Comparación de modelos generativos compactos para respuesta automática en español mediante RAG

Comparison of Compact Generative Models for Automatic Question Answering in Spanish via Retrieval-Augmented Generation

Jean Phol Curi Garrafa [A], Victor R. Ortega Marocho [B] y Wilson Mamani Rodrigo[C]

Resumen— Este estudio compara cinco modelos generativos compactos (≤ 8 mil millones de parámetros) para respuesta automática en español integrados a un esquema de generación aumentada con recuperación (RAG) y ejecutados localmente. Se evalúa la calidad mediante F1, BLEU-4 y un juicio semántico externo (LLM-Judge), junto con indicadores de eficiencia (latencia P95, memoria, GPU/CPU). Los resultados muestran que Mistral 7B alcanza el mejor desempeño medio en F1 y valoración global, mientras que OpenHermes 7B ofrece una precisión prácticamente equivalente con la menor huella de memoria. Zephyr 7B-β destaca en documentos extensos y Phi-3 Mini reduce las colas de latencia bajo condiciones adversas. Un análisis de Pareto F1–RAM identifica a Mistral 7B y OpenHermes 7B como soluciones no dominadas, proporcionando pautas de selección según el objetivo operativo (precisión absoluta vs. eficiencia de recursos). El trabajo aporta una comparación reproducible en español bajo RAG y criterios accionables para despliegues locales.

Palabras clave: Español, evaluación, modelos compactos, rag.

C&T Riqchary

Revista de Investigación en Ciencia y Tecnología
ISSN: 2810-8124 (en línea) / ISSN: 2706-543x
Universidad Nacional Micaela Bastidas de Apurimac – Perú

Vol. 7 Núm. 2 (2025) - Publicado: 19/08/25 - Indexaciones
Número: doi.org/10.57166/riqchary/v7.n2.2025
Páginas: 9- 18 | Recibido 01/01/2025 ; Aceptado 01/02/2025

doi.org/10.57166/riqchary.v7.n2.2025.2

Autores:

A. ORCID iD https://orcid.org/0009-0006-5536-7055
Jean Phol A. Curi Garrafa, Universidad Nacional Micaela Bastidas de Apurímac 191204@unamba.edu.pe

B. ORCID iD https://orcid.org/0009-0006-7868-5507
Victor R. Ortega Marocho, Universidad Nacional Micaela Bastidas de Apurímac 191225@unamba.edu.pe

C. ORCID iD https://orcid.org/0000-0003-3901-0268
Wilson Mamani Rodrigo, trabaja en el Departamento de Ingeniería de la Universidad Nacional Micaela Bastidas de Apurímac wmamanir@unamba.edu.pe

Abstract— This study compares five compact generative models (≤ 8 billion parameters) for Spanish question answering under a retrieval-augmented generation (RAG) pipeline executed locally. We assess response quality using F1, BLEU-4, and an external semantic judge (LLM-Judge), alongside efficiency indicators (P95 latency, memory, GPU/CPU). Results show Mistral 7B achieves the highest average F1 and semantic scores, whereas OpenHermes 7B attains nearly identical accuracy with the lowest memory footprint. Zephyr 7B-β performs well on very long documents, and Phi-3 Mini minimizes tail latency under adverse conditions. A Pareto analysis of F1–RAM identifies Mistral 7B and OpenHermes 7B as non-dominated solutions, yielding practical guidelines depending on operational goals (maximum accuracy vs. resource efficiency). The paper contributes a reproducible Spanish-language comparison under RAG and actionable criteria for local deployments.

Keywords: Compact models, evaluation, rag, spanish.

1 INTRODUCCIÓN

En el Procesamiento de Lenguaje Natural, los modelos generativos "compactos" han cobrado protagonismo porque permiten un equilibrio razonable entre calidad de las respuestas y costo computacional, algo especialmente valioso cuando el objetivo es operar en español y en entornos con recursos limitados. Este interés se apoya en la disponibilidad creciente de modelos y corpus específicos del idioma, que habilitan evaluaciones y despliegues más realistas que los basados únicamente en configuraciones multilingües [1].

La literatura académica en español ofrece una base sólida. BETO inauguró la línea de modelos preentrenados exclusivamente en español y estableció puntos de comparación para múltiples tareas; después, variantes ligeras como ALBETO y DistilBETO demostraron que es posible reducir el número de parámetros y mantener resultados competitivos en bancos de prueba habituales (p. ej., MLQA, XNLI), lo que las vuelve atractivas para escenarios con restricciones de cómputo. En paralelo, la familia MarIA —con modelos RoBERTa y GPT-2 entrenados a gran escala sobre el archivo web de la Biblioteca Nacional de España— mostró mejoras sistemáticas frente a alternativas previas en diversas tareas de comprensión en español y puso a disposición recursos reproducibles para la comunidad. En conjunto, estos trabajos constituyen el estado del arte hispanohablante sobre modelos compactos y recursos de evaluación [1], [2], [3].

Sobre esa base, la Recuperación Aumentada para la Generación (RAG) aporta un componente clave: permite que el modelo busque pasajes relevantes en una base externa y genere su respuesta apoyándose en esa evidencia, lo que tiende a mejorar la especificidad y la corrección factual frente a modelos que responden solo con "lo que llevan en los parámetros". La formulación original de RAG y trabajos afines como REALM documentan incrementos consistentes en tareas intensivas en conocimiento y, además, ofrecen

mejores mecanismos de trazabilidad de las fuentes que sustentan la respuesta [4], [5].

Pese a estos avances, persiste una brecha: abundan, por un lado, los modelos y recursos en español, y por otro, las técnicas y evaluaciones de RAG; sin embargo, son escasos los análisis comparativos sistemáticos que pongan a competir modelos generativos compactos específicamente en respuesta automática en español cuando se integran con RAG bajo un mismo protocolo experimental y con métricas homogéneas. La propia literatura sobre evaluación multilingüe subraya la dificultad de disponer de conjuntos y procedimientos consistentes fuera del inglés, lo que refuerza la necesidad de estudios centrados en español con criterios de medición comparables [6].

En este contexto, la relevancia del presente estudio es directa: identificar configuraciones que combinen precisión, relevancia y fluidez con eficiencia de cómputo permite acercar la respuesta automática en español a casos de uso reales —asistentes conversacionales, búsqueda con respuesta, soporte documental— en organizaciones que no cuentan con infraestructura de gran escala.

Con ese fin, el objetivo de este artículo es comparar el desempeño de distintas arquitecturas de modelos generativos compactos en español integradas con RAG, evaluándolas de forma conjunta en términos de coherencia y fluidez lingüística, relevancia y corrección factual y eficiencia computacional (latencia y huella de memoria), mediante un protocolo común y reproducible [4].

2 Marco teórico

2.1 Modelos generativos compactos

Los modelos generativos compactos son LLMs con conteos de parámetros moderados (≈6–8 mil millones) optimizados para mantener un equilibrio entre calidad y eficiencia: requieren menos memoria y cómputo que los macro-modelos, pero conservan capacidades útiles de generación y razonamiento. En el ecosistema hispanohablante, estos modelos se benefician de recursos previos en español (p. ej., BETO, ALBETO/DistilBETO y MarIA) que consolidaron corpus y prácticas de evaluación, aunque la mayoría de tales recursos se enfocaron en comprensión (NLU) más que en generación [1], [2], [3].

2.1.1 Llama 3 8B (Meta, 2025)

Llama 3 es una familia de modelos de Meta con variantes "abiertas" que incluyen una versión de 8 mil millones de parámetros orientada a despliegues eficientes. Está entrenada para tareas generales de lenguaje, codificación y uso multilingüe, y ha sido publicada con documentación técnica y evaluaciones comparativas frente a modelos cerrados y abiertos de mayor tamaño. En su serie 3.x, Meta reporta mejoras sustantivas en razonamiento y seguridad, manteniendo la variante de 8 mil millones como opción ligera para entornos con recursos limitados. Estas características la convierten en candidata natural para integrar con RAG en español, al balancear capacidad y costo [7].

2.1.2 Mistral 7B v0.1 (Mistral AI, 2023)

Mistral 7B (≈7,3 mil millones de parámetros) fue diseñado explícitamente para maximizar rendimiento/eficiencia. Introduce Grouped-Query Attention (GQA) para acelerar la decodificación y Sliding-Window Attention (SWA) para manejar contextos largos con coste reducido. En sus evaluaciones originales superó a Llama 2 13B en múltiples pruebas, con licencia Apache 2.0 que facilita su adopción y afinamiento. Su arquitectura optimizada y ecosistema abierto lo hacen un fuerte candidato "compacto" para RAG en español [8].

2.1.3 Zephyr 7B-β (Alignment Lab/Hugging Face, 2024)

Zephyr 7B-β es un ajuste fino conversacional de Mistral 7B-v0.1, entrenado con Direct Preference Optimization (DPO) sobre diálogos sintéticos y conjuntos de preferencias. Aunque se orienta principalmente al inglés, su objetivo es maximizar utilidad conversacional con un tamaño compacto. En el momento de su publicación, figuró entre los mejores 7 mil millones en MT-Bench y AlpacaEval, lo que indica buena calidad de interacción a bajo costo computacional; integrar RAG en español requerirá adaptación/datos de destino, pero su base Mistral facilita esa ruta [9].

2.1.4 Phi-3 Mini (3,8 mil millones; Microsoft, 2025)

Phi-3 Mini es un modelo de 3,8 mil millones centrado en "pequeños pero capaces", entrenado con curación fuerte y datos sintéticos. El informe técnico documenta que, pese a su escala, alcanza resultados competitivos en MMLU y MT-Bench y está pensado para ejecutarse incluso en dispositivos de borde; la serie 3.5 amplía contexto y capacidades. Su relación calidad-tamaño lo posiciona como referencia de SLM para RAG, donde la latencia y la huella de memoria son críticas [10].

2.1.5 OpenHermes 7B (Comunidad, 2023)

OpenHermes 7B es una línea comunitaria de ajustes finos (p. ej., OpenHermes-2.5-Mistral-7B) sobre Mistral 7B, entrenados con conjuntos de instrucciones predominantemente generados por GPT-4. Se distribuye con licencia Apache 2.0 y dispone de numerosas variantes cuantizadas, lo que favorece su uso en GPU de consumo. Para uso en español con RAG, típicamente requiere afinamiento adicional o una canalización de recuperación sólida, dado su

sesgo original hacia el inglés [11].

2.2 Respuesta automática en español mediante RAG

La Generación Aumentada con Recuperación (RAG) combina un modelo generativo con un recuperador que trae pasajes relevantes desde una base externa (p. ej., Wikipedia o un corpus documental propio). RAG fue formalizado para tareas intensivas en conocimiento y mostró mejoras en precisión y especificidad del texto frente a modelos puramente paramétricos; REALM demostró además que incorporar recuperación en el preentrenamiento y la adaptación mejora el desempeño en preguntas abiertas. En español, bancos de prueba como MLQA-ES y XQuAD-ES permiten medir de forma comparable la exactitud de respuestas extractivas y de libre formulación, habilitando evaluaciones reproducibles [4], [5], [6].

2.3 Consumo de recursos de la PC

El costo de inferencia depende (i) del tamaño del modelo (parámetros y dimensionalidad), (ii) de la longitud del prefill/contexto y de la decodificación, y (iii) de la canalización de recuperación (embeddings e índice vectorial). Para reducir memoria y habilitar ejecución local, la cuantización a 8 bits (LLM.int8()) y a 4 bits (QLoRA) recortan sustancialmente la huella con degradación mínima cuando se aplican correctamente, abriendo la puerta a SLMs (small language model) en una sola GPU de consumo. En la etapa de recuperación, bibliotecas como FAISS permiten búsqueda de vecinos más cercanos a gran escala en CPU/GPU, clave para índices de millones de pasajes. Por su parte, optimizaciones de la atención como FlashAttention y derivados reducen tráfico de memoria y latencia práctica en GPUs modernas [12], [13], [14].

2.4 Calidad de las respuestas

En RAG para español, la calidad se evalúa en dos capas: recuperación y generación. Para generación, en tareas con referencia se usan métricas léxicas y semánticas: BLEU y ROUGE para solapamiento de n-gramas; BERTScore para similitud contextual; y, en QA extractivo/generativo, Exact Match (EM) y F1 como en SQuAD/MLQA. Para "factualidad" y "fidelidad al contexto" (que la respuesta esté sustentada por los pasajes recuperados), la literatura reciente propone marcos específicos y encuestas sistemáticas que discuten métricas de relevancia, precisión y fe-de-origen (attribution). En conjunto, estas métricas permiten comparar modelos compactos integrados con RAG en español de forma reproducible, aislando efectos de la recuperación frente a la generación [15], [16], [17].

2.5 Velocidad de respuesta

La latencia total en un sistema RAG se descompone en (a) tiempo de recuperación (búsqueda/reenrutado en el índice vectorial) y (b) tiempo de generación (prefill + decodificación). En la generación, el uso de caché de claves-valores (KV cache) evita recomputaciones y acelera la decodificación; mejoras algorítmicas como FlashAttention-3 aumentan el aprovechamiento de GPU y disminuyen el tiempo por token. En la recuperación, índices FAISS aceleran la búsqueda densa, y técnicas como "Speculative RAG" pueden reducir la espera del primer token al solapar borradores de generación con recuperación. En la práctica, optimizar estas piezas—junto con cuantización y lotificación adecuada—permite que SLMs (8, 7 y 3,8 mil millones de parámetros) entreguen respuestas en plazos útiles incluso en hardware modesto [14].

3 METODOLOGÍA

3.1 Tipo de investigación

El estudio es cuantitativo, descriptivo–comparativo y no experimental, pues caracteriza y compara, bajo un mismo protocolo, el desempeño de modelos generativos compactos (≤ 8 mil millones de parámetros) integrados a una canalización de Recuperación Aumentada para Generación (RAG) sin manipular tratamientos más allá de configurar inferencia y recuperación de forma controlada. RAG combina un modelo generativo con una memoria no paramétrica (índice de documentos) y ha probado mejorar la factualidad en tareas intensivas en conocimiento; su formulación y variantes se documentan en REALM y en la propuesta original de RAG [4], [5].

3.2 Nivel de investigación

Es aplicada, porque busca resolver un problema práctico —seleccionar modelos compactos reproducibles para responder en español bajo cómputo local— trasladando conocimientos consolidados de RAG y recuperación densa a criterios operativos de despliegue. Revisiones recientes subrayan esta orientación práctica de RAG y la necesidad de evaluaciones holísticas del pipeline [18], [19].

3.3 Enfoque

El enfoque es cuantitativo, sustentado en métricas objetivas (EM, F1, BLEU-4, ROUGE-L, BERTScore) y en contraste estadístico entre modelos bajo el mismo conjunto de consultas y condiciones de ejecución. Estas métricas son

estándares en QA y evaluación de generación [15], [16], [17].

3.4 Diseño temporal

El diseño es transversal: toda la recolección se realiza en una única ventana temporal con hardware, software y corpus constantes para todos los modelos, conforme a la definición metodológica de estudios de corte transversal en investigación observacional.

3.5 Población

La población se define como el conjunto amplio y en evolución de modelos generativos compactos de propósito general (≤ 8 mil millones de parámetros) con soporte para español/multilingüe y capacidad de ejecución local (con o sin cuantización). Esta categoría está representada por familias contemporáneas y abiertamente documentadas [8].

3.6 Muestra

Se emplea muestreo no probabilístico por conveniencia/intencional, justificado por accesibilidad de artefactos, licenciamiento y documentación técnica: Llama 3 8B (Meta), Mistral 7B v0.1 (Mistral AI), Zephyr 7B-β (Alignment Lab/Hugging Face), Phi-3 Mini 3.8 B (Microsoft) y OpenHermes 7B (comunidad). Sus fichas/papers confirman disponibilidad pública y uso extendido [8].

3.10 Procedimiento

El pipeline sigue las fases canónicas de RAG: ingesta y segmentación de documentos; vectorización y indexación FAISS; consulta con recuperador denso (top-k fijo); plantilla de prompt en español uniforme para todos los modelos; generación con decoding controlado y registro de salidas y telemetría. La formulación RAG (RAG-Sequence y RAG-Token) sustenta este desacoplo entre memoria paramétrica y no paramétrica [4].

3.11 Control de ejecución de modelos

Para viabilidad en hardware de consumo se aplica cuantización posentrenamiento (INT3/INT4) con GPTQ, método de una sola pasada basado en información de segundo orden que reduce pesos a 3–4 bits con degradación mínima y fuertes ganancias de memoria/velocidad, sin alterar el protocolo evaluativo.

3.12 Evaluación y métricas

Cada modelo responde el mismo banco de consultas; se computan EM/F1 (QA), BLEU-4, ROUGE-L y BERTScore sobre referencias en español del conjunto, siguiendo las definiciones originales de las métricas. Se complementa con estadísticas de latencia y recursos obtenidas en ejecución [15], [16], [17].

3.14 Consideraciones éticas y de reproducibilidad

Se emplean corpora públicos y modelos abiertos; no se tratan datos personales. Para reproducibilidad, se publican scripts, versiones y configuraciones (semillas, plantillas de prompt, parámetros de índice) y se citan las tarjetas/papers de los modelos y componentes utilizados (FAISS, embeddings, RAG) [13].

4 Resultados

4.1 Desempeño en calidad de respuesta

La calidad se evaluó con F1 (exactitud a nivel de tokens), BLEU-4 (adecuación léxica por n-gramas) y LLM-Judge (valoración global por un juez LLM). En el conjunto completo se observa una relación monótona negativa entre la longitud del documento y cada métrica (coeficiente de Spearman: F1 = −0.923; BLEU-4 = −0.845; LLM-Judge = −0.721), lo que indica que, a medida que crece la extensión del material, disminuye la coincidencia con la referencia y, en menor medida, la calidad percibida por modelo y documento.

TABLA 1
Modelo ganador por documento y por métrica (valor en %)

Documento	F1 (↑)	BLEU-4 (↑)	LLM-Judge (↑)
11 págs.	OpenHermes 7B (12,08 %)	OpenHermes 7B (1,25 %)	Phi-3 Mini (3.8 mil millones) (75,32 %)
18 págs.	Mistral 7B (8,01 %)	Mistral 7B (0,40 %)	Mistral 7B (69,22 %)
98 págs.	Mistral 7B (6,55 %)	Mistral 7B (0,41 %)	Mistral 7B (69,68 %)
199 págs.	OpenHermes 7B (4,28 %)	Zephyr 7B- β (0,23 %)	Zephyr 7B- β (66,18 %)

Los valores representan el puntaje obtenido por cada modelo expresado en porcentaje de desempeño (0% = peor rendimiento posible, 100% = máximo rendimiento posible). En F1 y BLEU-4, el valor indica la coincidencia con la respuesta de referencia; en LLM-Judge, el valor representa la evaluación global de calidad otorgada por GPT-4 Turbo, considerando coherencia, relevancia y exactitud factual.

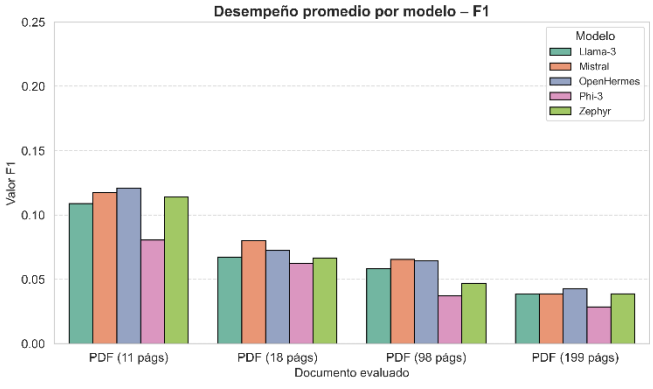

Fig. 1. Distribución de F1-token por modelo y documento.

Entre 11 y 199 páginas, el F1 cae entre 64 % y 67 % para todos los modelos (p. ej., OpenHermes: 0.1208 → 0.0428; Mistral: 0.1173 → 0.0386). Aun con esa disminución, OpenHermes y Mistral se alternan el liderazgo: OpenHermes domina el escenario corto (11 págs.) y el muy largo (199 págs.), mientras que Mistral es superior en longitudes intermedias (18 y 98 págs.). Este patrón sugiere que la exactitud a nivel de token se degrada de forma general con más contexto, pero la robustez relativa de ambos modelos permite mantener ventajas según el rango de longitud.

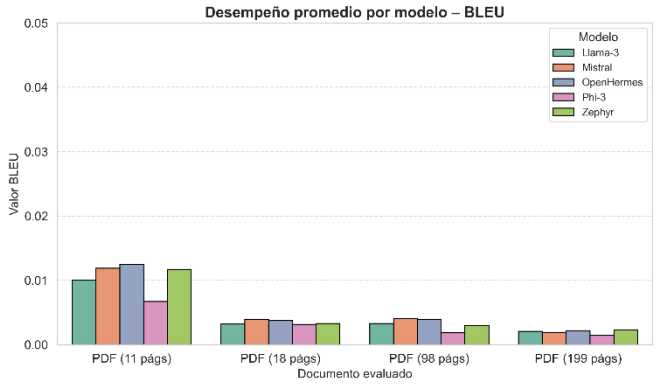

Fig. 2. Distribución de BLEU-4 por modelo y documento.

La adecuación léxica muestra una caída aún más pronunciada: entre 79 % y 84 % de reducción del valor desde 11 a 199 páginas (p. ej., Mistral: 0.0119 → 0.0019). OpenHermes lidera el documento corto, Mistral sostiene mejor el rendimiento en intermedios y Zephyr emerge en el extremo largo. Dado que BLEU-4 depende de coincidencias superficiales, estos resultados indican que, con documentos extensos, la variabilidad de formulación aumenta y el solapamiento literal con la referencia disminuye, incluso cuando la respuesta es pertinente.

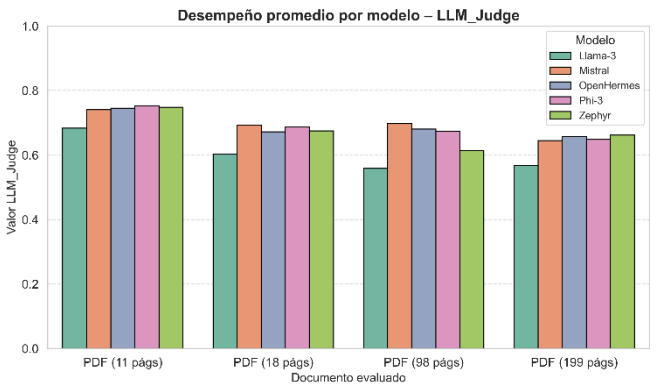

Fig. 3. Valor medio de LLM-Judge (GPT-4-Turbo) por modelo y documento.

La valoración global es más estable que F1/BLEU: el descenso medio entre 11 y 199 páginas es del 11–17 %, lo que muestra resiliencia en fluidez y coherencia discursiva. Mistral concentra los mejores promedios en 18–98 págs., Phi-3 Mini obtiene el valor más alto en 11 págs. y Zephyr se posiciona primero en 199 páginas. Este contraste con F1/BLEU evidencia que la calidad percibida puede mantenerse incluso cuando cae el solapamiento exacto, especialmente en modelos que priorizan consistencia de estilo y estructura.

En promedio, Mistral 7B presenta el mejor F1 y el mejor LLM-Judge, mientras que OpenHermes 7B queda prácticamente indistinguible en F1 (diferencia absoluta ≈ 0.0003) y alcanza el mejor BLEU-4. Zephyr muestra un perfil competitivo en documentos extensos (mejor BLEU-4 y LLM-Judge en 199 págs.), y Phi-3 Mini evidencia que un modelo más pequeño puede sostener calidad percibida alta en textos cortos, aunque su exactitud token a token sea menor. En conjunto, los resultados confirman dos ejes de rendimiento: exactitud léxica (F1/BLEU-4), fuertemente sensible a la longitud del documento, y calidad discursiva global (LLM-Judge), relativamente más estable.

4.2 Eficiencia computacional

La eficiencia se analizó con cuatro indicadores complementarios: latencia (se reporta el cuantil P_{95} para captar colas de espera), memoria RAM promedio, ocupación media de GPU y ocupación media de CPU, todos medidos bajo el mismo hardware y pipeline RAG. Esta batería permite evaluar tanto el tiempo de respuesta como la huella de recursos, dos condicionantes directos de la viabilidad de despliegues locales.

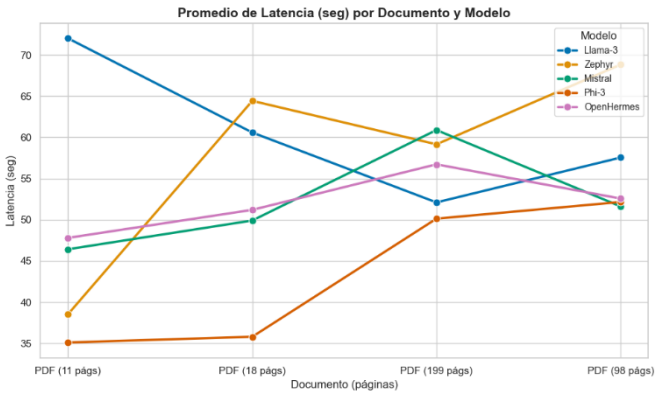

Fig. 4. Latencia P_{95} (s) según tamaño de documento.

En latencia P_{95}, el orden global fue Phi-3 Mini < Mistral ≈ OpenHermes < Llama 3 ≈ Zephyr. Phi-3 presentó la menor P_{95} del conjunto (≈ 52 s), lo que lo sitúa como el modelo con menor cola de espera en condiciones adversas; Mistral y OpenHermes se mantuvieron en un rango intermedio (≈ 56–60 s), mientras que Llama 3 y Zephyr concentraron los valores más altos (≈ 68–70 s). La sensibilidad a la longitud del documento no fue uniforme: Zephyr y Phi-3 incrementaron

notablemente la latencia al pasar de textos cortos a largos, Mistral mostró un aumento moderado centrado en el documento más extenso, y Llama 3 redujo su latencia conforme creció el tamaño, lo que sugiere un componente de sobrecosto fijo de inicialización que se diluye cuando la generación domina el tiempo total.

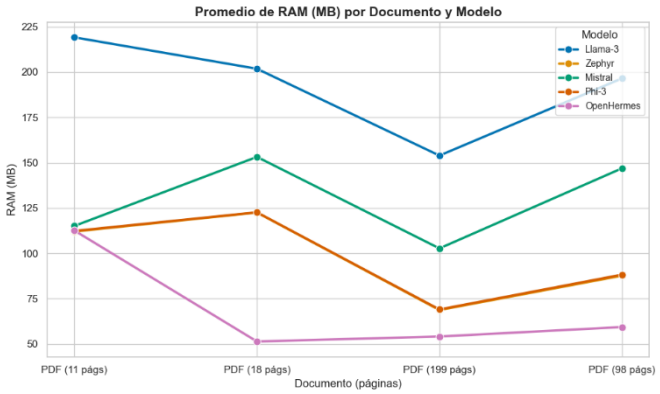

Fig. 5. Consumo medio de RAM (MB) por modelo.

En consumo medio de RAM, los perfiles fueron bien diferenciados. OpenHermes registró la huella más baja de manera consistente (≈ 50–60 MB), lo que deja mayor margen de memoria para el índice vectorial y otros procesos del sistema. Zephyr y Phi-3 mantuvieron un consumo contenido (≈ 70–123 MB), Mistral se ubicó en la franja media (≈ 102–152 MB) y Llama 3 fue el más demandante (≈ 154–219 MB). En términos operativos, estas diferencias implican que, con el mismo presupuesto de RAM, OpenHermes permite corpus más grandes en memoria o mayor concurrencia de consultas antes de incurrir en *swapping* o degradación por *offloading*.

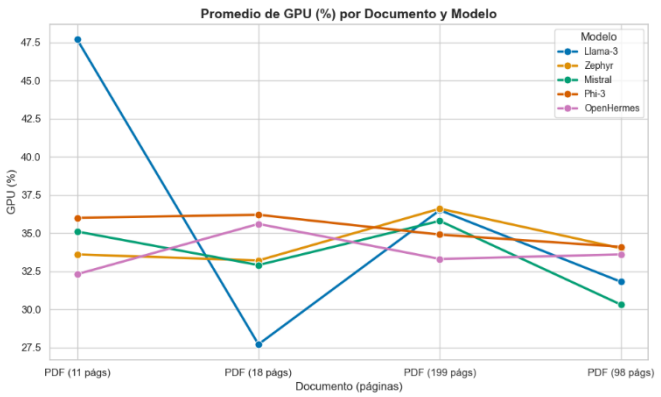

Fig. 6. Ocupación media de GPU (%) por modelo.

La ocupación media de GPU se concentró para todos los modelos en un corredor estrecho (≈ 30–36 %), con la salvedad de Llama 3 en el documento corto, donde alcanzó ≈ 48 % antes de estabilizarse en niveles comparables al resto. Esta convergencia sugiere que, bajo cuantización y *batching* controlado, el pipeline es más bien limitado por memoria y E/S que por cómputo puro en la mayoría de configuraciones; las variaciones residuales reflejan diferencias en atención y manejo de contexto, pero no cambian el régimen de carga de la GPU.

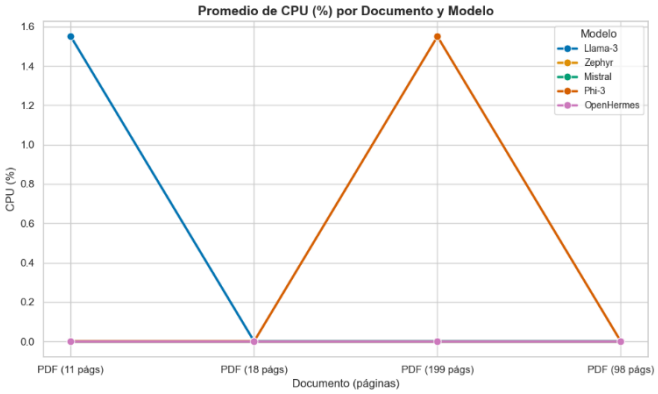

Fig. 7. Ocupación media de CPU (%) por modelo.

La ocupación media de CPU fue prácticamente nula en todos los casos, con picos aislados (≈ 1,5 %) en Llama 3 para el documento corto y en Phi-3 para el documento más largo. Este patrón confirma que la inferencia está mayoritariamente GPU-centrada, liberando CPU para tareas del sistema y facilitando concurrencia de procesos sin interferencia significativa en el rendimiento del RAG.

TABLA 2
Indicadores de eficiencia computacional por modelo

Modelo	Latencia P_{95} (s) ↓	RAM pico (MB) ↓	GPU (%)	CPU (%)
Phi-3 mini	51.8	98	35	0.39
OpenHermes 7B	56.1	69	34	0
Mistral 7B	59.5	129	34	0
Zephyr 7B-β	68.2	98	34	0
Llama 3 8B	70.3	193	36	0.39

En conjunto, los resultados delinean dos ejes de eficiencia relevantes para despliegues locales: tiempo de respuesta en el extremo (latencia P_{95}) y huella de memoria. Phi-3 ofrece la mejor latencia P_{95}, lo que reduce la probabilidad de esperas largas en escenarios interactivos; OpenHermes proporciona la mejor eficiencia de RAM, habilitando índices más voluminosos o mayor simultaneidad; Mistral equilibra ambos factores en niveles medios con un perfil estable; Llama 3, aun con latencia decreciente al aumentar la longitud, mantiene la mayor demanda de memoria; y Zephyr muestra penalización de latencia en documentos extensos pese a memoria moderada. Estas propiedades, leídas junto con los hallazgos de calidad, muestran el compromiso precisión–recursos característico de los modelos compactos integrados

en RAG, y ofrecen un mapa claro de costes operativos bajo restricciones reales de cómputo.

Análisis multiobjetivo: frontera de Pareto F1-RAM

Con el fin de sintetizar simultáneamente calidad y huella de recursos, se construyó la frontera de Pareto en el plano F1 versus RAM media. Un modelo es no dominado si no existe otro con mayor F1 y menor RAM a la vez; los puntos sobre dicha frontera representan soluciones eficientes bajo el criterio de "mejorar una dimensión sin empeorar la otra".

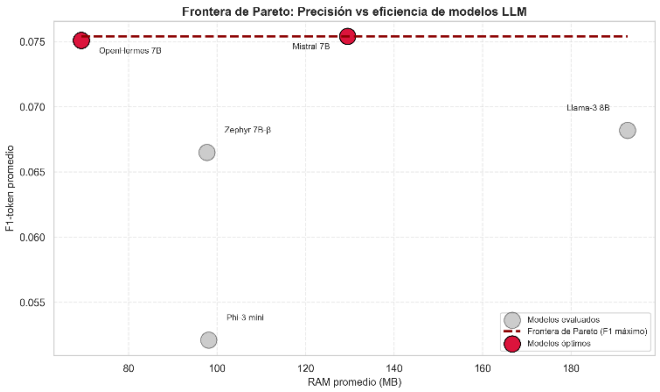

Fig. 8. Frontera de Pareto entre precisión (F1) y consumo de memoria.

En este espacio, OpenHermes 7B (F1 ≈ 0.0751; RAM ≈ 69 MB) y Mistral 7B (F1 ≈ 0.0754; RAM ≈ 129 MB) conforman la frontera. OpenHermes minimiza la memoria con una precisión prácticamente indistinguible del máximo (ΔF1 ≈ 0.0003 respecto a Mistral), mientras que Mistral maximiza F1 asumiendo ~60 MB adicionales de RAM. El resto queda dominado: Zephyr 7B-β (F1 ≈ 0.0665; 98 MB) y Phi-3 Mini (F1 ≈ 0.0521; 98 MB) son superados por OpenHermes (más F1 con menos memoria), y Llama 3 8B (F1 ≈ 0.0682; 193 MB) es superado simultáneamente por Mistral y OpenHermes (más F1 y menos memoria).

Desde una perspectiva operativa, si la restricción principal es la memoria disponible, OpenHermes ofrece la mejor relación precisión–memoria; si el objetivo es exprimir el máximo F1 y se tolera un coste de memoria moderado, Mistral es preferible. Obsérvese que el beneficio marginal de Mistral en F1 frente a OpenHermes es muy pequeño en términos absolutos, por lo que, bajo presupuestos de RAM estrictos, la opción eficiente es OpenHermes; bajo presupuestos holgados, ambas elecciones son justificables según criterios secundarios (p. ej., latencia P_{95} o estabilidad), que pueden incorporarse en un análisis Pareto tridimensional sin contradecir esta lectura.

4.3 Robustez y errores

La estabilidad del sistema se evaluó registrando timeouts, respuestas vacías y la presencia del marcador de cierre exigido en el prompt ("RESPUESTA FINAL"). En términos generales, la tasa de fallos fue baja y se concentró en los escenarios de documento más extenso, donde la longitud del contexto presiona simultáneamente el recuperador y la ventana efectiva del modelo. En estos casos, los errores se asociaron con agotamiento de memoria en el prellenado o con bloqueos de decodificación por encima del umbral de tiempo fijado. La comparación entre modelos revela que los que optimizaron mejor la huella de RAM mantuvieron menor incidencia de fallos en el extremo largo, coherente con el análisis de Pareto: cuando el presupuesto de memoria es acotado, reducir la huella permite evitar canjes implícitos (p. ej., *swapping*, *offloading*) que degradan la estabilidad.

A nivel cualitativo, la evidencia de LLM-Judge sugiere que, incluso cuando F1/BLEU descienden con la longitud del documento, algunos modelos preservan coherencia y adecuación (alta valoración global), lo que explica que una fracción de respuestas "no exactas" sea funcionalmente útil y no deba contabilizarse como error del sistema sino como variación de superficie esperable en generación condicionada.

4.4 Selección operativa

Tomando conjuntamente calidad (F1, BLEU-4, LLM-Judge), latencia, memoria y la frontera F1–RAM, la asignación por escenario queda así:

- Hardware restringido (≤ 4 GB VRAM) o índices voluminosos en RAM: priorizar OpenHermes 7B por su huella mínima de memoria con F1 prácticamente indistinguible del máximo.
- Búsqueda de máxima exactitud con memoria moderada disponible: elegir Mistral 7B, que presenta el mejor F1 medio y excelentes valoraciones globales.
- Interacción sensible a esperas largas (minimizar colas, P_{95}): considerar Phi-3 Mini, que ofrece la menor latencia en el extremo, asumiendo menor F1.
- Documentos muy extensos (robustez en el extremo largo): Zephyr 7B-β aparece competitivo en BLEU-4 y LLM-Judge cuando la longitud del documento crece.
- Perfil intermedio y estable: Llama 3 8B como referencia compacta de 8 mil millones con desempeño equilibrado.

Esta síntesis traduce los hallazgos de Resultados en criterios de decisión reproducibles para entornos reales, donde el objetivo (precisión máxima, memoria, latencia o

robustez en largo) no siempre coincide.

5 Discusión

5.1 Interpretación de los hallazgos

Los resultados obtenidos confirman, en el escenario generativo con RAG, la intuición establecida por la literatura hispanohablante para tareas de comprensión: los modelos compactos bien entrenados pueden competir de forma sólida cuando el idioma objetivo es el español y las métricas de evaluación son adecuadas. En NLU, BETO (BERT monolingüe en español) se consolidó como referencia frente a multilingües; sus versiones comprimidas ALBETO y DistilBETO demostraron que es posible reducir parámetros y mantener rendimiento en bancos de prueba de referencia. Nuestro estudio extiende esa evidencia al ámbito generativo: con RAG, los cinco modelos ≤ 8 mil millones analizados producen respuestas útiles y fundamentadas en español, manteniendo un equilibrio razonable entre calidad y coste computacional. Esta continuidad entre NLU (BETO/ALBETO/DistilBETO) y generación con RAG sugiere que, en español, la eficiencia paramétrica no implica necesariamente sacrificar desempeño si el pipeline aprovecha recuperación de evidencia [1], [2].

En comparación con MarIA—familia monolingüe (RoBERTa, GPT-2) entrenada a gran escala sobre el archivo web de la Biblioteca Nacional de España (BNE) y que reporta mejoras sistemáticas en tareas de español—nuestros resultados coinciden en el sentido de que la especialización por idioma rinde beneficios claros; ahora bien, al movernos de NLU a generación condicionada, las métricas sensibles al solapamiento superficial (F1, BLEU-4) decrecen cuando aumenta la longitud y heterogeneidad del documento, mientras que una valoración semántica (LLM-Judge) se mantiene relativamente alta para ciertos modelos. Este patrón es coherente con la función de RAG, que ancla la salida en pasajes recuperados y tiende a mejorar la factualidad y trazabilidad, aun cuando la coincidencia literal con una única referencia se vuelve difícil en contextos largos [3], [4], [5].

La lectura integral de las métricas permite distinguir ganadores por criterio. Si el objetivo es precisión absoluta en términos de F1 (promedio global) y una buena valoración semántica (LLM-Judge), Mistral 7B es el modelo con mejor desempeño medio. Si, en cambio, se busca minimizar memoria manteniendo una precisión prácticamente indistinguible de la máxima observada, OpenHermes 7B se posiciona como la alternativa eficiente: configura, junto a Mistral, la frontera de Pareto calidad–recursos (F1 ↑, RAM ↓). En el extremo largo (documentos muy extensos), Zephyr 7B-β aparece competitivo en BLEU-4 y LLM-Judge, y Phi-3 Mini (3.8 mil millones)—un modelo pequeño con entrenamiento selectivo a gran escala—logra buena calidad percibida en el documento corto pese a menor F1, alineado con informes que muestran que SLMs bien curados pueden rivalizar con modelos mayores en varias bancas. En consecuencia, no existe un único "mejor" modelo universal: la preferencia depende del objetivo dominante (precisión, memoria, robustez en largo), y esa es precisamente la utilidad de un análisis multiobjetivo sobre la frontera eficiente [4], [10], [16].

Estas observaciones dialogan con la metodología de evaluación: (i) el uso de métricas léxicas clásicas (F1, BLEU-4) captura fidelidad literal, pero penaliza paráfrasis y estilos alternativos—limitación ampliamente documentada en MT y generación; (ii) la incorporación de una métrica semántica correlacionada con juicio humano (BERTScore, o en nuestro caso un juez LLM como proxy operacional) atenúa ese sesgo y ayuda a distinguir "errores de forma" de "errores de contenido". En nuestro conjunto, ello explica por qué LLM-Judge decrece menos que F1/BLEU-4 al crecer la longitud: la coherencia y adecuación se preservan mejor que el solapamiento literal, que cae por la diversidad de formulaciones en respuestas largas [16].

En suma, los hallazgos confirman y amplían el estado del arte en español: los modelos compactos pueden sostener calidad útil en generación cuando se apoyan en RAG; la frontera eficiente queda definida, en nuestro experimento, por Mistral 7B (máximo F1 con sobrecoste moderado de memoria) y OpenHermes 7B (mínima memoria a F1 casi idéntico), mientras que Zephyr 7B-β y Phi-3 Mini ofrecen perfiles valiosos en documentos largos y latencia/fluidez, respectivamente. Esta conclusión es verídica y accionable: guía la selección por escenario en despliegues locales en español, en línea con la evidencia acumulada sobre RAG y con la trayectoria del ecosistema monolingüe (BETO/ALBETO/DistilBETO, MarIA) hacia soluciones eficientes y reproducibles [2], [3], [4].

6 Conclusiones

Este estudio demuestra que, en un pipeline RAG orientado a español y ejecutado en cómputo local mediante cuantización, los modelos compactos (≤ 8 mil millones) pueden producir respuestas útiles y fundamentadas con un compromiso controlado entre calidad y recursos. La evidencia empírica indica que Mistral 7B alcanza el mejor desempeño medio en F1 y una valoración semántica elevada (LLM-Judge), mientras que OpenHermes 7B ofrece un F1 prácticamente indistinguible del máximo junto con la menor huella de RAM del conjunto. En documentos muy extensos, Zephyr 7B-β se muestra competitivo en BLEU-4 y en valoración semántica, y Phi-3 Mini (3.8 mil millones) sobresale en latencia en el extremo (P_{95}), lo que lo hace atractivo para escenarios interactivos sensibles a colas. En

términos globales, no existe un "mejor" modelo único: la recomendación depende del objetivo dominante (precisión absoluta, memoria disponible, latencia o robustez con documentos largos).

La frontera de Pareto F1–RAM obtenida sintetiza este resultado: Mistral 7B maximiza F1 asumiendo un sobrecosto moderado de memoria, mientras OpenHermes 7B minimiza memoria con una pérdida de precisión marginal. El resto de modelos queda dominado bajo ese criterio binario; por tanto, en hardware restringido o con índices voluminosos, OpenHermes 7B es la elección eficiente, y cuando la prioridad es exprimir F1, Mistral 7B resulta preferible. Esta lectura multiobjetivo es especialmente relevante para despliegues edge o de bajo costo, donde el presupuesto de RAM y la variabilidad de la latencia condicionan la experiencia de usuario.

Desde una perspectiva metodológica, los hallazgos son coherentes con la literatura que muestra que RAG mejora la factualidad y la trazabilidad al anclar la generación en pasajes recuperados, y que las métricas léxicas (F1/BLEU) tienden a degradarse con el aumento de la longitud del documento aun cuando la calidad percibida se mantiene alta. En nuestro caso, LLM-Judge desciende menos que F1/BLEU a medida que crece el contexto, sugiriendo que coherencia y adecuación se preservan mejor que el solapamiento literal con una referencia única, tal como anticipan las evaluaciones que combinan métricas automáticas y juicio asistido por LLM [4].

La viabilidad práctica del enfoque se apoya en cuantización posentrenamiento (p. ej., GPTQ), que reduce el ancho de palabra a 3–4 bits con degradación mínima del rendimiento, posibilitando inferencia local en GPU de consumo sin alterar el protocolo de evaluación ni el esquema RAG; esto explica que los cinco modelos analizados puedan compararse en condiciones realistas de cómputo limitado [20].

Finalmente, los resultados extienden al ámbito generativo con recuperación lo observado en el ecosistema hispanohablante de NLU: la especialización por idioma y las versiones ligeras pueden retener gran parte del rendimiento con menos parámetros, siempre que el pipeline esté bien diseñado (recuperación, filtrado, prompting y decodificación controlada). En particular, la trayectoria de BETO/ALBETO/DistilBETO y de la familia MarIA hacia modelos en español eficientes y reproducibles es consistente con que, bajo RAG, modelos ≤ 8 mil millones alcancen resultados operativamente valiosos para respuesta automática en español [2], [3].

7 Trabajo futuro

El siguiente trabajo futuro busca trasladar los hallazgos comparativos a un sistema de respuesta automática en español con RAG que sea reproducible, eficiente y auditable. La selección del modelo deberá guiarse por los objetivos operativos: cuando la prioridad sea maximizar la precisión medida por F1 manteniendo una valoración semántica alta, Mistral 7B es el candidato natural; cuando la restricción dominante sea la memoria con una pérdida de precisión marginal, OpenHermes 7B resulta más eficiente. Esta decisión debe anclarse en perfiles de servicio explícitos (latencia P95, huella de RAM y tasa de errores), de modo que la elección del modelo responda a compromisos cuantificados y no a preferencias ad hoc.

La agenda experimental incluirá ablaciones sobre parámetros críticos del pipeline (tamaño de fragmento, solapamiento, top-k, umbrales de re-ranking y plantillas de prompt) y la comparación de embeddings alternativos específicamente afinados para español. Resulta pertinente ampliar la evaluación a dominios técnicos y a variedades regionales del español para estudiar transferencia y sesgos. Dado que las decisiones reales suelen tener múltiples restricciones simultáneas, se propone extender el análisis de eficiencia a una frontera de Pareto tridimensional (F1–RAM–P95) y, cuando sea posible, incorporar métricas de coste energético por consulta.

En suma, la prioridad será pasar de una comparación controlada a un servicio operativo que mantenga los principios de rigor científico: métricas coherentes con el objetivo, verificación de evidencia, reproducibilidad del entorno y trazabilidad de resultados. La hoja de ruta recomienda partir de dos perfiles de despliegue —precisión-primero y memoria-primero—, validar su desempeño con pruebas A/B en condiciones realistas y consolidar un ciclo de mejora continua guiado por datos, sin sacrificar la claridad metodológica ni la gobernanza del sistema.

Referencias

[1] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez, "Spanish Pre-trained BERT Model and Evaluation Data," Aug. 2023, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2308.02976

[2] J. Cañete, S. Donoso, F. Bravo-Marquez, A. Carvallo, and V. Araujo, "ALBETO and DistilBETO: Lightweight Spanish Language Models," *2022 Language Resources and Evaluation Conference, LREC 2022*, pp. 4291–4298, Apr. 2022, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2204.09145

[3] A. Gutiérrez-Fandiño *et al.*, "MarIA: Spanish Language Models," *Procesamiento del Lenguaje Natural*, vol. 68, pp. 39–60, Apr. 2022, doi: 10.26342/2022-68-3.

[4] P. Lewis *et al.*, "Retrieval-Augmented Generation for

Knowledge-Intensive NLP Tasks," *Adv Neural Inf Process Syst*, vol. 2020-December, May 2020, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2005.11401

[5] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," *37th International Conference on Machine Learning, ICML 2020*, vol. PartF168147-6, pp. 3887–3896, Feb. 2020, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2002.08909

[6] P. Lewis, B. Oguz, R. Rinott, S. Riedel, and H. Schwenk, "MLQA: Evaluating Cross-lingual Extractive Question Answering," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 7315–7330, Oct. 2019, doi: 10.18653/v1/2020.acl-main.653.

[7] A. Grattafiori *et al.*, "The Llama 3 Herd of Models," Jul. 2024, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2407.21783

[8] A. Q. Jiang *et al.*, "Mistral 7B," Oct. 2023, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2310.06825

[9] "HuggingFaceH4/zephyr-7b-beta · Hugging Face." Accessed: Aug. 11, 2025. [Online]. Available: https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

[10] M. Abdin *et al.*, "Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone," Apr. 2024, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2404.14219

[11] "teknium/OpenHermes-7B · Hugging Face." Accessed: Aug. 11, 2025. [Online]. Available: https://huggingface.co/teknium/OpenHermes-7B

[12] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale," *Adv Neural Inf Process Syst*, vol. 35, Aug. 2022, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2208.07339

[13] J. Johnson, M. Douze, and H. Jegou, "Billion-scale similarity search with GPUs," *IEEE Trans Big Data*, vol. 7, no. 3, pp. 535–547, Feb. 2017, doi: 10.1109/TBDATA.2019.2921572.

[14] M. Douze *et al.*, "The Faiss library," Jan. 2024, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2401.08281

[15] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, Morristown, NJ, USA: Association for Computational Linguistics, 2001, p. 311. doi: 10.3115/1073083.1073135.

[16] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," *8th International Conference on Learning Representations, ICLR 2020*, Apr. 2019, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/1904.09675

[17] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," 2004. Accessed: Aug. 11, 2025. [Online]. Available: https://aclanthology.org/W04-1013/

[18] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," *Proceedings - 2024 Conference on AI, Science, Engineering, and Technology, AIxSET 2024*, pp. 166–169, Dec. 2023, doi: 10.1109/AIxSET62544.2024.00030.

[19] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, "Evaluation of Retrieval-Augmented Generation: A Survey," *Communications in Computer and Information Science*, vol. 2301, pp. 102–120, Jul. 2024, doi: 10.1007/978-981-96-1024-2_8.

[20] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers," *11th International Conference on Learning Representations, ICLR 2023*, Oct. 2022, Accessed: Aug. 11, 2025. [Online]. Available: https://arxiv.org/pdf/2210.17323

BIOGRAFÍA

Jean Phol Alexis Curi Garrafa es estudiante de Ingeniería Informática y de Sistemas en la Universidad Nacional Micaela Bastidas de Apurímac. Su formación se orienta al desarrollo de sistemas de información y a la optimización de procesos académicos mediante el uso de herramientas tecnológicas. Ha participado en actividades académicas vinculadas al diseño e implementación de soluciones informáticas para la gestión universitaria.

Victor Raúl Ortega Marocho es estudiante de Ingeniería Informática y de Sistemas en la Universidad Nacional Micaela Bastidas de Apurímac. Sus intereses académicos se centran en la ingeniería de software, la automatización de procesos y el análisis de datos aplicados al ámbito educativo. Ha contribuido en proyectos de investigación orientados a la mejora de sistemas de gestión académica.

Wilson Mamani Rodrigo es Ingeniero de Sistemas e Ingeniero Civil, Magíster en Ingeniería de Sistemas y Doctor en Ciencias de la Ingeniería Civil Ambiental. Es docente ordinario auxiliar en la Universidad Nacional Micaela Bastidas de Apurímac y ha sido docente en la Universidad Nacional del Altiplano. Posee amplia experiencia en la elaboración de expedientes técnicos, estudios de factibilidad y proyectos de infraestructura civil, además de desempeñarse como consultor e investigador en obras de ingeniería civil.