Evaluación Comparativa de Configuraciones RAG para Documentos Científicos: Un Análisis de Arquitecturas y Modelos de Lenguaje

Sebastian Mindiola Garizado, Daniel Blanco Galán

***Abstract*—El avance de los sistemas de Generación Aumentada por Recuperación (RAG) ha estado dominado por soluciones propietarias, lo que limita su accesibilidad para investigadores y organizaciones con recursos restringidos. Este estudio presenta una evaluación exhaustiva de sistemas RAG construidos exclusivamente con herramientas de código abierto, demostrando su viabilidad tanto en entornos de prototipado como de producción. Se desarrollaron tres configuraciones arquitectónicas (básica, intermedia y avanzada), de las cuales se compararon formalmente las dos últimas empleando modelos generativos de última generación: LLaMA 3 70B, Qwen 3 32B y Gemini 2.0 Flash.**

La evaluación se realizó sobre el desafiante corpus PeerQA, diseñado para tareas de pregunta-respuesta en artículos científicos extensos, con una alta proporción de contenido irrelevante. La metodología incluyó dos enfoques: evaluación local con LLaMA 3.1 8B y evaluación remota vía API utilizando GPT-4o-mini, permitiendo contrastar precisión, costos y eficiencia operativa.

Los resultados muestran que la arquitectura avanzada superó consistentemente a la intermedia en todas las métricas, destacándose Qwen 3 32B con las mejores puntuaciones en *faithfulness* (0.863), *answer relevancy* (0.428), *context precision* (0.711) y *context recall* (0.611) vía API. Además, se evidenció que la evaluación remota es significativamente más eficiente: con un costo total de apenas $2.16 USD y tiempos de ejecución inferiores a 3 minutos por conjunto, frente a 40–60 minutos y requerimientos de GPU especializada (NVIDIA A5000, 24GB VRAM) en la evaluación local.

Este trabajo demuestra que es posible construir sistemas RAG robustos y competitivos empleando únicamente herramientas abiertas, lo que favorece escenarios con restricciones presupuestarias y necesidades de privacidad. Asimismo, se establece un marco metodológico práctico y reproducible para la evaluación de estos sistemas.

***Index Terms*—Generación Aumentada por Recuperación, RAG, Código Abierto, Modelos de Lenguaje Grande, Recuperación de Información, Procesamiento de Lenguaje Natural, PeerQA, Evaluación de Sistemas**

I. Introducción

En los últimos años, el vertiginoso crecimiento de los Grandes modelos de lenguaje (LLMs, por sus siglas en inglés) ha generado una revolución en el procesamiento de lenguaje natural(NLP), particularmente en tareas como la generación de texto, la interpretación semántica, la síntesis de información y el razonamiento automatizado [1], [2]. Modelos como GPT-4 [3], LLaMA-2 [4], T5 [5] o BERT [6] han demostrado una sorprendente capacidad para comprender y producir lenguaje humano, incluso en dominios complejos como la medicina, el derecho y la investigación científica [7]–[9]. Sin embargo, estas arquitecturas enfrentan limitaciones fundamentales, ya que su conocimiento está intrínsecamente restringido a los datos con los que fueron preentrenados. Esto ha dado lugar a fenómenos bien documentados como las ”alucinaciones”, es decir, respuestas plausibles pero erróneas o inventadas, así como a la incapacidad para la actualización a cambios recientes en el conocimiento científico [10], [11].

Este problema es particularmente grave en tareas altamente sensibles como las revisiones sistemáticas, donde el rigor metodológico y la veracidad de la información son esenciales para la generación de directrices clínicas, la elaboración de políticas públicas y la toma de decisiones basadas en la ciencia [12], [13]. Una revisión sistemática mal conducida puede inducir errores clínicos y promover intervenciones ineficaces o incluso dañinas [14]. Tradicionalmente, este tipo de estudios demanda una inversión de tiempo considerable, dependiendo del volumen de estudios a procesar y la experiencia de los investigadores [15]. Con el objetivo de acelerar esta tarea, el uso de modelos como GPT-3.5 Turbo ha sido explorado para la extracción de información desde artículos científicos, mostrando resultados mixtos: excelente rendimiento en datos fácilmente localizables (como país de estudio), pero limitaciones importantes en información compleja como razones de riesgo o criterios de inclusión/exclusión [16].

Ante esta dualidad entre capacidad y limitación, ha emergido una solución arquitectónica prometedora: el marco de generación aumentada con recuperación (Retrieval-Augmented Generation, RAG). RAG permite a los modelos consultar en tiempo real fuentes externas de conocimiento para contextualizar y fundamentar sus respuestas, superando así las restricciones impuestas por su entrenamiento estático [17], [18]. A diferencia de los LLMs tradicionales, un sistema RAG incorpora una etapa de recuperación de información desde una base de datos externa a partir de una consulta del usuario. La información recuperada se integra posteriormente en la generación del texto, dando lugar a respuestas más conherentes y precisas [19], [20].

El término RAG fue introducido por primera vez en un artículo de investigación de Facebook AI Research, en el cual se propone una arquitectura de tres etapas: ingesta, recuperación y generación. Esta propuesta inicial, conocida como Naive RAG, marcó el inicio del paradigma de generación aumentada por recuperación [21].

S. Mindiola Garizado y D. Blanco Galán estudiantes de la Universidad Nacional de Colombia, Sede de La Paz. Contacto: smindiola@unal.edu.co, danblanco@unal.edu.co.

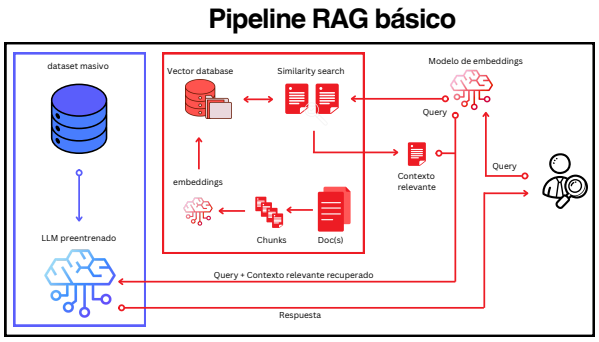

Fig. 1. **Flujo de trabajo de un sistema RAG.** El esquema ilustra los dos componentes clave de estos sistemas. En primer lugar, el *recuperador* (marcado en rojo), que incluye las etapas de extracción, limpieza y segmentación de la información. A partir de una consulta del usuario, este componente realiza la búsqueda en la base de conocimiento y entrega, junto con la consulta, un *contexto relevante*. Este contexto es procesado por el segundo componente, el *generador* (marcado en azul), donde el LLM genera una respuesta más precisa.

A partir de este punto, han surgido múltiples avances y variantes arquitectónicas que buscan mejorar la precisión, eficiencia y adaptabilidad de los sistemas RAG. Uno de los enfoques más destacados es el Modular RAG, que separa la arquitectura en componentes independientes (como el retriever, reranker, generador, etc.), permitiendo una configuración más flexible y personalizable del pipeline [22].

Por su parte, GraphRAG introduce representaciones estructuradas del conocimiento mediante grafos, donde las entidades y relaciones se preservan explícitamente. Esto permite realizar razonamiento multi-hop y mantener la coherencia semántica entre elementos relacionados [23], [24].

En la línea de la integración de múltiples fuentes, MuRAG o Multimodal RAG permite combinar tanto texto como datos no estructurados, como imágenes, tablas o figuras, ampliando la capacidad de recuperación y enriqueciendo la generación final con mayor contexto [25], [26].

Otro avance significativo es SELF-RAG, un sistema que introduce los denominados tokens de reflexión, que permiten al modelo tomar decisiones internas sobre cuándo necesita recuperar más información y cuándo debe criticar o revisar su propia salida para optimizarla [11].

Asimismo, Speculative RAG plantea un enfoque dual basado en dos modelos: un modelo especialista, más pequeño y enfocado en tareas específicas, que genera múltiples borradores de respuesta; y un modelo generalista, más grande, que verifica, refina y selecciona la mejor respuesta final [27].

Entre las propuestas más recientes se encuentra Agentic RAG, una familia de sistemas en los que cada etapa del pipeline es delegada a agentes autónomos capaces de planificar y razonar sobre la tarea específica en función de la consulta del usuario. Esto permite construir respuestas mucho más adaptadas y orientadas a la intención del usuario [28], [29].

En el ámbito científico, investigaciones recientes han comenzado a aplicar el marco RAG a la automatización de revisiones sistemáticas de literatura, una tarea que implica múltiples fases complejas como la búsqueda, el cribado, la extracción de datos y la síntesis de resultados. Estos trabajos han optimizado algunas de las arquitecturas existentes y propuesto nuevos enfoques adaptados a la estructura de los documentos científicos. Un ejemplo de ello es HipeRAG, que realiza un análisis estructural (parseo) de los documentos mediante una red neuronal entrenada para detectar automáticamente layouts complejos, permitiendo una conversión precisa a texto utilizable para recuperación y generación [30]. Otro caso relevante es DR-RAG, una arquitectura en dos fases: primero, realiza una recuperación tradicional de documentos estáticos relevantes (static-relevant documents), y posteriormente, combina estos con la consulta original para generar nuevas consultas expandidas. Esta segunda fase permite recuperar documentos dinámicamente relevantes (dynamic-relevant documents), que enriquecen el contexto aunque no coincidan directamente con la consulta inicial [31].

Además, sistemas como LitLLM [32], RefAI [33] y MEDRAG [34] han implementado arquitecturas RAG orientadas a la recuperación, evaluación y generación de conocimiento científico. Estos modelos emplean técnicas como extracción semántica de palabras clave, reordenamiento basado en contexto y generación guiada por evidencia. En ciertos casos, han superado a modelos generalistas como Gemini o ScholarAI, especialmente en tareas específicas del dominio.

Sin embargo, los sistemas enfocados en insights científicos tienden a apoyarse en configuraciones altamente especializadas, muchas veces basadas en modelos preentrenados para el dominio, como SciBERT [35] o ChemGPT [36], los cuales pueden interpretar con mayor fidelidad el lenguaje técnico.

A pesar de los crecientes avances en los sistemas RAG, la dimensión evaluativa sigue estando enfocada principalmente en dominios altamente específicos, como la ingeniería de software, las telecomunicaciones, la educación o la medicina [27], [37]–[39]. En estos casos, el énfasis suele recaer sobre corpus de evaluación diseñados exclusivamente para tareas particulares, lo que limita la posibilidad de generalización. Además, persiste una escasa comparación entre distintas configuraciones de sistemas RAG aplicadas al dominio académico en sentido amplio, lo que dificulta comprender qué combinaciones de componentes resultan realmente efectivas para tareas complejas como la síntesis y exploración de literatura científica.

En este contexto, se vuelve necesario avanzar hacia un marco experimental centrado en documentos científicos, que permita comparar y validar distintas configuraciones de sistemas RAG bajo criterios adecuados a las exigencias del dominio académico.

En este trabajo, se propone una evaluación comparativa de distintas configuraciones de un pipeline RAG, diferenciadas por el grado de sofisticación en sus componentes: desde una arquitectura básica, pasando por una configuración intermedia, hasta una versión avanzada. Todos los sistemas se construyen empleando exclusivamente herramientas y modelos de código abierto.

Aunque se diseñaron e implementaron las tres arquitecturas, el análisis empírico se centró en las configuraciones intermedia y avanzada. Esta decisión responde tanto a limitaciones de tiempo y recursos computacionales, como a la notable dis-

paridad estructural entre la versión simple (más experimental y desarrollada de forma individual) y las otras dos configuraciones, las cuales comparten una mayor complejidad, madurez tecnológica y comparabilidad.

Adicionalmente, se plantea una metodología de evaluación basada en RAGAS (Retrieval-Augmented Generation Assessment Suite) [40], la cual será analizada críticamente en función de su capacidad para capturar aspectos clave del lenguaje científico. Este estudio busca aportar evidencia sobre las fortalezas y limitaciones actuales de los sistemas RAG en contextos especializados, y así guiar futuros desarrollos hacia soluciones más precisas, verificables y eficientes para la síntesis automatizada de literatura científica.

II. Metodología

A. Enfoque general

Este estudio adopta un enfoque experimental comparativo para evaluar el desempeño de arquitecturas RAG aplicadas a la síntesis de literatura científica. La evaluación se realiza empleando el dataset público *PeerQA*, que contiene preguntas y respuestas humanas asociadas a artículos científicos. Este corpus introduce mayor complejidad en términos de diversidad temática, dificultad de las preguntas y volumen del conocimiento requerido. Se trata de un entorno más cercano a escenarios reales de producción y permite probar la escalabilidad y robustez de las arquitecturas evaluadas.

Cada sistema se somete a evaluación mediante el marco RAGAS, lo que permite cuantificar aspectos clave como fidelidad de las respuestas, relevancia respecto a la pregunta y precisión en la recuperación del contexto [41]. Asimismo, se incorporan dos variantes del entorno de evaluación: uno local, usando modelos abiertos, y otro basado en servicios vía API, lo cual enriquece el análisis al incorporar dos escenarios de evaluación diferentes.

B. Diseño experimental

Con el objetivo de analizar el impacto del diseño arquitectónico en el desempeño de sistemas RAG, se implementaron tres configuraciones diferenciadas por su nivel de complejidad: *simple*, *intermedia* y *avanzada*. A continuación, se describe detalladamente cada una de estas arquitecturas. Si bien todas fueron desarrolladas funcionalmente, el proceso de evaluación cuantitativa se centró exclusivamente en las dos últimas debido a su mayor potencial representativo y al uso más intensivo de técnicas semánticas, además de restricciones prácticas de tiempo y recursos.

1) Configuración simple: La Figura 2 ilustra la arquitectura del sistema RAG básico. Esta configuración emplea un flujo de procesamiento directo y sencillo. La extracción del texto se realiza con la librería `PyMuPDF`, que permite acceder al contenido textual de los documentos en formato PDF sin preservar estructura ni semántica [42].

La segmentación se realiza mediante *chunks* fijos de 1000 caracteres con un solapamiento de 200, valores estándar en literatura previa. Para la generación de representaciones vectoriales (embeddings), se utiliza el modelo `all-MiniLM-L6-v2`, con 22.7 millones de parámetros [43]. Este modelo genera vectores de 384 dimensiones, que se almacenan en memoria.

La recuperación se realiza mediante una búsqueda semántica basada en la similitud del coseno entre la consulta del usuario y los embeddings de los fragmentos de texto. Para cada consulta $\mathbf{q}$, se calcula la similitud con cada fragmento $\mathbf{d}_i$ de la base de datos vectorial y se seleccionan los k más relevantes según esta medida [44].

$$\text{sim}(\mathbf{q}, \mathbf{d}_i) = \frac{\mathbf{q} \cdot \mathbf{d}_i}{\|\mathbf{q}\|\|\mathbf{d}_i\|} \tag{1}$$

$$\text{TopK}(\mathbf{q}) = \underset{i \in \{1,\ldots,N\}}{\arg \text{top-k}} \, \text{sim}(\mathbf{q}, \mathbf{d}_i) \tag{2}$$

Estos fragmentos se concatenan y se proporcionan como contexto adicional al modelo LLM junto con la consulta original, utilizando una plantilla básica de prompt. Si bien esta arquitectura fue implementada funcionalmente, no fue incluida en la evaluación cuantitativa debido a su bajo nivel de sofisticación y su escasa comparabilidad directa con las otras dos variantes.

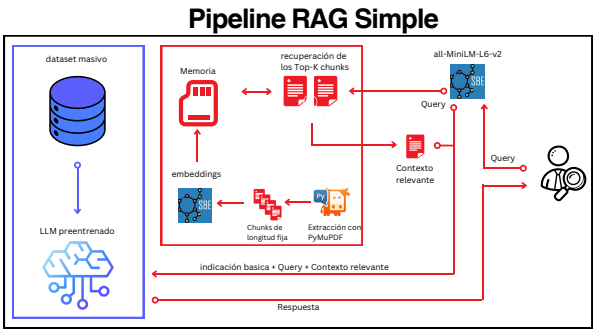

Fig. 2. **Flujo de trabajo de la configuración RAG simple.**

2) Configuración intermedia: La configuración intermedia (Figura 3) utiliza la herramienta `unstructured` para extraer contenido de documentos, preservando su jerarquía (títulos, subtítulos, tablas, imágenes) y asignando metadatos como la sección de origen (introducción, resultados, etc.) [45]. Los documentos se dividen en fragmentos (*chunks*) mediante `RecursiveCharacterTextSplitter` de LangChain, configurado con un tamaño de 512 caracteres y un solapamiento de 150 caracteres para garantizar coherencia semántica [46].

Los *embeddings* se generan con el modelo `all-mpnet-base-v2`, que cuenta con 109 millones de parámetros y produce vectores de 768 dimensiones [43]. Estos vectores se almacenan en una base vectorial `ChromaDB`, configurada con un cliente HTTP en un contenedor Docker, permitiendo filtrado por metadatos como el identificador del documento [47].

La recuperación combina búsqueda semántica, basada en la similitud de *embeddings*, con búsqueda por palabras clave mediante BM25, implementada a través de

`EnsembleRetriever` con pesos de 0.7 y 0.3, respectivamente [46]. La puntuación BM25 se define como:

$$\text{BM25}(q,d) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,d)\cdot(k_1+1)}{f(q_i,d)+k_1\cdot\left(1-b+b\cdot\frac{|d|}{\text{avgdl}}\right)} \quad (3)$$

$$\text{IDF}(q_i) = \log\left(\frac{N-df(q_i)+0.5}{df(q_i)+0.5}\right) \quad (4)$$

Los documentos recuperados y la consulta se integran en un *prompt* estructurado, orquestado por `LangChain`, para la generación final de la respuesta.

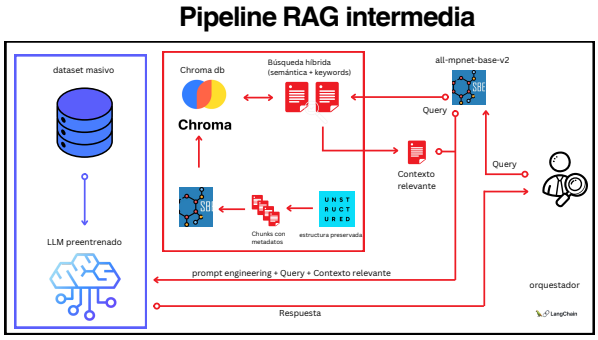

Fig. 3. **Flujo de trabajo de la configuración RAG intermedia.**

3) Configuración avanzada: La configuración avanzada (Figura 4) implementa un sistema RAG optimizado para maximizar la precisión y relevancia en la recuperación y generación de respuestas. Para la extracción de texto, se utiliza `Docling`, que preserva la jerarquía del documento (títulos, subtítulos, tablas, imágenes) y realiza un análisis lingüístico para detectar entidades, relaciones y roles semánticos, enriqueciendo los metadatos para procesos posteriores [48]. Los documentos se dividen en fragmentos semánticos mediante `spaCy`, configurado con el modelo `es_core_news_sm` y un tamaño máximo de 400 tokens por fragmento, garantizando coherencia contextual [49].

Los *embeddings* se generan con el modelo `BGE-M3`, que produce vectores de 1024 dimensiones y está optimizado para tareas multilingües y multimodales [50]. Estos vectores se almacenan en `Qdrant`, una base vectorial diseñada para búsqueda híbrida, configurada con métrica de distancia coseno [51]. La recuperación combina búsqueda semántica (basada en *embeddings*) y búsqueda léxica (BM25) mediante un enfoque híbrido, ponderando los puntajes con un 60% para la búsqueda semántica y un 40% para BM25. Además, se implementa un *cross-encoder* (`ms-marco-MiniLM-L-12-v2`) para reordenar los resultados, combinando los puntajes híbridos y del *cross-encoder* con pesos de 0.4 y 0.6, respectivamente [52].

Adicionalmente, se aplican técnicas de expansión de consultas, generando hasta tres variaciones de la consulta original (por ejemplo, sin *stopwords* o con sinónimos), y expansión de contexto, incorporando fragmentos adyacentes al texto recuperado para enriquecer la información disponible.

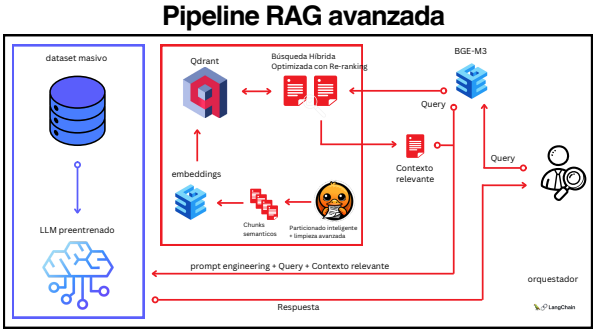

Fig. 4. **Flujo de trabajo de la configuración RAG avanzada.**

C. Modelos de lenguaje utilizados

Ya definidas las configuraciones, se observa que las variaciones se concentran principalmente en el componente recuperador. Sin embargo, el modelo generador también influye significativamente en la calidad y eficiencia de las respuestas. Con el fin de evaluar dicho impacto, se seleccionaron tres LLMs con características distintas en cuanto a velocidad, tamaño y especialización. Todos los modelos son accesibles a través de claves APIs con esquemas de uso gratuito limitado, lo que facilita su integración en entornos experimentales.

- **Gemini 2.0 Flash (Google)**
- **LLaMA 3 70B (Meta, vía Groq)**
- **Qwen 3 32B (Alibaba, vía OpenRouter)**

D. Corpus de evaluación

Para la evaluación del rendimiento de las distintas configuraciones RAG en el dominio académico, se utilizó el dataset **PeerQA**, un recurso especializado de Question Answering a nivel documental derivado de revisiones por pares científicas.

1) Corpus de evaluación: PeerQA contiene 579 pares pregunta-respuesta provenientes de 208 artículos académicos, principalmente en campos como Machine Learning y NLP, aunque también incluye dominios como geociencia y salud pública [53]. Las preguntas en PeerQA fueron extraídas directamente de los informes de revisión anónimos, mientras que las respuestas fueron proporcionadas y validadas por los propios autores de los artículos, lo que garantiza una correspondencia precisa con el contenido original (ground truth).

PeerQA representa un desafío significativo para sistemas RAG debido al tamaño promedio de los documentos (12,000 tokens) y la necesidad de procesar contexto extenso y variado, proporcionando un escenario de evaluación realista en condiciones de producción.

Para cada una de las configuraciones evaluadas se generaron conjuntos de datos en formato JSON, que contenían la pregunta original, la respuesta generada por la arquitectura RAG correspondiente, el contexto recuperado por el retriever y la respuesta de referencia (ground truth). Estos archivos fueron estructurados de acuerdo con los requisitos de entrada de RAGAS.

E. Métricas de evaluación

RAGAS proporciona una serie de métricas que permiten evaluar cada componente del pipeline RAG. En este trabajo se utilizarán las siguientes cuatro métricas principales:

- **Context Precision:** Mide la relación señal-ruido del contexto recuperado. Evalúa cuán relevante es el contenido proporcionado por el recuperador respecto a la consulta. Se calcula utilizando la consulta y los contextos seleccionados.
- **Context Recall:** Analiza si se obtuvo efectivamente toda la información requerida para responder a la pregunta. Esta métrica depende de los *ground truths* y se calcula comparando los contextos recuperados con las respuestas de referencia.
- **Faithfulness:** Evalúa la autenticidad factual de la respuesta producida, es decir, si las afirmaciones hechas por el modelo se sustentan explícitamente en los contextos proporcionados. Se calcula como la proporción de afirmaciones correctas dentro de la respuesta, considerando la pregunta, los contextos y la respuesta generada.
- **Answer Relevancy:** Evalúa qué tan relevante es la respuesta generada frente a la consulta original. No se enfoca en la veracidad, sino en si la respuesta aborda de forma completa y directa la pregunta. Se calcula usando la pregunta y la respuesta.

Cada una de estas métricas se expresa en una escala normalizada de 0 a 1, donde valores más cercanos a 1 indican un mejor desempeño en la dimensión evaluada.

F. Entornos de evaluación

Para llevar a cabo la evaluación automática de las respuestas generadas por cada arquitectura y modelo, se implementaron dos entornos de ejecución diferenciados:

- **Evaluación local:** se utilizó el modelo `llama3.1:8b` mediante el framework `Ollama`, ejecutado en una máquina local. Para la generación de embeddings requerida por RAGAS, se empleó el modelo `all-MiniLM-L6-v2` por su menor tamaño y velocidad de procesamiento.
- **Evaluación remota:** se empleó la implementación oficial de RAGAS en Python, configurada para usar como backend evaluador el modelo `gpt-4o-mini` a través del proveedor `OpenRouter`. En este caso, se utilizó el modelo de embeddings `all-mpnet-base-v2`.

Ambos entornos ejecutaron los mismos scripts base, adaptando únicamente el evaluador LLM y el modelo de embeddings utilizado. Esta decisión metodológica permite contrastar los resultados obtenidos bajo condiciones locales frente a condiciones remotas, y analizar el impacto de estas configuraciones sobre las métricas de fidelidad, relevancia y precisión del contexto.

III. Resultados

A continuación se presentan los resultados de la evaluación automática de las respuestas generadas por las arquitecturas intermedia y avanzada. La arquitectura simple no fue incluida en esta etapa, dado que su configuración sencilla distaba considerablemente de las otras dos y su adaptación al volumen de artículos del dataset requería un alto poder computacional y tiempo de procesamiento. Por tanto, no cumplía con los requisitos mínimos para una evaluación significativa bajo las métricas de RAGAS.

Los resultados se organizan en función del entorno de evaluación empleado (remoto y local), y reflejan el desempeño de cada arquitectura y modelo en términos de tres métricas principales: *faithfulness* (fidelidad), *answer relevancy* (relevancia de la respuesta) y *context precision* (precisión del contexto recuperado). Adicionalmente, se incluye la métrica *context recall* (recobrado del contexto relevante), la cual solo pudo obtenerse en el entorno remoto. En las pruebas locales, su cálculo generaba errores persistentes, probablemente relacionados con dependencias internas del modelo evaluador, por lo que fue descartada de dicha configuración.

A. Evaluación en entorno remoto

La evaluación remota se realizó mediante la API de OpenRouter, utilizando como modelo evaluador `gpt-4o-mini` provisto por OpenAI. Este entorno permitió calcular todas las métricas definidas por RAGAS, incluyendo *context recall*, cuya estimación no fue posible en la evaluación local debido a errores persistentes. A continuación se presentan los resultados obtenidos para cada combinación de arquitectura y modelo en este entorno.

Las siguientes abreviaturas se utilizan en las tablas de esta sección: *Faith.* (Faithfulness), *Ans. Rel.* (Answer Relevancy), *Ctx. Prec.* (Context Precision), *Ctx. Rec.* (Context Recall).

TABLE I
RESULTADOS DE LA EVALUACIÓN CON CONFIGURACIÓN INTERMEDIA

Modelo	Faith.	Ans. Rel.	Ctx. Prec.	Ctx. Rec.
LLaMA 3 70B	0.763	0.113	0.277	0.168
Qwen 3-32B	0.789	0.209	0.289	0.157
Gemini 2.0 Flash	0.750	0.237	0.272	0.141

La Tabla I presenta los resultados obtenidos con la configuración intermedia. En esta configuración, Qwen 3-32B se destaca con el mejor rendimiento en Faithfulness (0.789), seguido por LLaMA 3 70B (0.763). En cuanto a Answer Relevancy, Gemini 2.0 Flash obtiene el mejor resultado (0.237), mientras que LLaMA 3 70B presenta el rendimiento más bajo (0.113). Para las métricas de contexto, Qwen 3-32B lidera en Context Precision (0.289), mientras que LLaMA 3 70B supera ligeramente en Context Recall (0.168).

TABLE II
RESULTADOS DE LA EVALUACIÓN CON CONFIGURACIÓN AVANZADA

Modelo	Faith.	Ans. Rel.	Ctx. Prec.	Ctx. Rec.
LLaMA 3 70B	0.756	0.385	0.602	0.505
Qwen 3-32B	0.814	0.428	0.711	0.553
Gemini 2.0 Flash	0.863	0.370	0.688	0.611

La Tabla II muestra los resultados con la configuración avanzada, donde se observa una mejora significativa en todas las métricas respecto a la configuración intermedia. Gemini 2.0 Flash alcanza el mejor rendimiento en Faithfulness (0.863) y Context Recall (0.611), mientras que Qwen 3-32B se destaca en Answer Relevancy (0.428) y Context Precision (0.711). Es notable que LLaMA 3 70B, aunque mantiene un rendimiento competitivo en Faithfulness (0.756), presenta el menor rendimiento en Answer Relevancy (0.385) dentro de esta configuración.

Al comparar ambas configuraciones, se evidencia que la configuración avanzada supera consistentemente a la intermedia en todas las métricas evaluadas, con mejoras particularmente notables en Answer Relevancy y las métricas de contexto, donde los incrementos superan el 100% en varios casos.

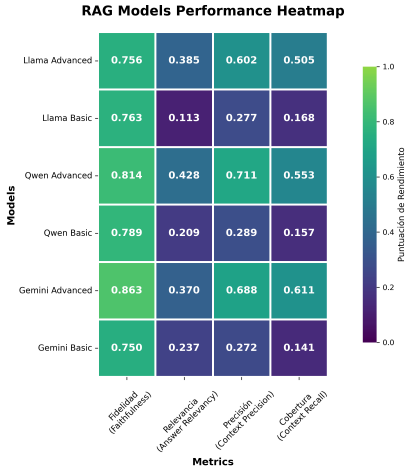

Fig. 5. Mapa de calor comparativo del rendimiento de los modelos evaluados. La escala de colores va desde morado (valores cercanos a 0, menor rendimiento) hasta verde (valores cercanos a 1, mejor rendimiento).

La Figura 5 presenta una visualización comparativa que facilita la identificación de patrones de rendimiento entre modelos y configuraciones. El contraste de colores revela inmediatamente que la configuración avanzada (tonos más verdes) supera consistentemente a la configuración intermedia (tonos más morados) en todas las métricas. Particularmente notable es el bajo rendimiento en Answer Relevancy para la configuración intermedia, donde todos los modelos muestran valores cercanos a 0 (tonos morados intensos), mientras que en la configuración avanzada esta métrica mejora significativamente. Qwen 3-32B destaca con los mejores valores en Faithfulness y Context Precision, evidenciados por los tonos verdes más intensos en estas métricas.

B. Evaluación en entorno local

La evaluación local se realizó utilizando el modelo `llama-3.1-8b` de Ollama como evaluador. En este entorno fue posible calcular las métricas de *faithfulness*, *answer relevancy* y *context precision*, aunque no se pudo obtener la métrica de *context recall* debido a limitaciones técnicas del entorno local. A continuación se presentan los resultados obtenidos para cada combinación de arquitectura y modelo en este entorno.

Las siguientes abreviaturas se utilizan en las tablas de esta sección: *Faith.* (Faithfulness), *Ans. Rel.* (Answer Relevancy), *Ctx. Prec.* (Context Precision).

TABLE III
RESULTADOS DE LA EVALUACIÓN LOCAL CON CONFIGURACIÓN INTERMEDIA

Modelo	**Faith.**	**Ans. Rel.**	**Ctx. Prec.**
Gemini 2.0 Flash	0.626	0.240	0.492
LLaMA 3 70B (Groq)	0.592	0.421	0.481
Qwen 3 32B	0.578	0.412	0.494

La Tabla III presenta los resultados obtenidos con la configuración intermedia en el entorno local. En esta configuración, Gemini 2.0 Flash se destaca con el mejor rendimiento en Faithfulness (0.626), mientras que LLaMA 3 70B (Groq) obtiene el mejor resultado en Answer Relevancy (0.421). En Context Precision, Qwen 3 32B alcanza el mejor resultado (0.494), aunque la diferencia con Gemini 2.0 Flash es mínima (0.492). Es notable que en esta configuración intermedia, Gemini 2.0 Flash presenta un valor considerablemente más bajo en Answer Relevancy (0.240) comparado con los otros modelos.

TABLE IV
RESULTADOS DE LA EVALUACIÓN LOCAL CON CONFIGURACIÓN AVANZADA

Modelo	**Faith.**	**Ans. Rel.**	**Ctx. Prec.**
Gemini 2.0 Flash	0.656	0.418	0.665
LLaMA 3 70B	0.642	0.548	0.590
Qwen 3 32B	0.725	0.565	0.669

La Tabla IV muestra los resultados con la configuración avanzada en el entorno local, donde se observa un comportamiento claramente superior. Qwen 3 32B se destaca con el mejor rendimiento en todas las métricas evaluadas: Faithfulness (0.725), Answer Relevancy (0.565) y Context Precision (0.669). LLaMA 3 70B obtiene el segundo mejor resultado en Answer Relevancy (0.548), mientras que Gemini 2.0 Flash presenta un rendimiento competitivo en Context Precision (0.665), muy cercano al líder.

Al comparar ambas configuraciones en el entorno local, se observa que la configuración avanzada supera significativamente a la intermedia en la mayoría de las métricas, particularmente en Faithfulness y Answer Relevancy. Este patrón muestra una clara mejora del rendimiento cuando se utiliza la arquitectura RAG más sofisticada, especialmente notable en el caso de Qwen 3 32B que logra los mejores resultados absolutos en todas las métricas con la configuración avanzada.

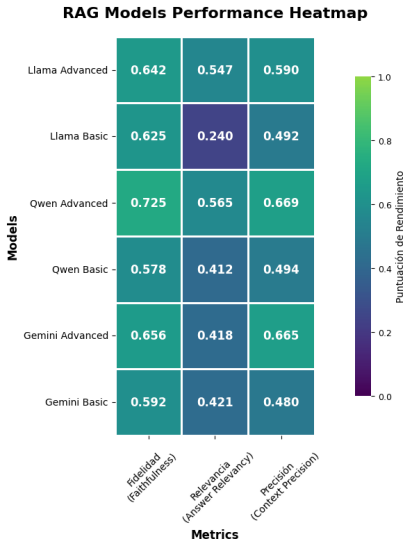

Fig. 6. Mapa de calor comparativo del rendimiento de los modelos evaluados en entorno local. La escala de colores va desde morado (valores cercanos a 0, menor rendimiento) hasta verde (valores cercanos a 1, mejor rendimiento).

La Figura 6 presenta una visualización comparativa del rendimiento en el entorno local que revela patrones interesantes. A diferencia de los resultados del entorno remoto, aquí la configuración avanzada muestra tonos más verdes en general, indicando un mejor rendimiento comparativo. Qwen 3 32B destaca particularmente en la configuración avanzada con los valores más altos en todas las métricas, mientras que en la configuración intermedia Gemini 2.0 Flash muestra una notable caída en Answer Relevancy.

C. Comparación entre entornos de evaluación

La evaluación de los sistemas RAG se realizó en dos entornos distintos: local utilizando llama-3.1-8b como modelo evaluador, y remoto mediante API utilizando GPT-4o-mini. Esta comparación revela diferencias significativas tanto en aspectos operacionales como en los resultados obtenidos.

1) Aspectos operacionales: El entorno local presentó limitaciones considerables en términos de eficiencia temporal. Cada archivo JSON evaluado requería aproximadamente 40 minutos a 1 hora de procesamiento, considerando únicamente tres métricas (faithfulness, answer relevancy y context precision). Esto ocurrió a pesar de contar con recursos computacionales significativos, incluyendo una tarjeta gráfica NVIDIA A5000 con 24GB de VRAM, lo que evidencia las limitaciones inherentes del modelo evaluador local.

Adicionalmente, el entorno local experimentó problemas frecuentes relacionados con el formato de salida del modelo evaluador. Estos errores se manifestaban como excepciones del tipo `OutputParserException`, generadas cuando el modelo no producía respuestas en el formato JSON esperado. Un ejemplo típico de estos errores fue:

```
Batch 121/300: 0%| | 0/1 [00:00<?, ?it/s]
Exception raised in Job[120]:
OutputParserException(Invalid json output:
The provided text fragments do not contain
information about the number of distinct
phrases used for phrase addition...)
```

Estos errores impactaron negativamente la evaluación, ya que RAGAS asigna automáticamente una puntuación de cero cuando no puede procesar correctamente la respuesta del modelo evaluador.

En contraste, la evaluación mediante API demostró una eficiencia superior, con tiempos de procesamiento de aproximadamente 3 minutos por dataset, sin experimentar los problemas de formato observados en el entorno local. Esta diferencia de rendimiento se debe tanto a la optimización de los servicios API como a la mayor robustez del modelo GPT-4o-mini en la generación de respuestas estructuradas.

Consideraciones económicas y de accesibilidad: La evaluación completa mediante API tuvo un costo total de $2.16 USD, incluyendo la evaluación de todos los modelos en ambas configuraciones y las cuatro métricas. Este costo relativamente bajo, combinado con la capacidad de ejecutar las evaluaciones en cualquier máquina sin requerimientos específicos de hardware, representa una ventaja significativa sobre el entorno local que requiere recursos computacionales especializados como GPUs de alta gama. Esta accesibilidad democratiza el proceso de evaluación de sistemas RAG, permitiendo que investigadores con recursos limitados puedan realizar evaluaciones comprehensivas sin la necesidad de invertir en infraestructura computacional costosa.

2) Análisis de las discrepancias en resultados: La comparación entre entornos revela diferencias sustanciales en las puntuaciones obtenidas, lo que sugiere que el modelo evaluador influye significativamente en los criterios de evaluación aplicados.

Diferencias en faithfulness: El entorno local mostró valores generalmente menores (rango: 0.578-0.725) comparado con el entorno remoto (rango: 0.750-0.863). Particularmente notable es el caso de Gemini 2.0 Flash, que obtuvo 0.656 localmente versus 0.863 remotamente en configuración avanzada, representando una diferencia del 31.6%. Esto sugiere que GPT-4o-mini aplica criterios menos estrictos para evaluar la fidelidad de las respuestas al contexto proporcionado.

Diferencias en answer relevancy: Esta métrica presenta el patrón inverso más pronunciado. Los valores locales (rango: 0.240-0.565) superan consistentemente a los remotos (rango: 0.113-0.428). Un ejemplo destacado es Qwen 3 32B en configuración avanzada: 0.565 local versus 0.428 remoto. Esta discrepancia indica que GPT-4o-mini emplea criterios más rigurosos para determinar la relevancia de las respuestas, posiblemente considerando aspectos semánticos más profundos o requiriendo mayor precisión en la correspondencia pregunta-respuesta.

Diferencias en context precision: Los valores locales (rango: 0.480-0.669) son comparables pero generalmente menores que los remotos (rango: 0.272-0.711). Sin embargo, las diferencias son menos pronunciadas que en las otras métricas, sugiriendo mayor consistencia en los criterios de evaluación para esta métrica específica.

Context recall: Esta métrica solo fue calculable en el entorno remoto debido a limitaciones técnicas del entorno

local, proporcionando información adicional valiosa sobre el rendimiento de recuperación de contexto (rango: 0.141-0.611).

3) Consistencia en patrones de rendimiento: A pesar de las diferencias absolutas en las puntuaciones, ambos entornos mantienen cierta consistencia en los patrones relativos de rendimiento. Qwen 3 32B emerge como el modelo más consistente, liderando en múltiples métricas en ambos entornos, especialmente en configuración avanzada. Asimismo, la superioridad de la configuración avanzada sobre la básica se mantiene en ambos entornos, validando la efectividad de las mejoras arquitectónicas implementadas.

4) Implicaciones para la evaluación de sistemas RAG: Los resultados evidencian que la elección del modelo evaluador constituye un factor crítico que trasciende aspectos meramente operacionales, influyendo directamente en la interpretación cuantitativa del rendimiento del sistema. La mayor rigurosidad de GPT-4o-mini en answer relevancy podría ser más apropiada para aplicaciones que requieren alta precisión semántica, mientras que los criterios menos estrictos del modelo local podrían ser suficientes para evaluaciones exploratorias.

Esta variabilidad subraya la importancia de establecer benchmarks estandarizados y considerar múltiples modelos evaluadores para obtener una perspectiva más robusta del rendimiento de sistemas RAG. Adicionalmente, la viabilidad económica y técnica de la evaluación mediante API ($2.16 USD por evaluación completa, ejecutable en cualquier hardware) la posiciona como una alternativa altamente atractiva para la investigación en RAG, especialmente en contextos donde la reproducibilidad, comparabilidad de resultados y accesibilidad de recursos son fundamentales.

IV. Conclusiones

Este estudio demuestra la viabilidad de implementar sistemas RAG robustos utilizando exclusivamente herramientas de código abierto, proporcionando una alternativa accesible y efectiva para entornos prototípicos y de investigación. Los resultados obtenidos revelan aspectos importantes tanto sobre el rendimiento de diferentes configuraciones como sobre las metodologías de evaluación empleadas.

A. Viabilidad de sistemas RAG con herramientas de código abierto

La investigación confirma que es posible desarrollar sistemas RAG funcionales y competitivos mediante herramientas completamente abiertas. Aunque la configuración más simple no fue evaluada formalmente en este estudio, las pruebas preliminares demostraron su efectividad para tareas básicas como procesamiento de PDFs, síntesis de información y respuesta a consultas sencillas, validando su utilidad para casos de uso fundamentales.

Las configuraciones intermedia y avanzada enfrentaron el desafío significativo que representa el dataset PeerQA, el cual simula condiciones más realistas y cercanas a entornos de producción. Este dataset, caracterizado por una base de conocimiento extensa y alta presencia de información irrelevante, demandó mayor precisión en los mecanismos de recuperación y generación, reflejando fielmente los retos que enfrentan los sistemas RAG en aplicaciones del mundo real.

B. Rendimiento diferencial de las configuraciones

Los resultados confirman las expectativas teóricas respecto al rendimiento superior de la configuración avanzada. Esta configuración demostró consistentemente mejores métricas en ambos entornos de evaluación, validando la efectividad de las técnicas avanzadas de recuperación implementadas, incluyendo el reranking con modelos de embedding especializados y estrategias de chunking más sofisticadas.

La configuración intermedia, aunque mostró un rendimiento inferior, logró resultados aceptables considerando su menor complejidad computacional, posicionándose como una opción equilibrada para aplicaciones con restricciones de recursos pero que requieren un rendimiento superior al básico.

C. Satisfacción de objetivos y aplicabilidad práctica

Los resultados obtenidos proporcionan una satisfacción considerable respecto a los objetivos planteados inicialmente. La demostración exitosa de que herramientas de código abierto pueden generar sistemas RAG potentes abre oportunidades significativas para investigadores, desarrolladores y organizaciones que requieren soluciones de generación aumentada por recuperación.

Particularmente relevante es la aplicabilidad de estos sistemas en entornos prototípicos y personales, donde las ventajas en términos de privacidad de datos representan un valor agregado considerable. La capacidad de procesar información sensible localmente, sin depender de servicios externos, constituye una ventaja estratégica para organizaciones con requerimientos estrictos de confidencialidad.

D. Recomendaciones sobre metodologías de evaluación

La comparación entre entornos de evaluación local y remoto reveló diferencias sustanciales que trascienden aspectos meramente técnicos. La evaluación mediante API emergió como la opción más práctica y recomendable, ofreciendo:

- **Eficiencia operacional superior**: Reducción de tiempos de evaluación de horas a minutos
- **Viabilidad económica**: Costo total de $2.16 USD para evaluación completa
- **Accesibilidad universal**: Ejecución en cualquier hardware sin requerimientos especiales
- **Robustez técnica**: Ausencia de problemas de formato y mayor estabilidad

Los autores consideran que la evaluación vía API representa una inversión altamente rentable que democratiza el acceso a evaluaciones comprehensivas de sistemas RAG, eliminando barreras técnicas y económicas que podrían limitar la investigación en esta área.

E. Perspectivas futuras

Los resultados obtenidos establecen una base sólida para futuras investigaciones en sistemas RAG con herramientas de código abierto. Las configuraciones desarrolladas pueden servir como punto de partida para optimizaciones adicionales, incluyendo la exploración de nuevos modelos de embedding,

técnicas de chunking adaptativas y estrategias de reranking más sofisticadas.

La metodología de evaluación establecida, particularmente la evaluación vía API, proporciona un framework reproducible y accesible para futuras comparaciones, facilitando el avance colaborativo en el campo de la generación aumentada por recuperación.

En conjunto, este estudio demuestra que la democratización de tecnologías RAG avanzadas es no solo posible sino práctica, abriendo nuevas oportunidades para la innovación y aplicación de estas tecnologías en contextos diversos y con recursos variados.

REFERENCES

[1] B. Han, T. Susnjak, and A. Mathrani, "Automating Systematic Literature Reviews with Retrieval-Augmented Generation: A Comprehensive Overview," *Applied Sciences*, vol. 14, no. 19, p. 9103, Oct. 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/19/9103

[2] M. Cremaschi, F. D'Adda, and A. Maurino, "stEELlm: An LLM for Generating Semantic Annotations of Tabular Data," *ACM Transactions on Intelligent Systems and Technology*, 2 2025. [Online]. Available: https://doi.org/10.1145/3719206

[3] R. Mao, G. Chen, X. Zhang, F. Guerin, and E. Cambria, "GPTEval: A Survey on Assessments of ChatGPT and GPT-4," *arXiv (Cornell University)*, 1 2023. [Online]. Available: https://arxiv.org/abs/2308.12488

[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open Foundation and Fine-Tuned Chat Models," Jul. 2023, arXiv:2307.09288. [Online]. Available: http://arxiv.org/abs/2307.09288

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423/

[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[7] A. Y. Aytar, K. Kilic, and K. Kaya, "A Retrieval-Augmented Generation Framework for Academic Literature Navigation in Data Science," *arXiv (Cornell University)*, 12 2024. [Online]. Available: http://arxiv.org/abs/2412.15404

[8] P. J. Rathod, "Efficient Usage of RAG Systems in the World of LLMs," *International Journal For Multidisciplinary Research*, vol. 6, no. 4, 7 2024. [Online]. Available: https://doi.org/10.36948/ijfmr.2024.v06i04.36364

[9] Y. S. Low, M. L. Jackson, R. J. Hyde, R. E. Brown, N. M. Sanghavi, J. D. Baldwin, C. W. Pike, J. Muralidharan, G. Hui, N. Alexander, H. Hassan, R. V. Nene, M. Pike, C. J. Pokrzywa, S. Vedak, A. P. Yan, D.-H. Yao, A. R. Zipursky, C. Dinh, P. Ballentine, D. C. Derieg, V. Polony, R. N. Chawdry, J. Davies, B. B. Hyde, N. H. Shah, and S. Gombar, "Answering real-world clinical questions using large language model, retrieval-augmented generation, and agentic systems," *Digital Health*, vol. 11, 5 2025. [Online]. Available: https://doi.org/10.1177/20552076251348850

[10] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, and S. Shi, "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models," Sep. 2023, arXiv:2309.01219 version: 2. [Online]. Available: http://arxiv.org/abs/2309.01219

[11] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," Oct. 2023, arXiv:2310.11511. [Online]. Available: http://arxiv.org/abs/2310.11511

[12] L. Uttley, D. S. Quintana, P. Montgomery, C. Carroll, M. J. Page, L. Falzon, A. Sutton, and D. Moher, "The problems with systematic reviews: a living systematic review," *Journal of Clinical Epidemiology*, vol. 156, pp. 30–41, Apr. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0895435623000112

[13] J. Yang, H. Xu, S. Mirzoyan, T. Chen, Z. Liu, Z. Liu, W. Ju, L. Liu, Z. Xiao, M. Zhang, and S. Wang, "Poisoning medical knowledge using large language models," *Nature Machine Intelligence*, vol. 6, no. 10, pp. 1156–1168, Oct. 2024. [Online]. Available: https://www.nature.com/articles/s42256-024-00899-3

[14] S. H. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw, "Clinical guidelines: Potential benefits, limitations, and harms of clinical guidelines," *BMJ*, vol. 318, no. 7182, pp. 527–530, Feb. 1999. [Online]. Available: https://www.bmj.com/lookup/doi/10.1136/bmj.318.7182.527

[15] B. Nussbaumer-Streit, M. Ellen, I. Klerings, R. Sfetcu, N. Riva, M. Mahmić-Kaknjo, G. Poulentzas, P. Martinez, E. Baladia, L. Ziganshina, M. Marqués, L. Aguilar, A. Kassianos, G. Frampton, A. Silva, L. Affengruber, R. Spjker, J. Thomas, R. Berg, M. Kontogiani, M. Sousa, C. Kontogiorgis, and G. Gartlehner, "Resource use during systematic review production varies widely: a scoping review," *Journal of Clinical Epidemiology*, vol. 139, pp. 287–296, Nov. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0895435621001712

[16] C. C. Y. Gue, N. D. A. Rahim, W. Rojas-Carabali, R. Agrawal, P. Rk, J. Abisheganaden, and W. F. Yip, "Evaluating the OpenAI's GPT-3.5 Turbo's performance in extracting information from scientific articles on diabetic retinopathy," *Systematic Reviews*, vol. 13, no. 1, p. 135, May 2024. [Online]. Available: https://systematicreviewsjournal.biomedcentral.com/articles/10.1186/s13643-024-02523-2

[17] Y. Huang and J. Huang, "A Survey on Retrieval-Augmented Text Generation for Large Language Models," *arXiv (Cornell University)*, 4 2024. [Online]. Available: http://arxiv.org/abs/2404.10981

[18] S. Innovation, "Implementing a Retrieval-Augmented Generation (RAG) System with OpenAI's API using LangChain," Nov. 2023. [Online]. Available: https://bit.ly/44wgIgC

[19] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv (Cornell University)*, 1 2023. [Online]. Available: https://arxiv.org/abs/2312.10997

[20] Z. Li, Z. Wang, W. Wang, K. Hung, H. Xie, and F. L. Wang, "Retrieval-Augmented Generation for Educational Application: A Systematic Survey," *Computers and Education Artificial Intelligence*, p. 100417, 5 2025. [Online]. Available: https://doi.org/10.1016/j.caeai.2025.100417

[21] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 5 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[22] Y. Gao, Y. Xiong, M. Wang, and H. Wang, "Modular RAG: Transforming RAG Systems into LEGO-like Reconfigurable Frameworks," *arXiv (Cornell University)*, 7 2024. [Online]. Available: http://arxiv.org/abs/2407.21059

[23] Z. Zhu, T. Huang, K. Wang, J. Ye, X. Chen, and S. Luo, "Graph-based Approaches and Functionalities in Retrieval-Augmented Generation: A Comprehensive Survey," 4 2025. [Online]. Available: https://arxiv.org/abs/2504.10499v1

[24] B. Sarmah, D. Mehta, B. Hall, R. Rao, S. Patel, and S. Pasquali, "HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction," *ICAIF '24: Proceedings of the 5th ACM International Conference on AI in Finance*, pp. 608–616, 11 2024. [Online]. Available: https://doi.org/10.1145/3677052.3698671

[25] K. Thiyagarajan, "Multimodal RAG for Enhanced Information Retrieval and Generation in Retail," *2025 International Conference on Visual Analytics and Data Visualization (ICVADV)*, pp. 102–106, 3 2025. [Online]. Available: https://doi.org/10.1109/icvadv63329.2025.10961713

[26] W. Chen, H. Hu, X. Chen, P. Verga, and W. Cohen, "MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text," *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1 2022. [Online]. Available: https://doi.org/10.18653/v1/2022.emnlp-main.375

[27] S. Wang, J. L. S. Song, J. Cheng, Y. Fu, P. Guo, K. Fang, Y. Zhu, and Z. Dou, "DomainRAG: A Chinese Benchmark for Evaluating Domain-specific Retrieval-Augmented Generation," *arXiv (Cornell University)*, 6 2024. [Online]. Available: https://arxiv.org/abs/2406.05654

[28] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei, "Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG," *arXiv (Cornell University)*, 1 2025. [Online]. Available: http://arxiv.org/abs/2501.09136

[29] X. Xu, D. Zhang, Q. Liu, Q. Lu, and L. Zhu, "Agentic RAG with Human-in-the-Retrieval," 3 2025, pp. 498–502. [Online]. Available: https://doi.org/10.1109/icsa-c65153.2025.00074

[30] O. Gokdemir, C. Siebenschuh, A. Brace, A. Wells, B. Hsu, K. Hippe, P. Setty, A. Ajith, J. G. Pauloski, V. Sastry, S. Foreman, H. Zheng, H. Ma, B. Kale, N. Chia, T. Gibbs, M. Papka, T. Brettin, F. Alexander, A. Anandkumar, I. Foster, R. Stevens, V. Vishwanath, and A. Ramanathan, "HiPerRAG: High-Performance Retrieval Augmented Generation for Scientific Insights," *PASC '25: Proceedings of the Platform for Advanced Scientific Computing Conference*, pp. 1–13, 6 2025. [Online]. Available: https://doi.org/10.1145/3732775.3733586

[31] Z. Hei, W. Liu, W. Ou, J. Qiao, J. Jiao, Z. Zhu, and G. Song, "DR-RAG: Applying Dynamic Document Relevance to Retrieval-Augmented Generation for Question-Answering," *arXiv (Cornell University)*, 6 2024. [Online]. Available: https://arxiv.org/abs/2406.07348

[32] S. Agarwal, G. Sahu, A. Puri, I. H. Laradji, K. D. Dvijotham, J. Stanley, L. Charlin, and C. Pal, "LitLLM: A Toolkit for Scientific Literature Review," Mar. 2025, arXiv:2402.01788. [Online]. Available: http://arxiv.org/abs/2402.01788

[33] Y. Li, J. Zhao, M. Li, Y. Dang, E. Yu, J. Li, Z. Sun, U. Hussein, J. Wen, A. M. Abdelhameed, J. Mai, S. Li, Y. Yu, X. Hu, D. Yang, J. Feng, Z. Li, J. He, W. Tao, T. Duan, Y. Lou, F. Li, and C. Tao, "RefAI: a GPT-powered retrieval-augmented generative tool for biomedical literature recommendation and summarization," *Journal of the American Medical Informatics Association*, vol. 31, no. 9, pp. 2030–2039, Sep. 2024. [Online]. Available: https://academic.oup.com/jamia/article/31/9/2030/7690757

[34] G. Xiong, Q. Jin, Z. Lu, and A. Zhang, "Benchmarking Retrieval-Augmented Generation for Medicine," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6233–6251. [Online]. Available: https://aclanthology.org/2024.findings-acl.372/

[35] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3615–3620. [Online]. Available: https://aclanthology.org/D19-1371/

[36] N. C. Frey, R. Soklaski, S. Axelrod, S. Samsi, R. Gómez-Bombarelli, C. W. Coley, and V. Gadepally, "Neural scaling of deep chemical models," *Nature Machine Intelligence*, vol. 5, no. 11, pp. 1297–1305, Nov. 2023. [Online]. Available: https://www.nature.com/articles/s42256-023-00740-3

[37] S. Simon, A. Mailach, J. Dorn, and N. Siegmund, "A Methodology for Evaluating RAG Systems: A Case Study On Configuration Dependency Validation," *arXiv (Cornell University)*, 10 2024. [Online]. Available: http://arxiv.org/abs/2410.08801

[38] S. Roychowdhury, S. Soman, H. G. Ranjani, N. Gunda, V. Chhabra, and S. K. Bala, "Evaluation of RAG Metrics for Question Answering in the Telecom Domain," *arXiv (Cornell University)*, 7 2024. [Online]. Available: http://arxiv.org/abs/2407.12873

[39] A. Thomo, "PubMed Retrieval with RAG Techniques," *Studies in health technology and informatics*, 8 2024. [Online]. Available: https://doi.org/10.3233/shti240498

[40] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," *arXiv (Cornell University)*, 1 2023. [Online]. Available: https://arxiv.org/abs/2309.15217

[41] ExplodingGradients, "Ragas: Supercharge your llm application evaluations," https://github.com/explodinggradients/ragas, 2024.

[42] PyMuPDF developers, "PyMuPDF Documentation," https://pymupdf.readthedocs.io/en/latest/, 2025.

[43] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1 2019. [Online]. Available: https://aclanthology.org/D19-1410/

[44] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[45] Unstructured.io developers, "Unstructured: Open-source library for data pre-processing," https://unstructured-io.github.io/unstructured/, 2024, accedido: 28 de junio de 2025.

[46] LangChain developers, "LangChain Documentation," https://www.langchain.com/, 2024, accedido: 20 de junio de 2025.

[47] Chroma Team, "Deploy with docker – chroma documentation," 2024. [Online]. Available: https://docs.trychroma.com/guides/deploy/docker

[48] N. Livathinos, C. Auer, M. Lysak, A. Nassar, M. Dolfi, P. Vagenas, C. B. Ramis, M. Omenetti, K. Dinkla, Y. Kim, S. Gupta, D. L. R. Teixeira, V. Weber, L. Morin, I. Meijer, V. Kuropiatnyk, and P. W. J. Staar, "Docling: An Efficient Open-Source Toolkit for AI-driven Document Conversion," *arXiv (Cornell University)*, 1 2025. [Online]. Available: http://arxiv.org/abs/2501.17887

[49] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.

[50] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity text embeddings through Self-Knowledge distillation," *arXiv (Cornell University)*, 2 2024. [Online]. Available: https://arxiv.org/abs/2402.03216

[51] Qdrant developers, "Qdrant Documentation," https://qdrant.tech/documentation/, 2025, accedido: 20 de junio de 2025.

[52] O. Sanseviero, "Sentence embeddings: Cross-encoders and re-ranking," 2024, blog post. [Online]. Available: https://osanseviero.github.io/hackerllama/blog/posts/sentence_embeddings2/

[53] T. Baumgärtner, T. Briscoe, and I. Gurevych, "PeerQA: A Scientific Question Answering Dataset from Peer Reviews," *arXiv (Cornell University)*, 2 2025. [Online]. Available: http://arxiv.org/abs/2502.13668