa para su posterior análisis utilizando LangSmith.

4.2.5. Creación del modelo generativo

Con el objetivo de comparar la eficiencia de un modelo RAG respecto a un modelo puramente generativo, se creó un modelo en que solamente se ejecutase el LLM de **GPT-3.5 Turbo** para generar las respuestas, empleando únicamente la pregunta del usuario como contexto para la generación, sin niguna información sobre las guías clínicas. De esta manera, la respuesta muestra la capacidad base del LLM para generar la respuesta empleando únicamente la información que obtuvo durante su entrenamiento, sin acceso a un contexto adicional como las guías clínicas especializadas.

El objetivo de este modelo es permitir evaluar las posibles diferencias en la generación de respuestas del modelo con RAG en base a un contexto actualizado, y medir el rendimiento en comparación con el modelo aumentado con recuperación de contexto.

Este modelo base tiene una arquitectura mucho más básica en comparación con el modelo de recuperación, ya que carece de las estructuras de recuperación, reranker o memoria, resumiendo su estructura a una llamada al LLM con un prompt personalizado. Debido a su sencillez, se destaca la necesidad de determinar correctamente un prompt claro y estructurado para que produzca una generación óptima.

Por lo tanto, el prompt tiene el objetivo de enfocar la generación de la respuesta en el ámbito clínico sin necesidad de añadir información innecesaria o fuera de contexto. También es importante reducir las posibles alucinaciones intrínsecas del modelo, incluyendo la instrucción explícita de que, si el modelo desconoce el contexto, reconozca sus limitaciones y no "invente" información (*"If you are unsure, say you don't know."*). Por otro lado, es importante que el LLM establezca un rol asistencial, que puede mejorar la calidad de la respuesta en términos de tono y coherencia.

El flujo de ejecución se basó en 5 simples pasos en todas las preguntas generadas previamente:

1. Se introduce la pregunta clínica (previamente diseñada y la misma que en el modelo RAG).
2. La pregunta se inserta en el *prompt* personalizado.
3. El *prompt* es enviado a la API del LLM (`GPT-3.5 Turbo`).
4. El modelo de LLM genera la respuesta basada solo en el conocimiento interno, sin contexto actualizado.
5. Se almacena la respuesta para su posterior análisis y evaluación.

En la Tabla 2 ubicada en el Anexo 2, se pueden observar los prompts usados en las diferentes interacciones con los LLM.

4.2.6. Evaluación

Uno de los factores clave para evaluar la precisión y el rendimiento de un sistema RAG es la comparación de sus respuestas con aquellas que proporcionarían los especialistas en reumatología. Sin embargo, debido a limitaciones de tiempo y a la dificultad de contactar con expertos clínicos durante el desarrollo de este trabajo, no ha sido posible realizar una validación externa basada en profesionales médicos humanos.

Como alternativa, se optó por una alternativa consistente en la creación de un conjunto de preguntas específicas, elaboradas a partir del contenido de las guías clínicas utilizadas. Estas preguntas permitieron evaluar la capacidad del sistema para recuperar y generar información basada en evidencia real. La evaluación se realizó mediante la comparación de las respuestas generadas por el sistema RAG (con recuperación de contexto) y las generadas por un modelo LLM sin acceso a contexto externo (sin la información de las guías).

Creación de las preguntas evaluadoras

Para la evaluación del rendimiento del sistema RAG, se agruparon los documentos en función de la patología siguiendo la clasificación proporcionada por la ACR, y se creó un conjunto de **10 preguntas evaluadoras** específicas para **cada patología**, basadas únicamente en el contenido de la documentación correspondiente. En total, se generaron **170 preguntas**, correspondientes a las **17 patologías** contempladas en el conjunto de documentos. Dichas preguntas se procesaron por el modelo RAG y por el modelo puramente generativo, y las respuestas permitirán evaluar, aunque de forma indirecta, la capacidad del sistema RAG para generar respuestas coherentes, relevantes y completas en comparación con un modelo generativo tradicional, sin contexto actualizado.

Con este objetivo, se creó un script que permitiese la carga de todas las guías clínicas mediante el vectorstore, recuperando todos los vectores y sus metadatos asociados y agrupándolos por patología gracias al campo *"Pathology"* siguiendo la clasificación de la ACR.

A continuación, se creó una función que permitiese, mediante un prompt personalizado y un modelo de LLM (GPT-3.5 turbo), escribiese 10 preguntas de cada patología, empleando únicamente la información contenida dentro de la guía clínica correspondiente. Al emplear un LLM, se debe tener en cuenta el límite de tokens disponibles en una misma solicitud, creando una limitación para aplicar una guía clínica completa de una sola vez. Para mitigar esta problemática, se fragmentó la guía clínica en fragmentos más pequeños (chunks diferentes a los generados en el vectorstore) manteniendo el límite de la API del LLM, y se generaron diferentes preguntas en base al contenido de cada fragmento.

Este proceso generó múltiples preguntas por fragmento, lo que resultara en un conjunto de preguntas que será filtrado y procesado, eliminando las preguntas duplicadas y seleccionando de manera aleatoria 10 preguntas de entre todas las generadas por el LLM. Al elegir las preguntas de forma aleatoria, se garantiza una representación diversa de toda la información contenida en la guía clínica de la patología, evitando que todas las preguntas provengan de un solo fragmento en específico.

Generación de las respuestas

Cada modelo, tanto el modelo generativo (LLM GPT-3.5 Turbo, **Generative**) como el modelo RAG (**RAG500**), recibió las preguntas previamente generadas y devolvió una respuesta textual que fue almacenada de manera estructurada para su posterior análisis. En ambos modelos, el LLM empleado fue GPT-3.5 Turbo.

En el caso del modelo generativo, dada su naturaleza más directa, la respuesta se obtuvo mediante una consulta al LLM, sin ningún paso intermedio adicional.

Por el contrario, el modelo RAG requiere de ejecutar toda la cadena de recuperación aumentada: recuperación de documentos relevante, reranking mediante Cohere y procesamiento de la memoria conversacional. Además, para cada respuesta generada, se almacenó el contexto (los chunks recuperados) utilizados por el modelo, lo que permite una trazabilidad completa del origen de la información, y a su vez, permitirá evaluar el rendimiento de la recuperación.

Es importante señalar que la API del re-ranker de Cohere está sujeta a limitaciones en la cantidad de solicitudes permitidas por minuto. Para evitar interrupciones durante la ejecución, se implementó una pausa automática, ralentizando el proceso pero estabilizando el sistema.

El procedimiento se llevó a cabo iterando por patología y por pregunta. Para cada combinación, se generaron y almacenaron las respuestas de los distintos modelos. Finalmente, los resultados se exportaron en formato JSON, generando **dos archivos por patología** con las preguntas y, en el caso del modelo generativo, la respuesta, o en el modelo RAG, la respuesta junto al contexto recuperado empleado.

Análisis estadístico. Comparación de ambos modelos

Con el objetivo de comparar el rendimiento de los diferentes enfoques propuestos, se realizó un análisis estadístico basado en métricas tanto cualitativas como cuantitativas. Este análisis, permite evaluar la calidad de las respuestas proporcionadas por cada modelo, identificando las fortalezas y debilidades en aspectos clave.

El análisis comparativo entre los modelos se llevó a cabo a partir de las respuestas generadas para cada patología y preguntas. Para ello, se emplearon métricas de evaluación cualitativas y cuantitativas con el objetivo de valorar el rendimiento de cada enfoque. En concreto, se evaluaron las siguientes métricas, en escalas del 1 al 10:

- **Fidelidad al contexto (faithfulness)**: Evalúa si, en el modelo RAG, la respuesta se basa exclusivamente en el contexto recuperado y no inventa o añade información adicional, incluso si esa información es verdadera. Esta métrica permite identificar alucinaciones dentro del contexto limitado, en que una alta puntuación indica la máxima fidelidad posible.
- **Relevancia (relevance)**: Evalúa qué tan bien la respuesta se enfoca y responde la pregunta clínica planteada. Cuanta mayor puntuación, más específica, directa y centrada es la respuesta, evitando divagaciones o ambigüedad.
- **Precisión factual (accuracy)**: Evalúa si la respuesta es verdadera y correcta, comparada con toda la guía clínica (no solo el contexto recuperado), determinando la veracidad de la respuesta completa y asegurando que no haya errores médicos o clínicos.
- **Completitud (completeness)**: Evalúa hasta qué punto la respuesta cubre todos los aspectos relevantes de la pregunta. Se subdivide en dos niveles:
  - **Relativa** (Given Retrieval): En el modelo RAG, mide la completitud de la respuesta teniendo en cuenta únicamente el contexto recuperado. Permite valorar la capacidad del modelo para aprovechar al máximo la información recuperada.
  - **Total** (Overall): Mide la completitud considerando el conocimiento completo de la guía clínica, reflejando la capacidad de la respuesta de abordar todos los elementos clínicamente relevantes requeridos por la pregunta.
- **Concisión (conciseness)**: Evalúa si la respuesta es breve, clara y libre de redundancias o contenido innecesario, comunicando la información de forma efectiva.

Además de la evaluación individual mediante métricas, también se solicitó al LLM evaluador que realizara un juicio global sobre cuál de las dos respuestas era superior en términos generales. Para ello, debía integrar todas las métricas evaluadas, pudiendo determinar si una respuesta

destacaba claramente sobre la otra, seleccionando entre el *modelo A* o el *modelo B*, o si ambas eran similares en calidad, categorizándolas como *Comparable*.

Cabe destacar que, mientras se utilizó el modelo de LLM de GPT-3.5 Turbo para la generación de las respuestas en ambos modelos, se empleó el modelo de **Gemini AI 1.5 Pro** como LLM de evaluación. Esta elección evita el sesgo de autoevaluación que podría surgir si el mismo modelo generara y evaluara las respuestas, lo que comprometería la objetividad y confiabilidad del análisis.

Además, los distintos LLM tienen arquitectura y capacidades propias, permitiendo enriquecer la evaluación comparando, no solo las respuestas, sino también como otro modelo de lenguaje interpreta y valora las respuestas según sus propios criterios de evaluación, como la fidelidad, relevancia o precisión.

Selección de tests estadísticos: Binomial y Wilcoxon

Para poder evaluar el rendimiento del modelo RAG, se realizaron comparaciones estadísticas, realizando **comparaciones 2 a 2** entre los modelos permitiendo que, en términos generales, el LLM evaluador permita escoger entre las respuestas con más calidad.

Las comparaciones entre los modelos son esenciales para evaluar de manera detallada las diferencias y similitudes en el rendimiento de cada modelo, permitiendo comparar de manera más precisa, identificando las fortalezas y debilidades de cada uno. De esta manera, se permite la comprensión más profunda de aspectos específicos como la fidelidad y la relevancia, aislando las diferencias entre modelos sin interferencias de otros factores.

El análisis estadístico se realizó mediante dos test estadísticos, el test Binomial y el test de Wilcoxon.

El objetivo de la **Prueba Binomial** se empleó para determinar si, en las comparaciones entre dos de los modelos, la selección de uno sobre el otro es estadísticamente significativa, analizando si la superioridad de un modelo sobre otro se debe a una tendencia real y no al azar. Dadas a sus características, el Test Binomial es apropiado cuando existen dos posibles resultados (A o B), basado en la diferencia significativa entre los modelos:

- **Hipótesis nula ($\mathbf{H_0}$):** No hay diferencia significativa entre los modelos (la probabilidad de que cualquiera de ellos sea mejor es 0.5).
- **Hipótesis alterna ($\mathbf{H_a}$):** Existe una desviación significativa de la probabilidad del 50 %, indicando que un modelo tiene mayor probabilidad de ser elegido.

En las situaciones en que Gemini no pudo decidir entre ambas opciones y las calificó como Comparable, se eliminaron dichas respuestas, dejando únicamente las variables dicotómicas.

La **Prueba de Wilcoxon** se utilizó para comparar las puntuaciones de las métricas (continuas, de una escala del 1 al 10) obtenidas en cada modelo, con el objetivo de evaluar si existen diferencias significativas entre ellos. Este test no es paramétrico, por lo que no asume que hay una distribución normal de los datos, por lo que es especialmente útil en escenarios donde la distribución puede ser asimétrica o tiene valores atípicos. Debido a que las métricas se basan en las mismas preguntas en base a modelos diferentes, se seleccionó el Test de rangos con signo de Wilcoxon.

- **Hipótesis nula (H_0):** No existen diferencias en la mediana de las métricas entre los modelos.
- **Hipótesis alterna (H_a):** Existen diferencias significativas en las medianas de las métricas entre el par de modelos evaluados.

Optimización del modelo. RAG1000

Tras analizar las diferencias entre el modelo generativo y el modelo RAG, se observó que el modelo RAG con chunks de 500 caracteres ha mostrado ciertas limitaciones en la recuperación de fragmentos de información relevantes. Esto se debe a que, con la longitud de 500 caracteres, los fragmentos recuperados pueden resultar incompletos o descontextualizados, especialmente en guías clínicas donde la información clave está repartida en párrafos de diferentes longitudes.

Para mitigar esta limitación, se desarrolló un nuevo modelo RAG, basado en un Vectorstore que emplea fragmentos de 1000 caracteres, con un solapamiento de 150 caracteres entre ellos (**RAG1000**). Al incrementar el tamaño de los chunks, se mejora la probabilidad de que los fragmentos recuperados contengan secciones clínicas más completas, favoreciendo la generación de respuestas más precisas y coherentes.

Este nuevo modelo se evaluó con la misma metodología que los anteriores, utilizando el mismo conjunto de preguntas y métricas con el objetivo de comparar el impacto de los chunks más grandes en la calidad de las respuestas generadas.

4.2.7. Análisis complementarios: LLM y tamaño muestral

Para complementar los hallazgos del análisis principal y explorar el efecto de las variables en el rendmiento de los sistemas, se llevaron a cabo dos análisis adicionales centrados en el modelo generativo usado y el tamaño muestral.

Estudio del efecto de GPT-4o-mini

Con el objetivo de analizar el impacto del modelo generativo en la arquitectura RAG, se llevó a cabo un segundo análisis, replicando el diseño anterior con los tres modelos (Generative, RAG500 y RAG1000), el mismo conjunto de preguntas y sistema de evaluación, pero reemplazando el modelo generativo base GPT-3.5 Turbo por el modelo **GPT-4o-mini**.

Este cambio permitió aislar el impacto que tiene el componente generativo del sistema sobre la calidad de las respuestas, tanto en el modelo generativo puro como con los modelos RAG. Es decir, permitió analizar si los beneficios observados anteriormente en los modelos RAG se mantuvieron al mejorar la capacidad generativa del LLM, o si el nuevo modelo redujo la diferencia entre ambos enfoques. Este análisis resultó crucial para determinar hasta que punto la arquitectura RAG aportó valor añadido en caso de disponer de un modelo generativo más potente.

Efecto del incremento del tamaño muestral

Tras analizar los resultados obtenidos en los dos primeros análisis, se diseñó un tercer análisis con dos objetivos principales: evaluar un LLM más avanzado que los dos anteriores y, al mismo tiempo, mejorar la potencia estadística del estudio.

Para ello, se utilizó el modelo **GPT-4.1 mini**, una versión más potente y robusta que GPT-4o-mini, y al mismo tiempo, se duplicó el tamaño muestral, pasando de 10 a **20 preguntas por patología**, incrementando el total de preguntas a 340. De este modo, se permitió una evaluación más representativa y redujo el riesgo a cometer errores de tipo II (no detectar diferencias significativas reales entre los modelos analizados).

Además, al emplear un LLM más avanzado, se facilitaba la generación de respuestas con una estructura más clara y con mayor capacidad de síntesis, lo que podría influir en la capacidad discriminatoria del evaluador (Gemini) entre el modelo generativo base y los modelos RAG.

4.3. Integración en una interfaz web

El despliegue de modelos RAG a través de una interfaz web es fundamental para permitir la accesibilidad de las capacidades del modelo a usuarios no técnicos. En este contexto, **Gradio** (74) se convierte en una herramienta ideal por su simplicidad, flexibilidad y, sobre todo, su capacidad de integración directa con modelos implementados en Python.

En este trabajo, se ha desarrollado una interfaz web utilizando Gradio para integrar el modelo **RAG1000**, ejecutando el workflow de la cadena de recuperación aumentada con capacidades de *re-ranking* y memoria conversacional (al integrar memoria en la cadena, no es necesario hacerlo en el modelo de Gradio).

Las principales características de la interfaz implementada son:

- **Cadena de recuperación aumentada:** Se invoca el modelo RAG mediante la función central de la cadena (`rag_chain1000`), ejecutando el workflow configurado (búsqueda de documentos relevantes, aplica el re-ranking de Cohere y genera la respuesta, almacenándola en el historial). Esta cadena también incluye trazabilidad mediante LangSmith y conservación del contexto a lo largo de la sesión.

- **Entrada del usuario:** Se ha definido una caja de texto (`Textbox`) que permite al usuario introducir las preguntas. Este componente está etiquetado como *Äsk a question about clinical guidelines"* e incluye un ejemplo para guiar al usuario en la formulación de su pregunta.

- **Respuesta del modelo:** La respuesta generada por el modelo se muestra en un segundo Textbox, etiquetado como `"Model's Answer"`.

- **Fuentes utilizadas:** En paralelo, se muestra un tercer `Textbox` con los metadatos de los documentos utilizados para generar la respuesta. Este componente está etiquetado como `"Sources Used"` y contiene información relevante como el título del documento, año de publicación, DOI, identificador de PubMed y el archivo de origen. De esta forma, el usuario puede verificar la trazabilidad de la información proporcionada.

En la Figura 4.2, se puede observar la interfaz web con una pregunta de ejemplo (*Is urate-lowering therapy (ULT) recommended for patients with asymptomatic hiperuricemia?*)

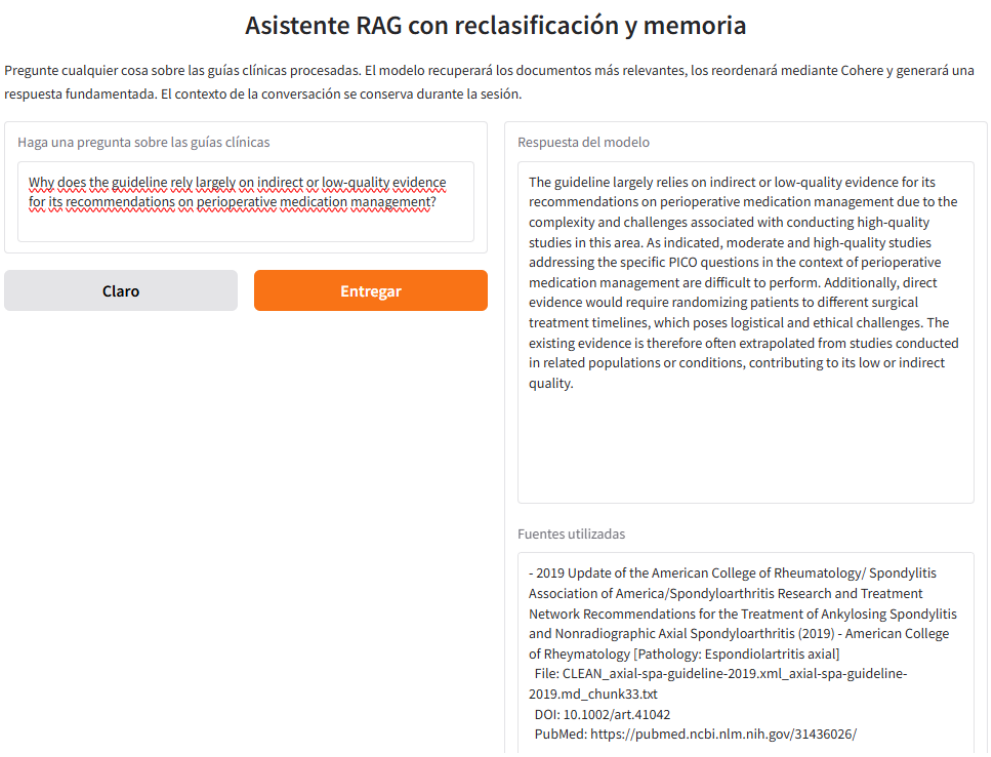


Figura 4.2: Interfaz web de Gradio con una pregunta de ejemplo.

Capítulo 5

Resultados

5.1. Introducción

En este capítulo se presentan los resultados obtenidos a partir de la evaluación de los modelos implementados. Se incluyen tanto las métricas cuantitativas empleadas para la comparación del rendimiento de cada modelo, como los análisis cualitativos que permitieron interpretar las métricas en el contexto clínico. Estos resultados permiten valorar el impacto de las distintas configuraciones del sistema RAG, tanto entre ellas, como frente a un modelo generativo base.

El conjunto completo de preguntas empleadas en la evaluación se incluye en el archivo `preguntas_evaluadoras.xlsx`, que acompaña a este trabajo como material adicional.

5.2. Análisis original. Comparación de los modelos

5.2.1. Preferencias por comparación

Durante la evaluación realizada por Gemini 1.5 pro, se solicitó que en la comparación de modelos se escogiese entre el modelo A, el modelo B o si era Comparable. En la Tabla 5.1 se pude observar un ejemplo comparativo de una respuesta generada por RAG y una respuesta generada por el modelo generativo.

La Figura 5.1 muestra un histograma que representa las proporciones en las que el modelo evaluador seleccionó un modelo sobre le otro (o comparable), ofreciendo una visión general de las preferencias observadas entre las comparaciones de las respuestas.

How is the choice between colchicine, NSAIDs, and glucocorticoids determined in the management of gout flares?	
Model	**Response**
RAG500	The choice between colchicine, NSAIDs, and glucocorticoids in the management of gout flares should be determined based on patient factors and preferences. According to the 2020 American College of Rheumatology Guideline, oral colchicine, NSAIDs, or glucocorticoids (oral, intraarticular, or intramuscular) are recommended as first-line therapy for gout flares. Patients and healthcare providers should consider factors such as effectiveness, tolerability, contraindications, and patient preferences when selecting the most appropriate medication for managing gout flares.
Generative	The choice between colchicine, NSAIDs, and glucocorticoids in the management of gout flares is typically determined based on factors such as the patient's medical history, comorbidities, contraindications, and preferences. Colchicine is often recommended as a first-line treatment for gout flares, particularly in patients with contraindications to NSAIDs or glucocorticoids. NSAIDs are also commonly used for gout flares, but caution is advised in patients with renal insufficiency, peptic ulcer disease, or cardiovascular disease. Glucocorticoids may be considered for patients who cannot tolerate colchicine or NSAIDs, or in cases of severe gout flares. Ultimately, the choice of medication should be individualized based on the patient's specific circumstances and needs.

Cuadro 5.1: Comparación de respuestas generadas por RAG500 y el modelo Generative.

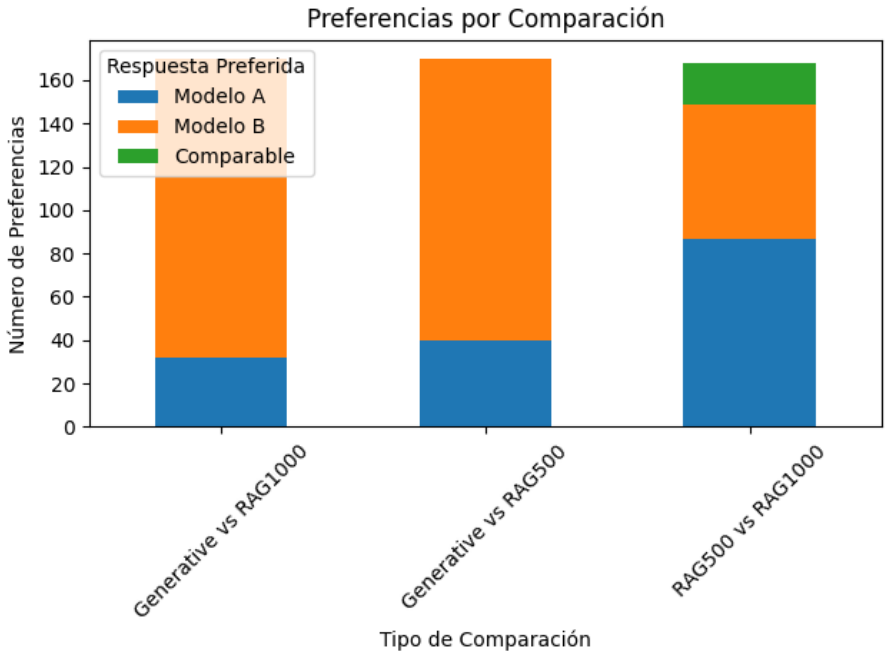


Figura 5.1: Preferencias por comparación entre los modelos generativos y RAG.

En la **primera comparación**, RAG500 fue claramente superior respecto al modelo generativo, obteniendo 130 de las 170 preguntas, mientras que el modelo generativo fue solo escogido en 40 preguntas.

En la **segunda comparación**, RAG1000 también fue superior al modelo generativo, siendo escogido 138 veces (en contraposición a las 32 veces que fue escogido el modelo generativo).

Finalmente, en la comparación **entre** los **modelos RAG**, el modelo RAG500 fue ligeramente superior, siendo escogido 87 veces, mientras que el modelo RAG1000 fue escogido 62 veces. En esta comparación, 21 de las respuestas fueron consideradas comparables para ambos modelos.

Para analizar si la selección de un modelo respecto al otro es significativa, se realiza la prueba Binomial, considerando únicamente los valores de A o B como posibles, descartando aquellos resultados categorizados como Comparables. En la Tabla 5.2 se puede observar el resultado del análisis estadístico.

Comparación	Preferencias A	Preferencias B	p-valor
Generative vs RAG500	40	130	$2{,}63 \times 10^{-8}$*
Generative vs RAG1000	32	138	$6{,}90 \times 10^{-13}$*
RAG500 vs RAG1000	87	62	$4{,}89 \times 10^{-2}$*

Cuadro 5.2: Resultados de las comparaciones pareadas entre modelos en cuanto a preferencias de respuesta, con sus respectivos valores de significancia estadística (p-valor).

* p-valor significativo con $\alpha = 0{,}05$

5.2.2. Análisis de las métricas de los modelos

Generativo VS RAG500

El objetivo de la comparación entre el modelo generativo base y el modelo RAG fue identificar si el uso de recuperación de contexto con 500 caracteres mejora aspectos clave en la generación de las respuestas.

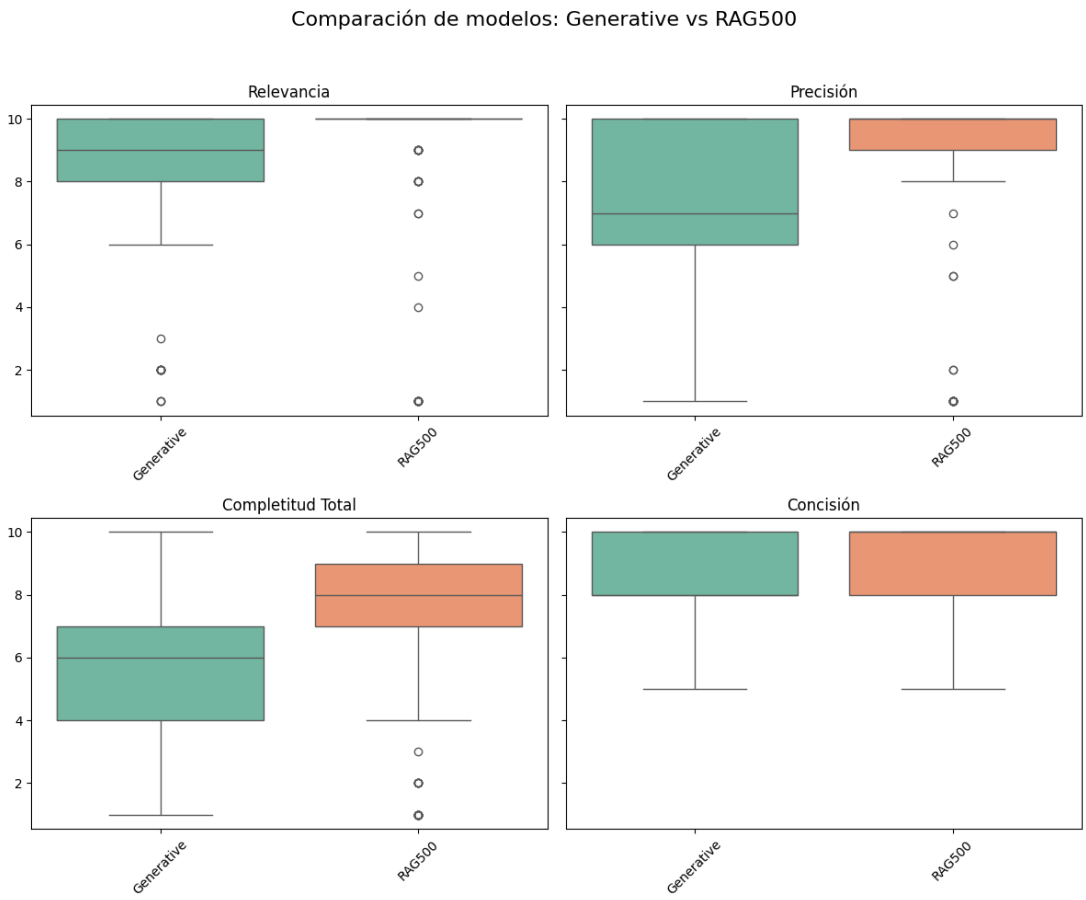


Figura 5.2: Comparación de las métricas de evaluación del LLM base vs el modelo RAG500.

En la Figura 5.2 se presenta una comparación visual mediante diagramas de caja (boxplots) entre el modelo generativo (Generative) y el modelo RAG500, evaluando 4 métricas. Además, en la Tabla 5.3 se pueden observar las medias, medianas y el rango intercuartílico Q1-Q3 resultantes de la comparación de los modelos Generative y RAG500.

Cuadro 5.3: Estadísticos descriptivos por modelo para la comparación Generative vs RAG500.

Métrica	Estadístico	Generative	RAG500
Fidelidad	Media		9.918
	Mediana		10.000
	Q1-Q3		10.000 - 10.000
Relevancia	Media	8.359	8.665
	Mediana	9.000	10.000
	Q1-Q3	8.000 - 10.000	10.000 - 10.000
Precisión	Media	6.694	8.600
	Mediana	7.000	10.000
	Q1-Q3	6.000 - 10.000	9.000 - 10.000
Completitud (parcial)	Media		8.959
	Mediana		10.000
	Q1-Q3		10.000 - 10.000
Completitud (total)	Media	5.471	7.218
	Mediana	6.000	8.000
	Q1-Q3	4.000 - 7.000	7.000 - 9.000
Concisión	Media	8.376	9.165
	Mediana	8.000	10.000
	Q1-Q3	8.000 - 10.000	8.000 - 10.000

- **Relevancia:** Ambos modelos tienen una puntuación alta, pero el modelo Generativo presenta mayor dispersión general (menor consistencia), aunque el modelo RAG tiene ciertos outliers. El modelo RAG500 muestra una mediana perfecta (10), superior al modelo generativo (9).
- **Precisión:** RAG tiene una mediana más alta y una dispersión mucho menor, agrupándose entorno a los valores 9-10, mientras que Generative tiene una variabilidad elevada, sugiriendo menos precisión, con una mediana de 7.
- **Completitd (total):** Generative es inferior a RAG500, con una mediana inferior y mayor dispersión, teniendo valores centrados entre 4-7 mientras que en RAG500 se mantienen sobre 7-9.
- **Concisión:** Las medianas de ambos modelos son bastante elevadas y parecidas, con una distribución estrecha. No parece haber diferencias importantes que se puedan visualizar gráficamente.

Generativo VS RAG1000

El objetivo de la comparación entre el modelo generativo base y el modelo RAG1000 es identificar si el uso de recuperación de contexto con 1000 caracteres por fragmento mejora aspectos clave en la generación de las respuestas.

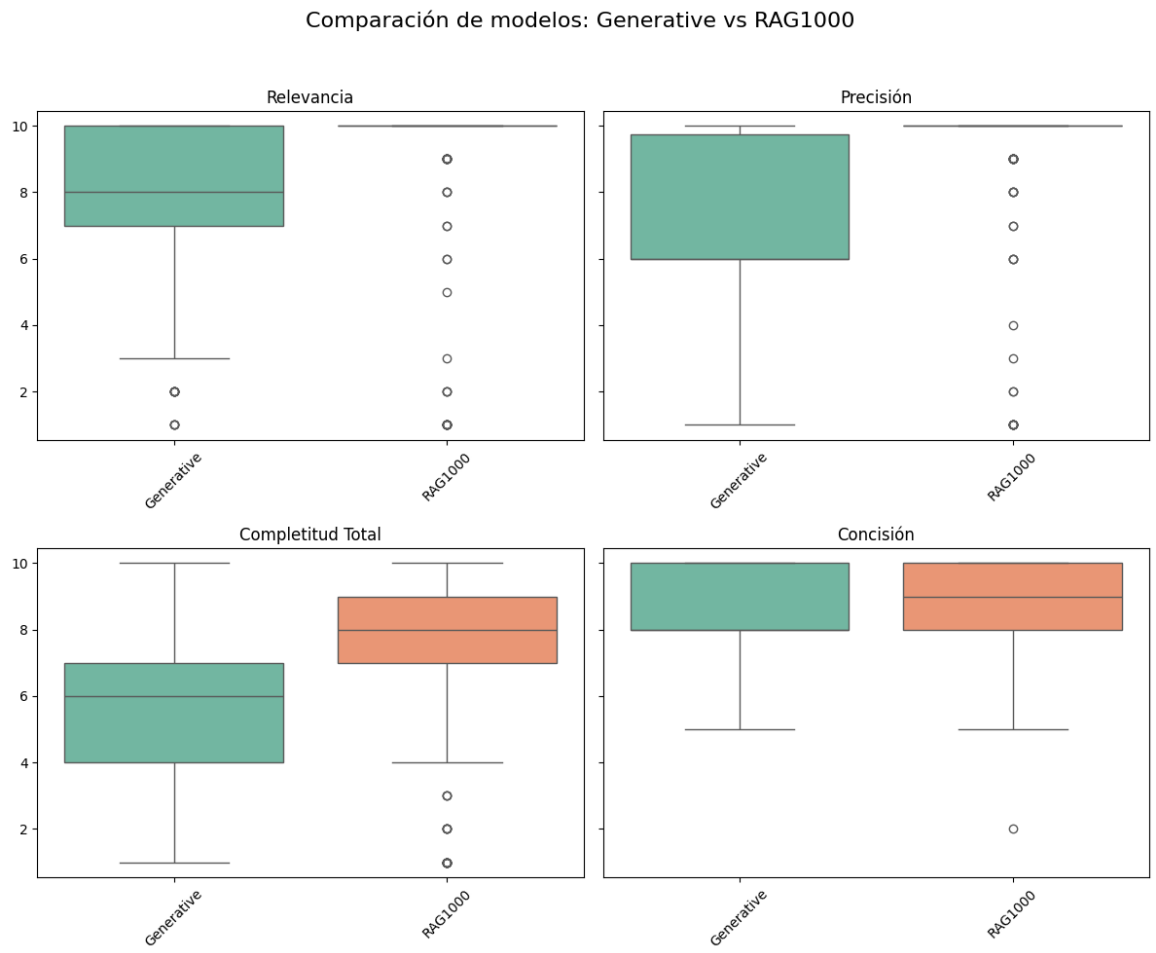


Figura 5.3: Comparación de las métricas de evaluación del LLM base vs el modelo RAG1000

Cuadro 5.4: Estadísticos descriptivos por modelo para la comparación Generative vs RAG1000

Métrica	Estadístico	Generative	RAG1000
Fidelidad	Media		9.729
	Mediana		10.000
	Q1-Q3		10.000 - 10.000
Relevancia	Media	8.129	8.935
	Mediana	8.000	10.000
	Q1-Q3	7.000 - 10.000	10.000 - 10.000
Precisión	Media	6.429	9.106
	Mediana	6.000	10.000
	Q1-Q3	6.000 - 9.750	10.000 - 10.000
Completitud (parcial)	Media		9.306
	Mediana		10.000
	Q1-Q3		10.000 - 10.000
Completitud (total)	Media	4.988	7.271
	Mediana	6.000	8.000
	Q1-Q3	4.000 - 7.000	7.000 - 9.000
Concisión	Media	8.406	8.665
	Mediana	8.000	9.000
	Q1-Q3	8.000 - 10.000	8.000 - 10.000

En la Figura 5.3 se presenta una comparación visual entre el modelo LLM generativo (Generative) y el modelo RAG1000, evaluando las 4 métricas. Además, en la Tabla 5.4 se puede observar el resumen estadístico de esta comparación.

- **Relevancia:** El modelo RAG1000 muestra una mediana perfecta (10), pero el modelo Generative tiene más dispersión y outliers con puntuaciones bajas, con una mediana de 8 e intervalo intercuartílico de 7-10.
- **Precisión:** Es similar a relevance, con RAG1000 concentrado en 10, pero Generative vuelve a tener más dispersión, con un rango intercuartílico de 6-9.75.
- **Completitud (total):** Ambos modelos son más dispersos, pero RAG1000 tiene una mediana superior (8) al modelo Generative (6). Ambos modelos muestran outliers en valores bajos.
- **Concisión:** Ambos modelos parecen tener una distribución similar, con medianas altas (entre valores 8-9), con outliers en valores bajos en el modelo RAG1000.

RAG500 VS RAG1000

El objetivo de la comparación entre el modelo RAG500 y el modelo RAG1000 es identificar si el uso de recuperación de contexto con 1000 caracteres respecto a la recuperación de contexto con 500 caracteres mejora aspectos clave en la generación de las respuestas.

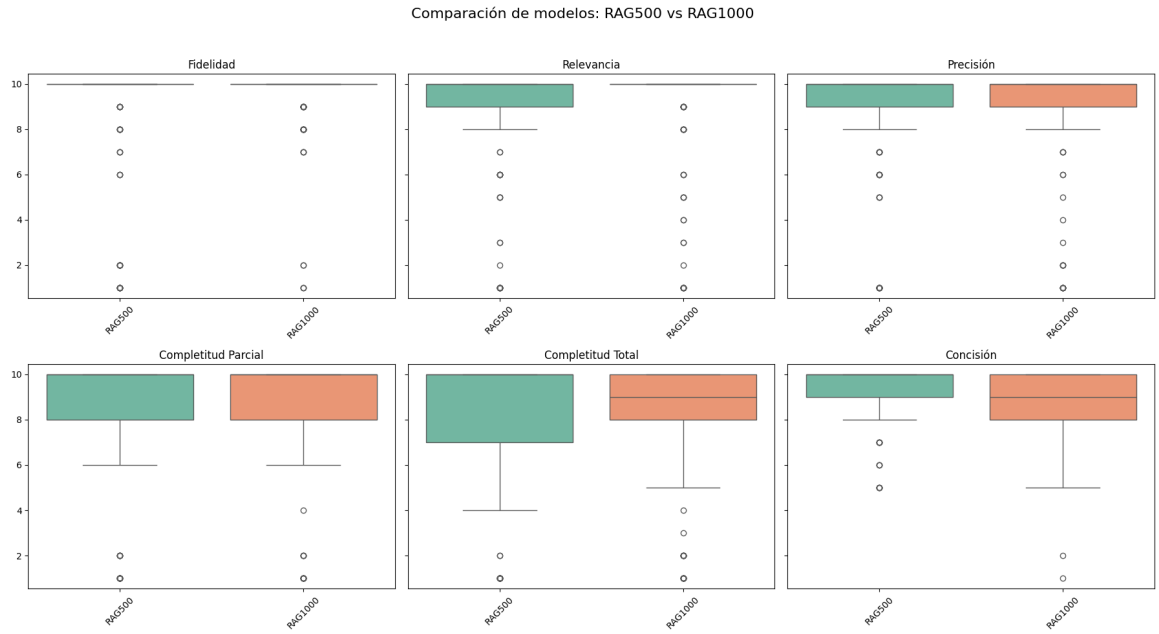


Figura 5.4: Comparación de las métricas de evaluación entre modelos RAG

En la Figura 5.4 se presenta una comparación visual entre ambos modelos RAG, mientras que en la Tabla 5.5 se observa el resumen estadístico de la comparación. En esta comparación, además de estar evaluando las 4 métricas anteriores, se evaluaron también Fidelidad y Completitud (parcial), ya que en ambos modelos se recuperó información que se empleó como contexto.

Cuadro 5.5: Estadísticos descriptivos por modelo para la comparación RAG500 vs RAG1000

Métrica	Estadístico	RAG500	RAG1000
Fidelidad	Media	8.641	9.494
	Mediana	10.000	10.000
	Q1-Q3	10.000 - 10.000	10.000 - 10.000
Relevancia	Media	8.359	8.882
	Mediana	10.000	10.000
	Q1-Q3	9.000 - 10.000	10.000 - 10.000
Precisión	Media	8.620	8.899
	Mediana	10.000	10.000
	Q1-Q3	9.000 - 10.000	9.000 - 10.000
Completitud (parcial)	Media	8.212	8.643
	Mediana	10.000	10.000
	Q1-Q3	8.000 - 10.000	8.000 - 10.000
Completitud (total)	Media	7.870	8.082
	Mediana	10.000	9.000
	Q1-Q3	7.000 - 10.000	8.000 - 10.000
Concisión	Media	9.412	8.781
	Mediana	10.000	9.000
	Q1-Q3	9.000 - 10.000	8.000 - 10.000

En consecuencia, se observan las siguientes diferencias:

- **Fidelidad:** Ambos modelos tienen una mediana perfecta (10), con outliers en los valores más bajos. La media de RAG500 es sensiblemente inferior a la media de RAG1000.
- **Relevancia:** El modelo RAG1000 muestra una mediana perfecta (10), pero el modelo RAG500 tiene más dispersión y outliers con puntuaciones bajas (rango intercuartílico de 9-10).
- **Precisión:** Ambos valores tienen una mediana considerablemente alta (10), pero RAG1000 parece tener más outliers.
- **Completitud (parcial):** Ambos modelos tienen valores altos, con puntuaciones entre 8 y 10, sin apreciarse diferencias. La media del modelo RAG500 es sensiblemente inferior al modelo RAG1000.
- **Completitud (total):** RAG500 tiene una mediana ligeramente más elevada (10), pero RAG1000 parece mostrar menor dispersión (rango intercuartílico de 8-10).
- **Concisión:** Ambos modelos tienen medianas elevadas, pero RAG500 muestra una mediana superior (10) mientras que en RAG1000 se observa mayor dispersión y mediana menor (9), así como outliers con puntuaciones más bajas.

5.2.3. Análisis estadístico

Para determinar si hay diferencias significativas entre los modelos, se realizaron pruebas de Wilcoxon entre las métricas de los modelos. La métrica de Fidelidad y Completitud parcial no se pueden analizar en la comparación con el modelo Generative, debido a que no contiene un contexto recuperado.

Comparación	Métrica	p-valor
Generative vs RAG500	Relevancia	3.23e-02*
Generative vs RAG500	Precisión	8.57e-07*
Generative vs RAG500	Completitud (total)	8.77e-08*
Generative vs RAG500	Concisión	4.25e-07*
Generative vs RAG1000	Relevancia	7.45e-05*
Generative vs RAG1000	Precisión	8.42e-13*
Generative vs RAG1000	Completitud (total)	7.78e-14*
Generative vs RAG1000	Concisión	6.50e-02
RAG500 vs RAG1000	Fidelidad	3.41e-03*
RAG500 vs RAG1000	Relevancia	7.15e-02
RAG500 vs RAG1000	Precisión	2.14e-01
RAG500 vs RAG1000	Completitud (parcial)	4.12e-01
RAG500 vs RAG1000	Completitud (total)	5.60e-01
RAG500 vs RAG1000	Concisión	2.54e-06*

Cuadro 5.6: **Resultados de la prueba de Wilcoxon** para comparación de métricas entre modelos.
Esta tabla resume los resultados de la prueba estadística de Wilcoxon aplicada a distintas métricas evaluativas entre los modelos Baseline, RAG500 y RAG1000. El símbolo (*) indica diferencias estadísticamente significativas con un umbral de $\alpha = 0{,}05$. La completitud se divide en dos conceptos: basada en el contexto proporcionado por la recuperación (parcial) y en el contexto global de la guía clínica (total).

Como se puede observar en la Tabla 5.6, se muestran diferencias significativas en algunas métricas de la valoración.

- En la comparación entre **Generative y RAG500**, se observaron diferencias significativas en todas las métricas evaluadas.
- Entre los modelos **Generative y RAG1000** las diferencias significativas se mantuvieron en casi todas las métricas, mientras que en Concisión, las diferencias no fueron significativas, pero mantuvieron un p-valor muy cercano a la significancia.
- Finalmente, **entre los modelos RAG**, únicamente se observaron diferencias significativas entre Fidelidad y Concisión, pero el resto de métricas no mostraron tanta diferencia. Cabe destacar que, en la métrica de Relevancia, se mostró un p-valor bastante bajo, cercano a la significancia.

5.3. Evaluación del LLM GPT-4o-mini

El objetivo de emplear el nuevo modelo de GPT-4o-mini es analizar si, mejorando las capacidades del LLM, causa diferencias importantes en la generación de las respuestas.

5.3.1. Preferencias por comparación

Así como en el análisis original, se solicitó al LLM evaluador que, mediante las métricas obtenidas de los modelos, escogiese, para cada pregunta, uno de los modelos que, en términos generales, obtuvo una mejor respuesta.

En la Figura 5.5, se pueden observar un histograma, donde se muestran, para cada comparación, cuál de los modelos era superior, si lo hubiese, o si, por otro lado, las respuestas eran comparables.

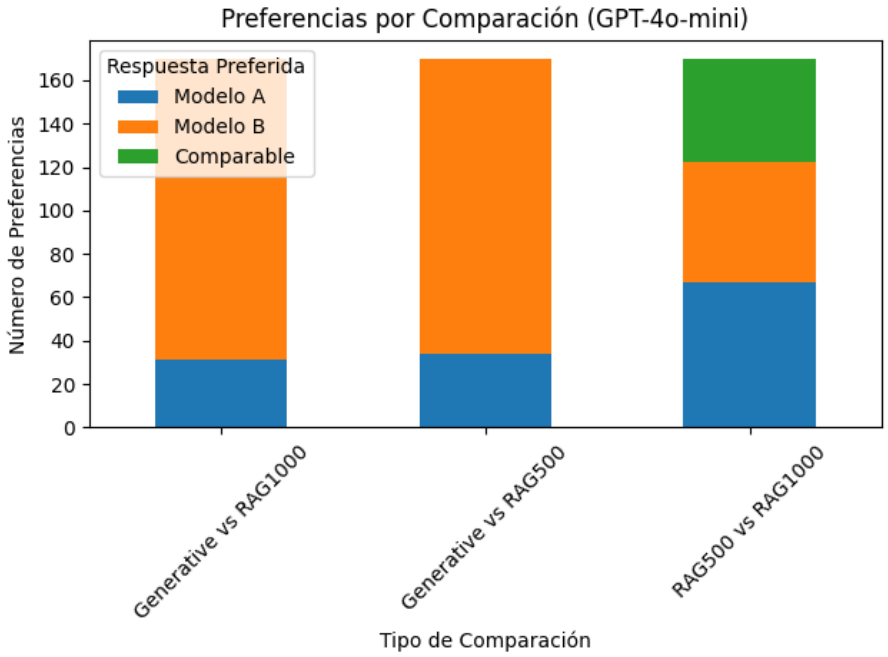


Figura 5.5: Preferencias por comparación entre los modelos generativos y RAG empleando GPT-4o-mini.

De este modo, se observa que entre los modelos, existe una aparente predisposición a seleccionar el modelo RAG sobre el LLM generativo. En la comparación de modelos RAG, parece haber una similitud en la respuesta, siendo comparable en una gran parte de las elecciones.

Para determinar si existe una inclinación significativa hacia uno de los modelos, se empleó una prueba Binomial, eliminando las opciones de *Comparable* del análisis.

Comparación	Preferencias A	Preferencias B	Total (A + B)	p-valor
Generative vs RAG500	34	136	170	1.19e-15*
Generative vs RAG1000	31	139	170	1.57e-17*
RAG500 vs RAG1000	67	55	122	3.19e-01

Cuadro 5.7: Comparación de preferencias entre modelos empleando GPT-4o-mini

* p-valor significativo con $\alpha = 0{,}05$

Como se puede observar en la Tabla 5.7, existe una inclinación a escoger un modelo RAG sobre un modelo puramente generativo. Además, no se aprecian diferencias significativas entre los modelos RAG (a diferencia del modelo GPT-3.5 Turbo).

En comparación con GPT-3.5 Turbo, se vio un incremento en la selección del modelo RAG sobre Generative, pero las diferencias entre ambos modelos RAG dejaron de ser significativas.

5.3.2. Análisis de las métricas de los modelos

En la Tabla 5.8, se presentan los resultados de la prueba de Wilcoxon, permitiendo analizar las diferencias significativas entre las métricas evaluadas para los diferentes modelos.

Comparación	Métrica	p-valor
Generative vs RAG500	Relevancia	1.91e-05*
Generative vs RAG500	Precisión	6.61e-13*
Generative vs RAG500	Completitud (total)	6.04e-07*
Generative vs RAG500	Concisión	1.16e-11*
Generative vs RAG1000	Relevancia	3.40e-07*
Generative vs RAG1000	Precisión	3.98e-15*
Generative vs RAG1000	Completitud (total)	1.22e-08*
Generative vs RAG1000	Concisión	1.07e-08*
RAG500 vs RAG1000	Fidelidad	6.24e-04*
RAG500 vs RAG1000	Relevancia	2.62e-03*
RAG500 vs RAG1000	Precisión	9.18e-03*
RAG500 vs RAG1000	Completitud (parcial)	2.71e-04*
RAG500 vs RAG1000	Completitud (total)	1.97e-02*
RAG500 vs RAG1000	Concisión	5.63e-03*

Cuadro 5.8: Resultados de la prueba de Wilcoxon para comparación de métricas entre modelos

* p-valor significativo con $\alpha = 0{,}05$

Se observaron diferencias significativas entre los modelos en varias métricas. En comparación con los modelos anteriores, en los nuevos modelos con GPT-4o-mini, todas las métricas mostraron diferencias significativas en todas las comparaciones.

Anteriormente, las diferencias entre **RAG1000 y el generativo** respecto a Concisión no fueron significativas (pese a estar muy cerca del valor de significancia, con un p-valor de 0.06503). Por el contrario, al emplear el modelo GPT-4o-mini, las diferencias fueron mucho más significativas.

Asimismo, las diferencias entre **ambos modelos RAG**, mostraron una diferencia mucho más elevada, habiendo una diferencia significativa entre ambos modelos en todas las métricas, mientras que en los modelos anteriores, solo se presenciaron diferencias en Fidelidad y Concisión.

Cabe destacar, que en la métrica de Concisión, la diferencia fue significativa entre ambos modelos, pero el modelo que mostró una media más alta fue, a diferencia del resto de métricas, RAG500, mostrando cierta superioridad respecto al modelo RAG1000.

5.4. Análisis del Aumento de Tamaño Muestral (n=340)

El objetivo de este análisis es evaluar cómo el incremento en el tamaño muestral, de n=170 a **n=340**, impacta en las métricas evaluadas en los modelos generados. Para ello, se compararon los resultados obtenidos con un tamaño muestral mayor, con el fin de determinar si las diferencias observadas persisten.

5.4.1. Preferencias por comparación

Inicialmente, se solicitó al evaluador que escogiese de entre dos modelos, que respuesta era mejor en términos generales, si la hubiese, o informase de que eran *Comparables*. El resultado de la comparación se puede observar en la Figura 5.6.

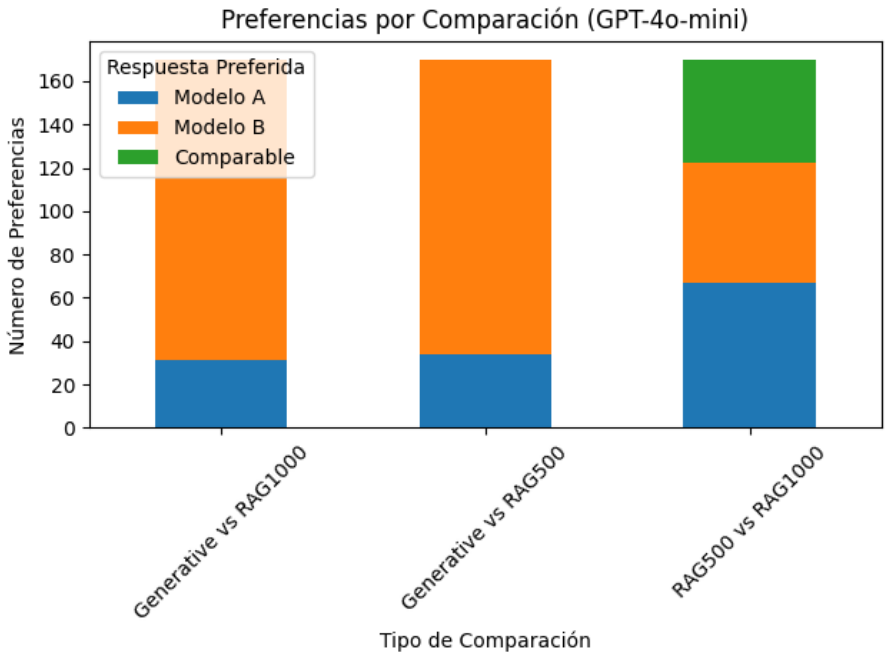


Figura 5.6: Preferencias por comparación entre los modelos generativos y RAG incrementando el tamaño muestral a n=340

.

Como se puede observar en la Tabla 5.9, que la inclinación a escoger un modelo RAG respecto a un modelo Generativo no se ha visto alterada. En la selección entre respuestas entre modelos RAG, la diferencia se ha visto disminuida, obteniendo una mayor cantidad de respuestas comparables, sin haber diferencias singificativas entre las respuestas.

Comparación	Preferencias A	Preferencias B	Total (A + B)	p-valor
Generative vs RAG500	87	252	339	9.4357e-20*
Generative vs RAG1000	72	267	339	1.7440e-27*
RAG500 vs RAG1000	141	144	285	9.0572e-01

Cuadro 5.9: Comparación de preferencias entre modelos

* p-valor significativo con $\alpha = 0{,}05$

5.4.2. Análisis de las métricas de los modelos con incremento muestral

Como se puede observar en la Tabla 5.10, se presentan los resultados del análisis de diferencias entre los modelos mediante la prueba de Wilcoxon. Este análisis permite evaluar de manera estadística las variaciones significativas en las métricas de los modelos comparados, permitiendo comparar el resultado con el modelo anterior.

En general, la mayoría de las métricas muestran un aumento en la estadística Wilcoxon con el incremento del tamaño muestral, lo cual es esperado. En concreto, se mostraron tres métricas que dejaron de tener diferencias significativas.

En particular, en la comparación con el **modelo Generative**, la métrica de Relevancia no mostró una diferencia significativa en ambos modelos RAG, mostrando un desempeño similar (a diferencia del LLM anterior).

Aún así, el resto de las métricas significativas en el modelo anterior se mantuvieron significativas, incluso tras incrementar la potencia estadística.

Comparación	**Métrica**	**p-valor**
Generative vs RAG500	Relevancia	5.94e-01
Generative vs RAG500	Precisión	1.18e-07*
Generative vs RAG500	Completitud (total)	9.04e-08*
Generative vs RAG500	Concisión	7.20e-03*
Generative vs RAG1000	Relevancia	1.32e-01
Generative vs RAG1000	Precisión	3.53e-09*
Generative vs RAG1000	Completitud (total)	5.84e-14*
Generative vs RAG1000	Concisión	1.19e-04*
RAG500 vs RAG1000	Fidelidad	1.30e-07*
RAG500 vs RAG1000	Relevancia	1.35e-03*
RAG500 vs RAG1000	Precisión	1.15e-03*
RAG500 vs RAG1000	Completitud (parcial)	1.24e-09*
RAG500 vs RAG1000	Completitud (total)	1.30e-05*
RAG500 vs RAG1000	Concisión	2.54e-03*

Cuadro 5.10: Resultados de la prueba de Wilcoxon para las métricas de los modelos con aumento de tamaño muestral.

* p-valor significativo con $\alpha = 0{,}05$

Capítulo 6

Discusión, limitaciones y líneas futuras de trabajo

6.1. Discusión de los resultados

En los resultados, se ha podido observar que la arquitectura RAG mejora sistemáticamente la calidad de las respuestas clínicas frente a modelos generativos puros, especialmente en precisión factual y completitud. De hecho, el LLM evaluador confirmó consistentemente la superioridad de RAG en más del 75 % de los casos en todos los experimentos.

También se confirmó que el **tamaño del chunk** es una variable que considerar, ya que el modelo basado en chunks de 1000 caracteres ofrecieron mejores resultados que los de 500, especialmente en fidelidad, precisión y completitud. De hecho, los modelos más potentes mejoraron las métricas en todos los casos, pero la ventaja de RAG se mantuvo incluso en los mejores modelos generativos, por lo que el uso de RAG en guías clínicas es especialmente útil para asegurar la precisión y la fidelidad clínica, minimizando las posibles alucinaciones y mejorando la utilidad médica de las respuestas generadas. Estos resultados se alinearon con la literatura revisada anteriormente, en que se observo el posible efecto de la fragmentación en la recuperación del contexto (48; 75).

Estos hallazgos se pueden explicar por varios factores clave. En primer lugar, la mejora sistemática mostrada por los modelos RAG respecto a los modelos generativos puros se puede atribuir al hecho de que **RAG reduce las alucinaciones** al basarse exclusivamente al contexto recuperado, conteniendo información explícita y validada previamente (4; 76). Al restringir la generación de respuestas a información recuperada de las guías clínicas, se evita que el modelo se base en conocimiento no verificado, erróneo o incompleto, lo que explica el incremento observado en métricas clave como la precisión factual.

Cabe destacar que las **guías clínicas** ofrecen un marco de referencia clínicamente validado, permitiendo que las respuestas generadas con RAG estén más alineadas con **estándares médicos reales**, incrementando la relevancia de las respuestas, reduciendo la ambigüedad o divagación. No obstante, la calidad de la información no solo depende de la fuente, sino también de cómo fue procesada para poder emplearla en el modelo. Esta mejora quedó reflejada en los modelos con chunks de 1000 caracteres, poniendo en manifiesto la **relevancia crítica del preprocesamiento** en arquitecturas RAG al proporcionar mayor contexto por fragmento. El chunking y su estructuración influyeron directamente en la capacidad del modelo para

recuperar y comprender contexto relevante y preciso, permitiendo respetar la lógica clínica de las guías, evitando cortes de secciones y recomendaciones clave, permitiendo preservar la coherencia semántica de cada chunk. Esto es especialmente relevante en un ámbito que requiere de mucha precisión, como el ámbito médico y clínico, ya que una segmentación errónea podría comprometer la calidad de la respuesta generada (77; 2).

Un aspecto relevante que destacar en los resultados es que, mientras los modelos con chunks de 1000 caracteres proporcionaban mayor coherencia, los **modelos con chunks de 500 caracteres** tendieron a generar respuestas **más concisas**. Esta mayor concisión podría deberse a la menor información proporcionada por cada chunk, reduciendo la posibilidad de que el modelo introduzca contenido redundante o divague. Al trabajar con fragmentos más cortos, el modelo se vió forzado a centrarse en la información esencial disponible, generando respuestas más compactas, sencillas y directas, pero sin perder la potencia respecto al modelo generativo base. Esta ventaja, pero, podría ir acompañada de una **pérdida parcial de completitud**, ya que, al disponer de menos contexto, se limitó la capacidad del modelo para abordar la totalidad de aspectos clínicos clave de la pregunta realizada. Por lo tanto, se debe buscar un equilibrio entre concisión y cobertura, donde el tamaño del chunk sería una variable clave en función de la prioridad en la generación de respuestas.

Durante el tercer análisis, se evidenció que el uso de un **LLM más avanzado** (en este caso, GPT-4.1 mini) permitió **mejorar el rendimiento** del modelo generativo puro en métricas como la **relevancia**, en comparación con modelos más sencillos (como el modelo original GPT-3.5 Turbo). Este resultado, sugiere que los LLMs más potentes son capaces de **generalizar mejor** a partir de los datos obtenidos durante el entrenamiento previo a su uso, de manera que generan respuestas más centradas incluso en ausencia de contexto externo especializado. Sin embargo, los modelos siguen mostrando menor fiabilidad incluso con la mejora, sobre todo en métricas claves como la precisión factual. Este comportamiento se explica debido a que, en ausencia de contexto específico que limite la generación, los LLM tienden a completar las respuestas de forma probabilística, pese a que no esté basada en hechos clínicamente validados. Esto indica que, aunque el aumento de la capacidad del modelo puede reducir en parte la dependencia de la recuperación de contexto, **no elimina el riesgo a posibles alucinaciones** en ausencia de información verificada, ni garantiza que las respuestas generadas se alineen con los estándares médicos necesarios (78). Por lo tanto, la integración de fuentes externas verificadas y fiables mediante la recuperación de una arquitectura **RAG sigue siendo fundamental** para asegurar la veracidad de las respuestas y su robustez en un entorno clínico y médico.

Además de las variables directamente evaluadas, existen otras variables que permitieron interpretar los resultados obtenidos, como la estructura semántica de las propias guías, las cuales están estructuradas y organizadas en bloques diferenciados, lo que favorece una recuperación eficiente de la información mediante los sistemas RAG, ya que la segmentación mantiene la coherencia, maximizando la capacidad para generar respuestas alineadas con la lógica de la consulta.

Asimismo, la métrica de **completitud** fue la más beneficiada por el uso de RAG, lo que se puede interpretar como un resultado directo al acceso explícito a información extensa y detallada sobre un ámbito clínico concreto.

Estos resultados, además de evidencia empírica sobre la eficacia de las arquitecturas RAG, también presentan implicaciones significativas para su aplicación práctica en entornos clínicos reales. Una de las principales ventajas observadas es la capacidad de **minimización de alu-**

cinaciones y mejora de la precisión factual de las respuestas, crucial en el ámbito médico en que los errores informativos o ambigüedades pueden tener consecuencias importantes para el paciente.

Los modelos generativos puros presentan un riesgo potencial al crear las respuestas basándose en inferencias probabilísticas, pero no siempre están basadas en evidencias médicas contrastadas. Por el contrario, al limitar la generación únicamente al contenido recuperado de fuentes médicas fiables, la arquitectura RAG garantiza un control de la información utilizada en la toma de decisiones, mejorando la fidelidad de las respuestas generadas. Este principio se puede aplicar a sistemas de apoyo a la decisión clínica, donde **RAG** podría implementarse como **mecanismo de respaldo**, permitiendo a los profesionales sanitarios consultar recomendaciones basadas en evidencia clínica actualizada y verificada.

De hecho, se podrían aplicar modelos RAG en el desarrollo de **asistentes virtuales especializados**, en especial en atención primaria o servicios de urgencias, donde el acceso a información fiable y estructurada de manera rápida y eficiente puede agilizar la toma de decisiones. Al integrar información como historiales clínicos digitales, estos asistentes podrían personalizar las recomendaciones, ofreciendo diagnósticos preliminares, guiar los tratamientos o detectar contradicciones en los tratamientos (79).

Por último, implementar sistemas RAG en entornos de salud pública permitiría estandarizar criterios de actuación o mejorar la coherencia entre especialistas médicos y centros, sobre todo en patologías crónicas o complejas. Por lo tanto, las arquitecturas RAG no suponen únicamente una mejora en la generación de respuestas, sino que tiene el potencial de convertirse en una herramienta segura, escalable y segura en la asistencia clínica moderna, siempre que su integración sea acompañada por validaciones humanas expertas y supervisión continua.

Estos resultados permitieron confirmar el cumplimiento de los objetivos planteados en este trabajo:

1. **Integración de guías clínicas de reumatología en un modelo RAG:** Se incorporaron con éxito las guías clínicas reumáticas en la arquitectura RAG, permitiendo que el modelo utilice información validada y actualizada para generar respuestas clínicas. Esto fue determinante para mejorar la fidelidad y precisión de las respuestas obtenidas.

2. **Evaluación comparativa del rendimiento del sistema RAG:** Se realizó un análisis exhaustivo que evidenció que los modelos RAG superan sistemáticamente a los modelos generativos puros en métricas como precisión factual y completitud. Además, se demostró que el tamaño del chunk es una variable crítica, con los chunks de 1000 caracteres optimizando la recuperación del contexto clínico.

3. **Desarrollo de una interfaz visual:** Aunque no se detalla en profundidad en esta sección, se desarrolló una interfaz gráfica intuitiva que integra el modelo RAG optimizado, facilitando el acceso información clínica verificada de manera rápida y precisa.

Asimismo, se abordaron y resolvieron los objetivos secundarios relacionados con el procesamiento de documentos clínicos, la revisión del estado del arte, la comparación de arquitecturas RAG, asi como el diseño, implementación y validación de la intrerfaz web. En conjunto, estos resultados evidencian que la propuesta técnica y metodológica planteada responde a los retos de integración y mejora de sistemas de lenguaje para aplicaciones clínicas como herramienta de soporte en la toma de decisiones.

6.2. Limitaciones y riesgos

A pesar de los hallazgos obtenidos, es importante reconocer ciertas limitaciones que pueden influir en los resultados y su generalización y, por lo tanto, deben ser consideradas en su interpretación y en posibles líneas futuras de trabajo.

6.2.1. Restricciones por dominio clínico específico

En primer lugar, el trabajo se enfocó en un **dominio clínico específico** (las enfermedades reumatológicas contenidas en las guías clínicas del ACR). Pese a obtener unos resultados óptimos y significativos, al haber empleado información de un solo ámbito médico, se limita la generalización de los resultados a otras especialidades médicas, que pueden tener estructuras o complejidades diferentes. Por ejemplo, estudios anteriores, como el de Adejumo et al. (2024) (69) en el ámbito cardiovascular o Miao et al. (2024) (73) en nefrología demostraron que el desempeño de RAG depende en gran medida de la naturaleza del dominio médico, el tipo de guías empleadas y la precisión requerida para interpretar factores clínicos definidos. Incluso se han implementado arquitecturas RAG específicas en contextos preoperatorios (Ke et al., 2024) (71) o diagnóstico visual (Bani-Harouni et al., 2024) (70), reforzando la idea de que **cada dominio clínico plantea limitaciones y desafíos únicos**. Por lo tanto, aunque los resultados de este trabajo fueron prometedores en el ámbito reumatológico, seria necesario verificar la adaptabilidad y eficacia en otras especialidades médicas, ya que la extrapolación y generalización sin ajustes específicos podría resultar inapropiada, sobre todo en áreas con documentación menos estructurada o específica.

6.2.2. Escalabilidad y cobertura

Además, pese a la alta calidad de la información proporcionada por las guías clínicas, la cantidad de documentos procesados fue reducida en comparación con el volumen real de información médica disponible (23 documentos para 17 patologías). Esto podría limitar la diversidad y complejidad del conocimiento recuperado, ya que al utilizar un **máximo de 2 documentos por patología** puede afectar negativamente a la generalización del modelo RAG, especialmente si se enfrenta a consultas mal formuladas o que abordan temas poco representados en las guías. Asimismo, al limitar la documentación a un conjunto de guías clínicas específica, es posible que ciertos aspectos, como las comorbilidades o tratamientos emergentes, no estén cubiertos, reduciendo la capacidad de recuperar información relevante y actualizada en consultas complejas.

De hecho, la propia evaluación del sistema se ve condicionada, ya que tanto las preguntas como las respuestas evaluadas están limitadas al conocimiento contenido del conjunto reducido de documentos y, en consecuencia, las métricas podrían no reflejar el rendimiento real en escenarios más amplios o con información más diversa.

Para mitigar estos problemas, los futuros proyectos podrían incorporar un **mayor número de documentos**, incluyendo guías clínicas de diferentes fuentes institucionales, revisiones u otras bases de conocimiento médico estructurado, permitiendo evaluar la fidelidad, la robustez y la escalabilidad del sistema generado.

6.2.3. Ausencia de validación humana experta

En este trabajo, la evaluación de las respuestas generadas por los modelos se llevó a cabo utilizando un LLM externo (Gemini 1.5 pro) con el objetivo de generar una valoración automática e independiente del rendimiento del modelo. Aunque este enfoque tiene ventajas, como la **rapidez y consistencia** en los criterios de evaluación, la dependencia de un único evaluador puede incorporar sesgos inherentes del propio modelo, derivados del entrenamiento y que puede interpretar atributos de manera diferente a como lo haría un médico experto humano, especialmente en dominios clínicos. Este riesgo se incrementa al no haber incorporado una evaluación cualitativa por parte de médicos expertos. De hecho, la **ausencia de validación humana** limita la capacidad de contrastar las métricas con juicios clínicamente informados. En un ámbito como el clínico y médico, el razonamiento clínico y la interpretación basada en experiencia real juegan un papel crucial, por lo que el juicio humano es insustituible para garantizar la utilidad real de la arquitectura propuesta.

Además, sin la participación de expertos durante la evaluación, es más difícil detectar posibles errores clínicos, ambigüedades o alucinaciones sutiles que pasan desapercibidas para un LLM, pero que supondrían un riesgo en su aplicación real. Por tanto, la fiabilidad clínica y la validez externa de los resultados se ven comprometidos, de modo que trabajos futuros deberían combinar evaluaciones automáticas con **revisiones de expertos médicos** que capturen la precisión técnica y la aplicabilidad clínica de las respuestas generadas, o incluso evaluar las diferencias identificadas en los dos métodos de evaluación, para determinar que características se ven potencialmente afectadas en la ausencia de revisiones generadas por médicos especializados.

6.2.4. Contexto simulado y métricas parciales

Otra de las principales limitaciones radica en que la evaluación se realizó en un **entorno clínico simulado**, utilizando métricas generadas por un LLM a partir de las propias guías clínicas reumatológicas, sin emplear casos clínicos reales ni ayuda de profesionales médicos. Aunque esta aproximación permite controlar variables y medir el rendimiento del sistema, también implica que las condiciones reales de la práctica médica, como comorbilidades, no estén representadas. Por lo tanto, se compromete la validez externa del sistema, ya que el comportamiento del sistema en situaciones reales podría verse alterada significativamente.

Además, la evaluación se centró en 5 métricas determinadas, que, si bien son apropiadas para valorar la calidad de las respuestas generadas, no abarcan otros atributos o áreas críticas para su aplicación clínica, como la interpretabilidad, la utilidad clínica efectiva o el riesgo potencial a errores diagnósticos. Se debe recordar que, en el contexto médico, no basta con que la respuesta sea correcta, ya que también debe ser comprensible, segura y práctica para la toma de decisiones.

6.2.5. Valoración económica

En este trabajo, se utilizaron diversos modelos de lenguaje y componentes como rerankers, que tienen costes asociados en función del número de tokens procesados, o el número de llamadas realizadas a la API.

Durante el embedding, la API de **OpenAI** generó gastos proporcionales al número de fragmentos de texto procesados para la indexación. En este trabajo, el coste del modelo de

embedding *text-embedding-large-3* fue de 0.00013$ por 1000 tokens, por lo que durante la generación de las dos bases de datos vectorial, el coste real fue de 0.19$, empleando un total de 1.924.553 tokens.

Asimismo, la trazabilidad permitida con **Langsmith** supusieron costos adicionales al habilitar la observabilidad, generando un costo por cada llamada registrada en la API. Actualmente, existe un plan gratuito con 5000 llamadas a la API por mes, independientemente de los tokens necesarios, con un coste adicional de 0.50$ por cada 1000 rastros adicionales (80).

Del mismo modo, **Cohere** proporciona una API de prueba, que presenta limitaciones de 10 consultas por minuto, con un límite de 1000 consultas al mes (81).

En lo referente a **Gemini 1.5 pro**, el coste es de 2.50$ por millón de tokens en entrada y 10.00$ por millón de tokens de salida. Durante el trabajo, se utilizó el plan gratuito, que permite una disponibilidad de 278.05$, incurriendo en un costo total de 16.75$ (82).

En cuanto a los modelos de lenguaje de **OpenAI**, se emplearon diferentes versiones en los análisis realizados, cada uno con precio diferente según el modelo y el tipo de token (entrada o salida):

- **GPT-3.5 Turbo**: 0.50$ por millón de tokens de entrada y 1.50$ por millón de tokens de salida.
- **GPT-4-o mini**: 0.15$ por millón de tokens de entrada y 0.60$ por millón de tokens de salida.
- **GPT-4.1 mini**: 0.40$ por millón de tokens de entrada y 1.60$ por millón de tokens de salida.

Durante el trabajo, OpenAI permitió un periodo de prueba gratuita, permitiendo hasta 1 millón de tokens por día en GPT-4.1 mini y GPT-4-o-mini, mitigando el coste final del trabajo (83). En total, se emplearon casi 3 millones de tokens de GPT-4.1 mini, y aproximadamente 1.5 millones de tokens para los modelos GPT-3.5 Turbo y GPT-4-o-mini.

6.3. Líneas futuras de investigación

Los resultados, aunque prometedores, dejan áreas que requieren de un análisis más profundo para lograr una implementación efectiva y generalizada de los modelos RAG. Dadas las limitaciones, se plantearon 4 posibles líneas futuras de investigación.

1. **Ampliación del número de documentos con guías clínicas de diferentes especialidades médicas:** El hecho de incrementar el conjunto de guías clínicas con otras especialidades médicas permitiría evaluar si el sistema RAG puede generalizar eficazmente en otros dominios, evaluando cómo las diferencias en la estructura de las guías y el conocimiento clínico complejo influirían en el rendimiento del sistema. El objetivo de esta línea de trabajo es asegurar la generalización de los resultados a otras áreas para integrar el sistema RAG a gran escala en la práctica clínica. Además, permitiría identificar posibles limitaciones en modelos existentes al aplicar un conjunto de documentos más amplio y diverso.

2. **Evaluación clínica real:** Debido a que la simulación de preguntas no refleja la complejidad de las decisiones médicas reales, validar el modelo en un entorno clínico real sería fundamental para verificar su eficacia, y podría revelar limitaciones o sesgos sutiles no detectados. Por lo tanto, realizar estudios piloto en centros de salud reales, donde se utilicen los modelos RAG para apoyar la toma de decisiones en situaciones clínicas reales permitiría, a la vez, contar con el punto de vista de profesionales médicos en tiempo real, y al mismo tiempo, evaluar la validez externa de los modelos.

3. **Creación de un nuevo conjunto de documentos:** Las guías clínicas, pese a que son rigurosas, no abarcan todos los casos, o algunas no han sido actualizadas desde hace tiempo. Por ende, incorporar fuentes adicionales aumentaría la cobertura del sistema, además de mejorar la capacidad de respuesta en consultas complejas y reforzando la utilidad práctica del modelo en situaciones clínicas reales. Asimismo, implementar una arquitectura RAG que combine guías clínicas con fuentes alternativas como artículos científicos o registro médicos estructurados, permitiría una recuperación más rica y contextualizada.

4. **Integración de razonamiento clínico:** En la práctica, la recuperación de información no es suficiente, por lo que integrar el razonamiento claro y seguro aumentaría la utilidad real del modelo y su aceptabilidad por el personal médico, incluso reducir errores en contextos críticos. Esta capacidad de razonamiento, incluiría funciones como el análisis de riesgos, sugerencia de tratamientos personalizados basados en el historial clínico o la interpretación de condiciones clínicas complejas. Se podría generar un sistema multiagente, en que agentes especializados colaboren para formar una cadena de pensamiento clínico, integrando diferentes etapas, como la evaluación diagnóstica, el análisis de riesgos, estudio del historial clínico del paciente. Así, el sistema justificaría las recomendaciones, permitiendo al especialista médico determinar la calidad de las respuestas en tiempo real.

5. **Verificación de la arquitectura RAG en otros ámbitos clínicos:** Debido a que cada especialidad tiene características y desafíos únicos, explorar la aplicación de la arquitectura RAG diseñada en otros dominios clínicos permitiría evaluar la adaptabilidad, así como identificar los ajustes necesarios en la arquitectura y mejorar la precisión y relevancia de respuestas en diferentes áreas médicas. Además, proporcionaría una base sólida para la generalización y expansión de RAG a nivel multiseccional en el ámbito clínico y médico, permitiendo su integración como herramienta de soporte a la decisión clínica en múltiples especialidades.

Capítulo 7

Conclusión

Este estudio ha demostrado que integrar arquitecturas RAG mejora de manera significativa la calidad de las respuestas generadas por los modelos de lenguaje en el contexto clínico, especialmente cuando se comparan con modelos generativos puros. Las mejoras son evidentes y consistentes en las métricas de fidelidad, precisión factual y completitud, con una superioridad observada en más del 75 % de los casos evaluados en todos los experimentos realizados.

Uno de los hallazgos clave fue la importancia del **tamaño de los chunks**. Se observaron diferencias notables en los resultados dependiendo de su longitud, ya que los **chunks de 1000** caracteres favorecieron la **fidelidad** y la **completitud**, mientras que los de **500 caracteres** promovieron una mayor **concisión**. Esto resalta la necesidad de equilibrar la concisión y la cobertura, ajustándolos según el tipo de pregunta o el escenario clínico abordado.

Al comparar los diferentes LLM, se observó que GPT-4.1 presentó mejoras generales sobre GPT-3.5 Turbo. Sin embargo, aún persistieron limitaciones, como las alucinaciones en ausencia de contexto externo, confirmando que **incluso los modelos más avanzados siguen siendo vulnerables** a errores sin un mecanismo de recuperación de información confiable. A pesar de los avances en la arquitectura del modelo, la integración de RAG sigue siendo un elemento clave que refuerza su utilidad, incluso cuando se utilizan modelos generativos de última generación.

Estas mejoras contribuyen a **reducir los errores** y **mejorar la precisión y relevancia** de las respuestas, lo que abre la puerta a su aplicación en sistemas de apoyo a la toma de decisiones clínicas y en el desarrollo de asistentes virtuales para profesionales médicos. Además, al proporcionar un contexto estandarizado y estructurado, como el que ofrecen las guías clínicas, RAG facilita una toma de decisiones más coherente, segura y basada en evidencia, con el potencial de estandarizar criterios entre centros y profesionales de la salud.

A pesar de estos avances, el estudio presenta algunas limitaciones. Al centrarse exclusivamente en el ámbito de la reumatología, no se puede asegurar que los resultados sean generalizables a otras especialidades sin validación adicional, dado que los documentos clínicos pueden variar significativamente en su estructura y contenido. Además, el corpus documental utilizado, compuesto por 23 guías clínicas, podría no ser representativo de situaciones más complejas o excepcionales.

Asimismo, la evaluación automatizada utilizando un LLM, sin la intervención de expertos humanos, limita la validez clínica de los resultados. Es posible que algunos errores sutiles hayan pasado desapercibidos, especialmente en aquellos casos que requieren un razonamiento clínico más profundo o interpretación contextual. Del mismo modo, el entorno simulado utilizado

no permite evaluar la aplicabilidad práctica del sistema en un entorno clínico real, ni medir aspectos como la interpretabilidad, el tiempo de respuesta o la aceptación por parte del personal sanitario.

En cuanto a las futuras líneas de investigación, se propone ampliar el corpus de guías clínicas a otras especialidades médicas para evaluar la capacidad de generalización de la arquitectura RAG. También se sugiere **validar el sistema en entornos clínicos reales** a través de estudios piloto con profesionales sanitarios. Además, sería interesante incorporar **nuevas fuentes de conocimiento**, como artículos científicos, protocolos hospitalarios o datos estructurados, y desarrollar capacidades de razonamiento clínico. Para ello, se plantea la implementación de un sistema multiagente que integre diversas fases del proceso clínico, como análisis de riesgos, personalización de respuestas y justificación explícita del razonamiento.

Bibliografía

[1] Garcia BT, Westerfield L, Yelemali P, Gogate N, Rivera-Munoz EA, Du H, et al. Improving Automated Deep Phenotyping Through Large Language Models Using Retrieval Augmented Generation. medRxiv. 2024 Dec. Preprint. Available from: https://doi.org/10.1101/2024.12.01.24318253.

[2] Kresevic S, Giuffre M, Ajcevic M, Accardo A, Croce LS, Shung DL. Optimization of hepatological clinical guidelines interpretation by large language models: a retrieval augmented generation-based framework. NPJ Digital Medicine. 2024 April 23;7(1):102. Available from: https://www.nature.com/articles/s41746-024-01091-y.

[3] Murugan M, Yuan B, Venner E, Ballantyne CM, Robinson KM, Coons JC, et al. Empowering personalized pharmacogenomics with generative AI solutions. Journal of the American Medical Informatics Association. 2024 May 20;31(6):1356-66. Available from: https://doi.org/10.1093/jamia/ocae039.

[4] Xiong G, Jin Q, Lu Z, Zhang A. Benchmarking Retrieval-Augmented Generation for Medicine; 2024. Available from: https://arxiv.org/abs/2402.13178.

[5] Delanoe P, Tchuente D, Colin G. Method and evaluations of the effective gain of artificial intelligence models for reducing CO2 emissions. Journal of Environmental Management. 2023;331:117261. Available from: https://www.sciencedirect.com/science/article/pii/S030147972300049X.

[6] Yokoi T. GenAI: Ventaja competitiva versus coste medioambiental; 2024. Accessed: 2025-02-21. Available from: https://www.imd.org/ibyimd/artificial-intelligence/genai-competitive-advantage-versus-environmental-cost/.

[7] Li P, Yang J, Islam MA, Ren S. Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models; 2025. Available from: https://arxiv.org/abs/2304.03271.

[8] Programa de las Naciones Unidas para el Medio Ambiente (PNUMA). La IA plantea problemas ambientales. Esto es lo que el mundo puede hacer al respecto. Programa de las Naciones Unidas para el Medio Ambiente. 2024 September 21. Reportaje Medio ambiente bajo revisión. Available from: https://www.unep.org/es/noticias-y-reportajes/reportajes/la-ia-plantea-problemas-ambientales-esto-es-lo-que-el-mundo-puede.

[9] Abi-Rafeh J, Henry N, Xu HH, Bassiri-Tehrani B, Arezki A, Kazan R, et al. Utility and Comparative Performance of Current Artificial Intelligence Large Language Models as Postoperative Medical Support Chatbots in Aesthetic Surgery. Aesthetic Surgery Journal. 2024 02;44(8):889-96. Available from: https://doi.org/10.1093/asj/sjae025.

[10] Morley J, Machado CCV, Burr C, Cowls J, Joshi I, Taddeo M, et al.. The ethics of AI in health care: A mapping review; 2020. Available from: https://www.sciencedirect.com/science/article/pii/S0277953620303919.

[11] Küster D, Schultz T. Künstliche Intelligenz und Ethik im Gesundheitswesen - Spagat oder Symbiose? Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz. 2023 Feb;66(2):176-83. Available from: https://link.springer.com/article/10.1007/s00103-022-03653-5.

[12] Zhang C, Zhang H, Khan A, Kim T, Omoleye O, Abiona O, et al.. Lightweight Mobile Automated Assistant-to-physician for Global Lower-resource Areas; 2021. Available from: https://arxiv.org/abs/2110.15127.

[13] Chu CH, Nyrup R, Leslie K, Shi J, Bianchi A, Lyn A, et al. Digital Ageism: Challenges and Opportunities in Artificial Intelligence for Older Adults. The Gerontologist. 2022 August;62(7):947-55. Available from: https://doi.org/10.1093/geront/gnab167.

[14] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation); 2016. Available from: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[15] Health Insurance Portability and Accountability Act of 1996 (HIPAA); 1996. Available from: https://www.hhs.gov/hipaa/for-individuals/index.html.

[16] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al.. Attention Is All You Need; 2023. Available from: https://arxiv.org/abs/1706.03762.

[17] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al.. Language Models are Few-Shot Learners; 2020. Available from: https://arxiv.org/abs/2005.14165.

[18] Wang Z, Chu Z, Doan TV, Ni S, Yang M, Zhang W. History, Development, and Principles of Large Language Models-An Introductory Survey; 2024. Available from: https://arxiv.org/abs/2402.06853.

[19] Wang F, Zhang Z, Zhang X, Wu Z, Mo T, Lu Q, et al.. A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness; 2024. Available from: https://arxiv.org/abs/2411.03350.

[20] Bengio Y, Ducharme R, Vincent P. A Neural Probabilistic Language Model. vol. 3; 2000. p. 932-8.

[21] Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S. Recurrent neural network based language model. Proceedings of Interspeech. 2010 01;2.

[22] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding; 2019. Available from: https://arxiv.org/abs/1810.04805.

[23] Huang L, Yu W, Ma W, Zhong W, Feng Z, Wang H, et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Transactions on Information Systems. 2025 Jan;43(2):1–55. Available from: http://dx.doi.org/10.1145/3703155.

[24] Zhou Y, Muresanu AI, Han Z, Paster K, Pitis S, Chan H, et al.. Large Language Models Are Human-Level Prompt Engineers; 2023. Available from: https://arxiv.org/abs/2211.01910.

[25] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, et al.. Scaling Instruction-Finetuned Language Models; 2022. Available from: https://arxiv.org/abs/2210.11416.

[26] Iyer S, Lin XV, Pasunuru R, Mihaylov T, Simig D, Yu P, et al.. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization; 2023. Available from: https://arxiv.org/abs/2212.12017.

[27] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright CL, Mishkin P, et al.. Training language models to follow instructions with human feedback; 2022. Available from: https://arxiv.org/abs/2203.02155.

[28] Saxena Y, Chopra S, Tripathi AM. Evaluating Consistency and Reasoning Capabilities of Large Language Models; 2024. Available from: https://arxiv.org/abs/2404.16478.

[29] Shin A, Kaneko K. Large Language Models Lack Understanding of Character Composition of Words; 2024. Available from: https://arxiv.org/abs/2405.11357.

[30] Barnett S, Kurniawan S, Thudumu S, Brannelly Z, Abdelrazek M. Seven Failure Points When Engineering a Retrieval Augmented Generation System; 2024. Available from: https://arxiv.org/abs/2401.05856.

[31] Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, et al. Survey of Hallucination in Natural Language Generation. ACM Computing Surveys. 2023 Mar;55(12):1–38. Available from: http://dx.doi.org/10.1145/3571730.

[32] Bruckhaus T. RAG Does Not Work for Enterprises; 2024. Available from: https://arxiv.org/abs/2406.04369.

[33] Dong Y, Mu R, Jin G, Qi Y, Hu J, Zhao X, et al.. Building Guardrails for Large Language Models; 2024. Available from: https://arxiv.org/abs/2402.01822.

[34] Hu J, Dong Y, Huang X. Trust-Oriented Adaptive Guardrails for Large Language Models; 2025. Available from: https://arxiv.org/abs/2408.08959.

[35] Farquhar S, Kossen J, Kuhn L, et al. Detecting hallucinations in large language models using semantic entropy. Nature. 2024;630:625-30.

[36] Abi-Rafeh J, Xu HH, Kazan R, Tevlin R, Furnas H. Large Language Models and Artificial Intelligence: A Primer for Plastic Surgeons on the Demonstrated and Potential Applications, Promises, and Limitations of ChatGPT. Aesthetic Surgery Journal. 2024 Feb 15;44(3):329-43.

[37] Wang Y, Leutner S, Ingrisch M, Klein C, Hinske LC, Danhauser K. Optimizing Data Extraction: Harnessing RAG and LLMs for German Medical Documents; 2024.

[38] Ge J, Sun S, Owens J, Galvez V, Gologorskaya O, Lai JC, et al. Development of a liver disease-specific large language model chat interface using retrieval-augmented generation. Hepatology. 2024 November;80(5):1158-68. Available from: https://doi.org/10.1097/HEP.0000000000000834.

[39] LangChain. LangChain Documentation; 2025. [consultado 20 de marzo de 2025]. [Internet]. Disponible en: https://python.langchain.com/docs/introduction/.

[40] LlamaIndex. LlamaIndex Documentation; 2025. [consultado 20 de marzo de 2025]. [Internet]. Disponible en: https://docs.llamaindex.ai/en/stable/.

[41] deepset ai. Haystack Documentation; 2025. [consultado 20 de marzo de 2025]. [Internet]. Disponible en: https://docs.haystack.deepset.ai/docs/intro.

[42] Perez A, Vizcaino X. Advanced ingestion process powered by LLM parsing for RAG system; 2024. Available from: https://arxiv.org/abs/2412.15262.

[43] Yepes AJ, You Y, Milczek J, Laverde S, Li R. Financial Report Chunking for Effective Retrieval Augmented Generation; 2024. Available from: https://arxiv.org/abs/2402.05131.

[44] Singh IS, Aggarwal R, Allahverdiyev I, Taha M, Akalin A, Zhu K, et al.. ChunkRAG: Novel LLM-Chunk Filtering Method for RAG Systems; 2024. Available from: https://arxiv.org/abs/2410.19572.

[45] In: Zong FyLWyNR Chengqing y Xia, editor. Hallazgos de la Asociación de Lingüística Computacional: ACL-IJCNLP 2021;. Available from: https://aclanthology.org/2021.hallazgos-acl.29/.

[46] Chen LC, Pardeshi MS, Liao YX, Pai KC. Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model. Computer Standards Interfaces. 2025;94:103995. Available from: https://www.sciencedirect.com/science/article/pii/S0920548925000248.

[47] Agarwal S, Sundaresan S, Mitra S, Mahapatra D, Gupta A, Sharma R, et al.. Cache-Craft: Managing Chunk-Caches for Efficient Retrieval-Augmented Generation; 2025. Available from: https://arxiv.org/abs/2502.15734.

[48] Zhong Z, Liu H, Cui X, Zhang X, Qin Z. Mix-of-Granularity: Optimize the Chunking Granularity for Retrieval-Augmented Generation; 2025. Available from: https://arxiv.org/abs/2406.00456.

[49] Nguyen PV, Tran MN, Nguyen L, Dinh D. Advancing Vietnamese Information Retrieval with Learning Objective and Benchmark; 2025. Available from: https://arxiv.org/abs/2503.07470.

[50] Rau D, Wang S, Déjean H, Clinchant S. Context Embeddings for Efficient Answer Generation in RAG; 2024. Available from: https://arxiv.org/abs/2407.09252.

[51] Hugging Face. MTEB Leaderboard - Massive Text Embedding Benchmark; 2024. [consultado 15 de abril de 2025]. [Internet]. Disponible en: https://huggingface.co/spaces/mteb/leaderboard.

[52] Han Y, Liu C, Wang P. A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge; 2023. Available from: https://arxiv.org/abs/2310.11703.

[53] Öztürk E, Mesut A. Performance Analysis of Chroma, Qdrant, and FAISS Databases. UNITECH – Selected Papers. 2024. Available from: https://unitechsp.tugab.bg/images/2024/4-CST/s4_p72_v3.pdf.

[54] MongoDB Inc . What are Vector Databases?; 2024. [consultado 23 de marzo de 2025]. [Internet]. Disponible en: https://www.mongodb.com/es/resources/basics/databases/vector-databases.

[55] MongoDB. MongoDB Atlas Vector Search; 2025. [consultado 23 de marzo de 2025]. [Internet]. Disponible en: https://www.mongodb.com/es/products/platform/atlas-vector-search.

[56] Facebook AI Research. FAISS: A Library for Efficient Similarity Search; 2025. [consultado 23 de marzo de 2025]. [Internet]. Disponible en: https://github.com/facebookresearch/faiss.

[57] ChromaDB. ChromaDB Documentation: Introduction; 2025. [consultado 23 de marzo de 2025]. [Internet]. Disponible en: https://docs.trychroma.com/docs/overview/introduction.

[58] Papadimitriou I, Gialampoukidis I, Vrochidis S, Ioannis, Kompatsiaris. RAG Playground: A Framework for Systematic Evaluation of Retrieval Strategies and Prompt Engineering in RAG Systems; 2024. Available from: https://arxiv.org/abs/2412.12322.

[59] Gupta S, Ranjan R, Singh SN. A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions; 2024. Available from: https://arxiv.org/abs/2410.12837.

[60] Fang K, Tang C, Wang J. Evaluating simulated teaching audio for teacher trainees using RAG and local LLMs. Scientific Reports. 2025 January 29;15(1):3633.

[61] Raminedi S, Shridevi S, Won D. Multi-modal transformer architecture for medical image analysis and automated report generation. Scientific Reports. 2024 August;14(1):19281.

[62] Tevissen Y, Guetari K, Petitpont F. Towards Retrieval Augmented Generation over Large Video Libraries; 2024. Available from: https://arxiv.org/abs/2406.14938.

[63] Su C, Wen J, Kang J, Wang Y, Su Y, Pan H, et al.. Hybrid RAG-empowered Multimodal LLM for Secure Data Management in Internet of Medical Things: A Diffusion-based Contract Approach; 2024. Available from: https://arxiv.org/abs/2407.00978.

[64] Xue J, Zheng M, Hu Y, Liu F, Chen X, Lou Q. BadRAG: Identifying Vulnerabilities in Retrieval Augmented Generation of Large Language Models; 2024. Available from: https://arxiv.org/abs/2406.00083.

[65] Sohn J, Park Y, Yoon C, Park S, Hwang H, Sung M, et al.. Rationale-Guided Retrieval Augmented Generation for Medical Question Answering; 2024. Available from: https://arxiv.org/abs/2411.00300.

[66] Ong CS, Obey NT, Zheng Y, Cohan A, Schneider EB. SurgeryLLM: a retrieval-augmented generation large language model framework for surgical decision support and workflow enhancement. npj Digital Medicine. 2024;7(1):364. Available from: https://doi.org/10.1038/s41746-024-01391-3.

[67] Uapadhyay R, Viviani M. Enhancing Health Information Retrieval with RAG by Prioritizing Topical Relevance and Factual Accuracy; 2025. Available from: https://arxiv.org/abs/2502.04666.

[68] Jeunen O, Potapov I, Ustimenko A. On (Normalised) Discounted Cumulative Gain as an Off-Policy Evaluation Metric for Top-n Recommendation; 2024. Available from: https://arxiv.org/abs/2307.15053.

[69] Adejumo P, Thangaraj P, Shankar SV, Dhingra LS, Aminorroaya A, Khera R. Retrieval-Augmented Generation for Extracting CHADS-VASc Risk Factors from Unstructured Clinical Notes in Patients with Atrial Fibrillation. medRxiv [Preprint]. 2024 Sep. Available from: https://doi.org/10.1101/2024.09.19.24313992.

[70] Bani-Harouni D, Navab N, Keicher M. MAGDA: Multi-agent guideline-driven diagnostic assistance; 2024. Available from: https://arxiv.org/abs/2409.06351.

[71] Ke Y, Jin L, Elangovan K, Abdullah HR, Liu N, Sia ATH, et al.. Development and Testing of Retrieval Augmented Generation in Large Language Models – A Case Study Report; 2024. Available from: https://arxiv.org/abs/2402.01733.

[72] Giuffre M, Kresevic S, Pugliese N, You K, Shung DL. Optimizing large language models in digestive disease: strategies and challenges to improve clinical outcomes. Liver International. 2024 September;44(9):2114-24. Epub 2024 May 31. Available from: https://doi.org/10.1111/liv.15974.

[73] Miao J, Thongprayoon C, Suppadungsuk S, Garcia Valencia OA, Cheungpasitporn W. Integrating Retrieval-Augmented Generation with Large Language Models in Nephrology: Advancing Practical Applications. Medicina (Kaunas). 2024 March 8;60(3):445.

[74] Gradio. Gradio: Build Machine Learning Web Apps — in Python; 2023. [consultado 24 de abril de 2025]. [Internet]. Disponible en: https://www.gradio.app/.

[75] Juvekar K, Purwar A. Introducing a new hyper-parameter for RAG: Context Window Utilization; 2024. Available from: https://arxiv.org/abs/2407.19794.

[76] Li J, Yuan Y, Zhang Z. Enhancing LLM Factual Accuracy with RAG to Counter Hallucinations: A Case Study on Domain-Specific Queries in Private Knowledge-Bases; 2024. Available from: https://arxiv.org/abs/2403.10446.

[77] Khan AA, Hasan MT, Kemell KK, Rasku J, Abrahamsson P. Developing Retrieval Augmented Generation (RAG) based LLM Systems from PDFs: An Experience Report; 2024. Available from: https://arxiv.org/abs/2410.15944.

[78] Li Z, Li C, Zhang M, Mei Q, Bendersky M. Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach; 2024. Available from: https://arxiv.org/abs/2407.16833.

[79] Zhao X, Liu S, Yang SY, Miao C. MedRAG: Enhancing Retrieval-augmented Generation with Knowledge Graph-Elicited Reasoning for Healthcare Copilot; 2025. Available from: https://arxiv.org/abs/2502.04413.

[80] LangChain. LangSmith Pricing; 2024. [consultado 5 de mayo de 2025]. [Internet]. Disponible en: https://www.langchain.com/pricing-langsmith.

[81] Cohere. Cohere API Rate Limits; 2024. [consultado 5 de mayo de 2025]. [Internet]. Disponible en: https://docs.cohere.com/v2/docs/rate-limits.

[82] AI G. Gemini API Pricing; 2024. [consultado 5 de mayo de 2025]. [Internet]. Disponible en: https://ai.google.dev/gemini-api/docs/pricing?hl=es-419.

[83] OpenAI. OpenAI API Pricing; 2024. [consultado 5 de mayo de 2025]. [Internet]. Disponible en: https://platform.openai.com/docs/pricing.

Anexos

Tabla de metadatos

Cuadro 1: Metadatos de las guías clínicas utilizadas en el trabajo

Name	Year	DOI	PubMed
2019 Update of the ACR/SAA/SRTN Recommendations for the Treatment of Ankylosing Spondylitis and Nonradiographic Axial Spondyloarthritis	2019	10.1002/art.41042	https://pubmed.ncbi.nlm.nih.gov/31436026/
2022 ACR Guideline for the Prevention and Treatment of Glucocorticoid-Induced Osteoporosis	2022	10.1002/art.42646	https://pubmed.ncbi.nlm.nih.gov/37845798/
2020 ACR Guideline for the Management of Gout	2020	10.1002/acr.24180	https://pubmed.ncbi.nlm.nih.gov/32391934/
2023 ACR Guideline for Exercise, Rehabilitation, Diet, and Additional Integrative Interventions for Rheumatoid Arthritis	2023	10.1002/acr.25117	https://pubmed.ncbi.nlm.nih.gov/37227116/
2023 ACR/CHEST Guideline for the Screening and Monitoring of Interstitial Lung Disease in Systemic Autoimmune Rheumatic Diseases	2023	10.1002/art.42860	https://pubmed.ncbi.nlm.nih.gov/38973714/
2024 ACR Guideline for the Screening, Treatment, and Management of Lupus Nephritis	2024	10.1002/acr.21664	https://pubmed.ncbi.nlm.nih.gov/22556106/

Name	Year	DOI	PubMed
2020 ACR Guideline for the Management of Reproductive Health in Rheumatic and Musculoskeletal Diseases	2020	10.1002/acr.24130	https://pubmed.ncbi.nlm.nih.gov/32090466/
2021 ACR Guideline for the Treatment of Rheumatoid Arthritis	2021	10.1002/art.41752	https://pubmed.ncbi.nlm.nih.gov/34101376/
2023 ACR Guideline for Vaccinations in Patients With Rheumatic and Musculoskeletal Diseases	2023	10.1002/acr.25045	https://pubmed.ncbi.nlm.nih.gov/36597813/
2021 ACR/Vasculitis Foundation Guideline for the Management of Antineutrophil Cytoplasmic Antibody–Associated Vasculitis	2021	10.1002/art.41773	https://pubmed.ncbi.nlm.nih.gov/34235894/
2019 ACR/Arthritis Foundation Guideline for the Management of Osteoarthritis of the Hand, Hip, and Knee	2019	10.1002/art.41142	https://pubmed.ncbi.nlm.nih.gov/31908163/
2022 ACR/AAHKS Guideline for the Perioperative Management of Antirheumatic Medication in Patients Undergoing Elective Total Hip or Knee Arthroplasty	2022	10.1016/j.arth.2022.05.043	https://pubmed.ncbi.nlm.nih.gov/35732511/
2015 Recommendations for the Management of Polymyalgia Rheumatica	2015	10.1002/art.39333	https://pubmed.ncbi.nlm.nih.gov/26352874/

Name	Year	DOI	PubMed
2018 ACR/National Psoriasis Foundation Guideline for the Treatment of Psoriatic Arthritis	2018	10.1002/art.40726	https://pubmed.ncbi.nlm.nih.gov/30499246/
2023 ACR/AAHKS Clinical Practice Guideline for the Optimal Timing of Elective Total Hip or Knee Arthroplasty for Patients with Symptomatic Moderate to Severe Osteoarthritis or Osteonecrosis	2023	10.1002/acr.25175	https://pubmed.ncbi.nlm.nih.gov/37743767/
2022 ACR Guideline for Vaccinations in Patients with Rheumatic and Musculoskeletal Diseases	2023	10.1002/art.42386	https://pubmed.ncbi.nlm.nih.gov/36597810/
Clinical Practice Guidelines by IDSA, AAN, and ACR: 2020 Guidelines for the Prevention, Diagnosis, and Treatment of Lyme Disease	2020	10.1002/art.41562	https://pubmed.ncbi.nlm.nih.gov/33251716/
2021 ACR Guideline for the Treatment of Juvenile Idiopathic Arthritis	2021	10.1002/acr.24853	https://pubmed.ncbi.nlm.nih.gov/35233986/
2023 ACR/AAHKS Guideline for the Perioperative Management of Antirheumatic Medication in Patients Undergoing Elective Hip or Knee Arthroplasty	2023	10.1002/acr.25175	https://pubmed.ncbi.nlm.nih.gov/37743767/
2019 ACR Guideline for the Management of Osteoarthritis	2019	10.1002/art.41142	https://pubmed.ncbi.nlm.nih.gov/31908163/

Name	Year	DOI	PubMed
2023 ACR Guideline for the Management of Systemic Lupus Erythematosus	2023	10.1002/acr.25117	https://pubmed.ncbi.nlm.nih.gov/37227116/
2023 ACR Guideline for the Management of Antiphospholipid Syndrome	2023	10.1002/acr.25117	https://pubmed.ncbi.nlm.nih.gov/37227116/
2023 ACR Guideline for the Treatment of Sjögren's Syndrome	2023	10.1002/acr.25117	https://pubmed.ncbi.nlm.nih.gov/37227116/

Tabla de Prompts empleados

Cuadro 2: Prompts utilizados en los distintos modelos y tareas realizadas

Modelo	Prompt
Modelo Generativo	`Answer the following question: {question}.` **Role:** You are an AI assistant. Provide a concise and informative response based solely on your knowledge. If you do not know the answer, say that you don't know.
Modelo RAG	You are a medical AI assistant specialized in **rheumatology**, with a strong focus on clinical diagnosis. Act as a board-certified rheumatologist providing evidence-based, concise, and medically accurate answers. Here is the previous conversation history: {`chat_history`} You have access to the following clinical guidelines and medical literature: {`context`} Based on the above information, respond to the user's question: {`question`} Only use the provided information to answer. If there is insufficient data to give a reliable response, clearly state that the information is not available. Do not speculate or fabricate any content.
Generación de Preguntas	You are a medical expert. Based on the following clinical guideline for the pathology "{pathology_name}", generate 10 clear and specific questions that could be used to assess medical knowledge or review key concepts. **Content:** "{`text`} " Write a list of 20 questions:

Modelo	Prompt
Comparación Final	You are a Clinical QA Evaluation Expert AI. Your task is to compare two answers (A and B) to a clinical question, and evaluate them on specific dimensions. Evaluation Instructions: Carefully analyze both answers against the Full Guideline Context and the specific Retrieved Context. Provide scores on a scale of 1 (very poor) to 10 (excellent). If not a RAG model: 1. Assess Answer A (Baseline LLM): a. Relevance: How relevant is Answer A to the Question (Score 1-10)? b. Factual Accuracy: How factually accurate is Answer A compared to the Full Guideline Context? (Score 1-10) c. Completeness: How comprehensively does Answer A address the Question based on the Full Guideline Context? (Score 1-10) d. Conciseness: Is Answer A concise? (Score 1-10) 2. Assess Answer B (RAG LLM): a. Faithfulness: How faithful is Answer B to the Retrieved Context? Does it hallucinate information not found in the retrieved snippet? (Score 1-10) b. Relevance: How relevant is Answer B to the Question (Score 1-10)? c. Factual Accuracy: How factually accurate is Answer B compared to the Full Guideline Context? (Score 1-10) d. Completeness_Given_Retrieval: How comprehensively does Answer B address the question using only information found in the Retrieved Context? (Score 1-10) e. Completeness_Overall: How comprehensively does Answer B address the Question based on the Full Guideline Context? (Score 1-10) f. Conciseness: Is Answer B concise? (Score 1-10)

Modelo	Prompt

Comparación Final (cont.)	**3. Comparison and Justification:**
	a. Compare Answer A and Answer B based on all your assessments.
	b. Which answer is better overall for answering the specific Question accurately, safely, and reliably based on the full guideline? ('A', 'B', or 'Comparable')
	c. Provide a detailed step-by-step reasoning for your choice. Discuss the impact of RAG. Specifically comment on:
	i. Differences in Factual Accuracy and Safety
	ii. Whether Answer B's faithfulness to its limited Retrieved Context aligned with the overall guideline truth
	iii. If the Retrieved Context seemed sufficient/good based on comparing Answer B's Completeness_Given_Retrieval vs Completeness_Overall and its Faithfulness vs Factual Accuracy
	
	Use this exact format:
	
	`Gemini evaluation output:`
	`{format_a}`
	`{format_b}`
	`Which answer is better overall:`
	`[A/B/Comparable]`
	`Justification: [Brief explanation comparing factual accuracy, completeness, and retrieved context use]`
	
	And include: `{question}`, `{context_full}`, `{context_retrieved}`, `{responseA}`, `{responseB}`