UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERÍA

Desarrollo de un prototipo de chatbot inteligente basado en arquitectura de generación aumentada por recuperación (RAG), para la automatización de consultas frecuentes en la mesa de ayuda de una empresa de servicios

Tesis para optar el Título de
Ingeniero Industrial y de Sistemas

Vladimir Franklin Manchay Garcia

Asesor:
Mgtr. Ing. Gerson Ommar La Rosa Lama

Piura, enero de 2026

UNIVERSIDAD DE PIURA

Declaración Jurada de Originalidad del Trabajo Final

Yo, Vladimir Franklin Manchay Garcia, egresado del Programa Académico de Ingeniería Industrial y de Sistemas de la Facultad de Ingeniería de la Universidad de Piura, identificado(a) con DNI: 75688569, declaro que:

Soy autor del trabajo final titulado:
"Desarrollo de un prototipo de chatbot inteligente basado en arquitectura de generación aumentada por recuperación (RAG), para la automatización de consultas frecuentes en la mesa de ayuda de una empresa de servicios".
El mismo que presento bajo la modalidad de Tesis para optar el Título profesional de Ingeniero Industrial y de Sistemas.

Que el trabajo se realizó en coautoría con los siguientes alumnos de la Universidad de Piura.

- Haga clic o pulse aquí para escribir texto, identificado con Elija un elemento: Escribir número
- Haga clic o pulse aquí para escribir texto, identificado con Elija un elemento: Escribir número

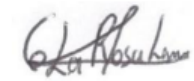
El texto de mi trabajo final es original y no vulnera los derechos de terceros o, de ser el caso, derechos de los coautores, incluidos los derechos de propiedad intelectual, datos personales, entre otros. En tal sentido, el texto de mi trabajo final no ha sido plagiado total ni parcialmente, para lo cual, he respetado las normas internacionales de citas y referencias de las fuentes consultadas. Asimismo, el texto del trabajo final que presento no ha sido publicado ni presentado antes en cualquier medio electrónico o físico; y que la investigación, los resultados, datos, conclusiones y demás información presentada que atribuyo a mi autoría son veraces.
En caso de detectarse el incumplimiento de lo declarado asumo frente a terceros, la Universidad de Piura y/o la Administración Pública toda responsabilidad que pueda derivarse por el trabajo final presentado. Lo señalado incluye responsabilidad pecuniaria incluido el pago de multas u otros por los daños y perjuicios que se ocasionen.

La asesoría del trabajo estuvo a cargo de los siguientes docentes de la Universidad de Piura:

- Mgtr. Ing. Gerson Ommar La Rosa Lama, identificado con DNI: 44727188

Declaro (declaramos) que:
Luego de haber empleado el software de coincidencia Turnitin, revisado las fuentes de información señaladas por el autor, y en razón de mi (nuestra) experiencia como investigador(es), declaro (declaramos) que las ideas expuestas en el trabajo final alcanzan las condiciones de calidad, integridad y originalidad acorde a los objetivos institucionales y estándares en materia de investigación. Finalmente, no asumo (asumimos) responsabilidad por la posible vulneración de derechos de autor en el trabajo final referido, pues tal responsabilidad es exclusiva del autor.

Fecha: 26/01/2026.



.....
Firma del asesor

.....
Firma del autor

.....
Firma del co-asesor

Dedicatoria

A Dios por concederme salud y vida; a mis padres y hermanos, por su apoyo incondicional en cada etapa; a Oriana por su valiosa ayuda y compañía; y a todas las personas que estuvieron presentes en este largo camino.

Resumen

La presente investigación aborda la ineficiencia operativa en mesas de ayuda corporativas causada por la saturación de consultas repetitivas, proponiendo como solución un prototipo de chatbot basado en la arquitectura Generación Aumentada por Recuperación (RAG). El objetivo principal fue implementar un sistema funcional capaz de operar en un entorno 100% local y de código abierto, garantizando la soberanía total de los datos y la nulidad de costos operativos variables.

La metodología siguió un enfoque experimental utilizando LlamaIndex para la orquestación, ChromaDB para el almacenamiento vectorial y el modelo Gemma3:4b para la generación. Para la validación, se aplicó un protocolo de evaluación cuantitativa con el framework Ragas, durante el cual se identificó y corrigió un sesgo metodológico crítico en los *prompts* de evaluación para el idioma español, un hallazgo clave para la validez de estudios futuros en la región.

Los resultados estadísticos revelan una dicotomía en el desempeño: el subsistema de recuperación demostró una robustez sobresaliente (*context_recall*: 0.890), mientras que el componente generativo presentó una variabilidad estocástica inherente a modelos pequeños (media de relevancia: 0.635). No obstante, la mediana de relevancia operativa de 0.699 valida que el sistema es capaz de resolver satisfactoriamente la mayoría de las interacciones típicas. Se concluye que es técnicamente viable desplegar soluciones de soporte automatizado en hardware restringido sin depender de APIs comerciales, ofreciendo una tasa de éxito operativo cercana al 70% con total privacidad.

Tabla de contenido

Lista de tablas

Lista de figuras

Introducción

En el contexto actual de la transformación digital, las mesas de ayuda de Tecnologías de la Información (TI) se han consolidado como el primer punto de contacto crítico para garantizar la continuidad operativa de las organizaciones. Sin embargo, estos centros de soporte enfrentan un desafío sistémico: la saturación por un alto volumen de consultas repetitivas y de baja complejidad (Nivel 1), tales como restablecimiento de accesos o configuraciones básicas. Este fenómeno no solo incrementa los costos operativos y desgasta al capital humano, sino que degrada la experiencia del usuario final debido a tiempos de espera prolongados.

Históricamente, la automatización de este soporte se limitaba a chatbots basados en reglas rígidas con escasa capacidad de comprensión. La irrupción de la Inteligencia Artificial Generativa y los Modelos de Lenguaje de Gran Escala (LLM) prometió superar estas barreras, ofreciendo asistentes capaces de razonar y conversar naturalmente. No obstante, la adopción masiva de estas tecnologías en entornos corporativos sensibles se ha visto frenada por dos obstáculos fundamentales: el costo operativo variable asociado a las APIs comerciales y, de manera crítica, el riesgo de exponer información confidencial o propiedad intelectual a proveedores externos en la nube.

Ante esta disyuntiva, surge la necesidad de explorar arquitecturas que equilibren la capacidad cognitiva de la IA moderna con los requisitos estrictos de privacidad y eficiencia de recursos. La presente investigación responde a esta necesidad mediante el diseño, implementación y evaluación de un prototipo basado en la arquitectura de Generación Aumentada por Recuperación (RAG). A diferencia de los modelos genéricos, el enfoque RAG ancla la generación de texto a una base de conocimiento corporativa verificable, reduciendo las alucinaciones y aumentando la precisión técnica.

La innovación central de este estudio reside en su implementación 100% local y de Código Abierto. Se prescinde de servicios en la nube para construir un ecosistema tecnológico soberano, integrando herramientas como LlamaIndex y ChromaDB con modelos cuantizados (Gemma3:4b) ejecutables en hardware de consumo estándar. Este enfoque busca validar si es posible alcanzar un desempeño operativo aceptable sin incurrir en costos por token ni comprometer la seguridad de los datos.

El documento se estructura para guiar al lector a través de este proceso experimental. El Capítulo 1 contextualiza la problemática y justifica la solución propuesta. El Capítulo 2 establece el marco teórico sobre el procesamiento de lenguaje natural y la arquitectura RAG. El Capítulo 3 detalla la metodología y el diseño del sistema. El Capítulo 4 presenta los resultados empíricos y un análisis estadístico riguroso del desempeño. Finalmente, el Capítulo 5 ofrece conclusiones y recomendaciones estratégicas para la adopción de esta tecnología en el sector servicios.

Capítulo 1
Marco contextual

Este capítulo aborda el problema de la ineficiencia en las mesas de ayuda de tecnologías de la información, originado por la alta demanda de consultas repetitivas. Se presentan los antecedentes, la justificación y la situación actual que evidencian la necesidad de soluciones más avanzadas basadas en inteligencia artificial, como la arquitectura RAG.

1.1 Planteamiento del problema

Las mesas de ayuda de Tecnologías de la Información (TI) constituyen una pieza esencial para garantizar la continuidad operativa de cualquier organización, al ofrecer el primer nivel de soporte frente a fallas o incidentes técnicos. Sin embargo, su rendimiento suele verse comprometido por una saturación derivada de consultas simples y reiterativas. Este patrón constituye un serio cuello de botella que afecta directamente en la eficiencia y costos operativos (Sanugommula, 2024).

En las organizaciones, la tecnología es indispensable, y cualquier falla, por mínima que sea, impacta negativamente la productividad. Un estudio reciente sobre los principales desafíos que enfrenta una mesa de ayuda revela que una gran cantidad de tickets se relacionan con problemas de conectividad, inicios de sesión, acceso a cuentas y actualizaciones de software (Marcel & Aotearoa, 2025).

Las consecuencias de mantener procesos manuales en la gestión de solicitudes se agravan con el tiempo. La dependencia continua de intervención humana para resolver consultas incrementa el costo por ticket y reduce la capacidad de respuesta. Las empresas que no toman medidas para resolver este problema se ven obligadas a mantener una estructura de costos más alta para satisfacer la demanda de consultas, lo que afecta su rentabilidad. El costo de mantener personal dedicado exclusivamente al soporte técnico se vuelve insostenible cuando el volumen de solicitudes diarias es alto (Salinas Santiago et al., 2024).

La experiencia del usuario también se ve afectada. La ineficiencia en la atención de solicitudes provoca tiempos de espera prolongados, lo que impacta negativamente en la satisfacción. En un entorno donde se demandan respuestas rápidas, la ineficiencia operativa genera frustración, ya que no se puede ofrecer la inmediatez que se espera. Esto deteriora la percepción del servicio y, a largo plazo, la lealtad de los clientes hacia la empresa se ve reducida (Rick et al., 2024).

En industrias de servicios como telecomunicaciones, banca y salud, donde las solicitudes de atención son variadas y frecuentes, la falta de soluciones eficientes aumenta significativamente la carga de trabajo. Según el Instituto Nacional de Estadística e Informática (INEI, 2023) el sector servicios de la capital de Perú representó más del 50% del PBI en 2023; sin embargo, las empresas continúan dependiendo de procesos manuales lentos que no

logran satisfacer la creciente demanda de consultas. Esta situación genera cuellos de botella en los procesos de atención, retrasa la resolución de problemas y eleva los costos operativos.

En conclusión, la ineficiencia de las mesas de ayuda, que dependen de procesos manuales para atender consultas repetitivas, genera un impacto directo en los tiempos de respuesta y en los costos operativos de las empresas. Este problema no solo afecta la eficiencia del equipo de soporte, sino que también deteriora la experiencia del usuario y la competitividad empresarial. Si no se implementan soluciones tecnológicas que optimicen la gestión de consultas y automaticen las tareas de bajo valor, las empresas continuarán enfrentando un modelo de soporte lento, costoso e insostenible.

1.2 Justificación

La Experiencia Digital del Empleado (DEX) se ha consolidado como un factor clave para la retención del talento y la productividad organizacional. Un soporte de TI lento o ineficaz genera fricción y frustración en el entorno laboral. Casi dos de cada tres colaboradores afirman que los problemas tecnológicos recurrentes afectan negativamente su estado de ánimo, compromiso y satisfacción laboral. La demora en resolver fallos simples, como errores de software o accesos restringidos, interrumpe la concentración y debilita la motivación. Ante esta problemática, resulta necesario implementar soluciones tecnológicas que reduzcan el tiempo de respuesta y garanticen asistencia inmediata y precisa. (Ivanti, 2025).

En ese contexto, la presente investigación propone un cambio frente a las ineficiencias operativas mediante el diseño, desarrollo y evaluación de un prototipo de chatbot inteligente basado en la arquitectura RAG. Esta tecnología representa un avance significativo en la inteligencia artificial conversacional, al mitigar las limitaciones de los Modelos Grandes de Lenguaje (LLM) y permitir respuestas más precisas y verificables que favorecen a la mejora de la productividad (James et al., 2025).

No obstante, aunque los LLM como el Transformador Generativo Preentrenado *4* (GPT-4) han demostrado un notable potencial generativo, su uso en entornos empresariales críticos presenta limitaciones significativas. Estos modelos funcionan como “cajas negras” de conocimiento paramétrico, lo que los hace propensos a la obsolescencia del conocimiento y a la generación de información incorrecta (alucinaciones). Al carecer de actualización continua, pueden ofrecer respuestas plausibles pero erróneas, lo cual resulta inaceptable en contextos como una mesa de ayuda de TI, donde la precisión es esencial para evitar fallos o riesgos de seguridad (Gill et al., 2025).

Frente a ello, la arquitectura RAG se presenta como una solución sólida y prometedora. Al separar la base de conocimiento del modelo, permite que las respuestas se fundamenten en información externa, controlada y actualizable. Esta investigación valida empíricamente dicha hipótesis mediante un prototipo que demuestra cómo RAG reduce las alucinaciones y

mejora la fidelidad factual de las respuestas, anclándolas a la documentación técnica de la empresa. Además, se analiza cómo la capacidad de citar fuentes recuperadas fortalece la precisión, transparencia y confianza en entornos corporativos (Chakraborty et al., 2025).

A nivel académico, la investigación aborda una brecha identificada en la literatura reciente (2024–2025): la limitada aplicación de la arquitectura RAG a datos empresariales caracterizados por datos complejos y heterogéneos. Los marcos tradicionales se han centrado en corpus textuales homogéneos, como Wikipedia o noticias, que no reflejan la diversidad semántica y estructural de la información corporativa. La aplicación directa de enfoques RAG convencionales a estos contextos tiende a disminuir el rendimiento, debido a la dificultad de preservar las relaciones y el contexto entre los distintos tipos de información, lo que puede generar respuestas parciales o imprecisas (Cheerla, 2025).

Desde la perspectiva organizacional, el chatbot basado en RAG también contribuye al desarrollo del capital humano. Estudios recientes, como el de IBM (2024) señalan que el 87% de los ejecutivos creen que la IA generativa potenciará las habilidades de los empleados en lugar de reemplazarlos. En este contexto, el chatbot basado en RAG automatiza tareas repetitivas, liberando al personal para abordar problemas más complejos. Esto fomenta el *upskilling* y transforma el rol del agente de soporte, convirtiéndolo en un puesto más estratégico, enfocado en el análisis y la optimización de la experiencia tecnológica.

Adicionalmente, el chatbot actúa como un nivelador organizacional, al garantizar soporte continuo las 24 horas y eliminar barreras de ubicación, horario e idioma gracias a las capacidades multilingües de los LLM. Su carácter impersonal también favorece a quienes evitan consultar a un agente humano por temor a hacer preguntas simples. De esta manera, el sistema promueve la equidad y la inclusión digital, asegurando que todos los colaboradores reciban el mismo nivel de apoyo. Así, su impacto trasciende la satisfacción individual y contribuye al fortalecimiento de la cohesión y la igualdad de oportunidades en una fuerza laboral diversa y distribuida (Mahajan, 2025).

Finalmente, la implementación del chatbot basado en RAG ofrece beneficios económicos significativos. Está diseñado para automatizar las consultas de alto volumen y baja complejidad que dominan la carga de trabajo de Nivel 1. De acuerdo con estimaciones de la industria, una parte significativa de los tickets de soporte puede resolverse completamente mediante automatización, con un costo marginal casi nulo por interacción. Para organizaciones que gestionan grandes volúmenes de solicitudes, este enfoque se traduce en ahorros operativos sustanciales y en la posibilidad de reorientar recursos hacia iniciativas estratégicas de crecimiento e innovación (MetricNet, 2021).

En síntesis, la presente investigación ofrece tres aportes principales al campo. En primer lugar, presenta un caso de estudio detallado sobre la implementación de RAG en bases de conocimiento internas, un tema de alta relevancia práctica, pero con poca evidencia aplicada. En segundo lugar, el prototipo desarrollado permitirá evaluar empíricamente

estrategias de recuperación y razonamiento sobre un corpus empresarial heterogéneo y con lenguaje técnico especializado, distinto de los conjuntos de datos académicos tradicionales. Finalmente, los resultados proponen un marco metodológico para la adopción efectiva de RAG en entornos corporativos, demostrando su viabilidad y ofreciendo lineamientos para su implementación adaptativa en contextos complejos.

1.3 Situación actual

La transformación digital ha pasado de ser una tendencia emergente a consolidarse como una prioridad estratégica en las empresas de servicios. En el contexto peruano, esta evolución es evidente, ya que las organizaciones han situado la experiencia del cliente en el centro de sus estrategias de valor y crecimiento. En este escenario, la gestión eficiente de las Tecnologías de la Información se convierte en un pilar fundamental para garantizar la continuidad operativa y fortalecer la competitividad empresarial. Las mesas de ayuda, antes percibidas como simples centros de costos, ahora son reconocidas como áreas que impactan directamente en la satisfacción del usuario y en la eficiencia corporativa (Ernst & Young, 2024).

Las mesas de ayuda en empresas de servicios enfrentan importantes desafíos operativos que limitan su eficiencia. La saturación de los canales de atención, causada por un alto volumen de consultas repetitivas, genera retrasos y dificulta la gestión de incidentes. Según Sosa Erazo (2024), la ausencia de procesos claros clasificar solicitudes puede incrementar los tiempos de respuesta hasta en un 40%, afectando la priorización de tareas críticas y la calidad del servicio. Esta acumulación de ineficiencias reduce la capacidad operativa del equipo de soporte y deteriora la experiencia del cliente, poniendo en riesgo su satisfacción y lealtad.

Las deficiencias operativas de las mesas de ayuda tradicionales impactan directamente en la confianza del cliente y en la competitividad empresarial. Aunque la mayoría de los consumidores (88%) prefiere la interacción humana en el servicio al cliente, una atención ineficiente y lenta deteriora la lealtad y afecta la reputación corporativa. Optimizar los puntos de contacto con el cliente se vuelve, por tanto, una necesidad estratégica más que operativa. En este contexto, la automatización se presenta como la principal alternativa para transformar el soporte técnico en un verdadero generador de valor (IPSOS, 2024).

La IA conversacional, mediante herramientas como los chatbots, ofrece una solución estratégica para automatizar la atención al cliente de primer nivel y optimizar la gestión de requerimientos. Su adopción responde directamente a la principal motivación de las empresas peruanas en la necesidad de reducir costos y simplificar procesos para impulsar la transformación digital, un factor prioritario para el 42% de las organizaciones. Al automatizar las respuestas a consultas frecuentes, estas tecnologías liberan al personal técnico para que pueda concentrarse en incidentes de mayor complejidad (Ernst & Young, 2024).

La evolución de los asistentes conversacionales ha mostrado un notable progreso en los últimos años. Los primeros chatbots, basados en sistemas de reglas predefinidas, ofrecían soluciones simples y económicas, pero carecían de la capacidad para gestionar interacciones complejas o no estructuradas. Si bien eran eficaces en tareas repetitivas, su rigidez impedía una adaptación real al lenguaje natural humano. En contraste, los asistentes modernos, impulsados por LLM e IA generativa, ofrecen interacciones más adaptativas, personalizadas y contextualmente coherentes, superando así las limitaciones de los enfoques tradicionales basados en respuestas estáticas (Garzón-Quiroz et al., 2025).

1.4 Antecedentes

Espinosa-Luna et al. (2023) desarrollaron un chatbot basado en GPT-3.5-Turbo para automatizar la atención a preguntas frecuentes de estudiantes universitarios, buscando reducir la saturación de los canales de soporte. Mediante un estudio preexperimental y pruebas de usabilidad, se observó que el 93% de los usuarios consideró útiles las respuestas, demostrando eficiencia y facilidad de uso. Este trabajo respalda el uso de modelos de lenguaje grande y RAG para automatizar consultas frecuentes en entornos de soporte TI.

Salinas Santiago et al. (2024) presentan la implementación de un chatbot inteligente para mejorar la eficiencia y calidad del soporte técnico en la mesa de ayuda de una empresa. El sistema, desarrollado con GPT y *frameworks* web como Nest.js, Next.js y PostgreSQL, permite ofrecer respuestas rápidas y precisas a los usuarios, aumentando la satisfacción y reduciendo costos operativos. Este caso evidencia el éxito de los chatbots en soporte TI, respaldando la aplicación de RAG en el presente proyecto.

Bhat et al. (2024) en un estudio en la industria de los restaurantes demuestra la efectividad del enfoque RAG en chatbots de servicio. Este sistema, que interactúa en lenguaje natural, alcanzó un puntaje Bilingual Evaluation Understudy (BLEU) de 60% en comprensión del lenguaje. La integración de RAG con modelos LLM mejora la precisión frente a modelos tradicionales, validando que RAG optimiza la atención al cliente mediante el uso de conocimientos externos.

Ordóñez-Camacho et al. (2024) abordan la necesidad de un servicio de mesa de ayuda virtual en la universidad mediante un chatbot basado en un LLM diseñado para ofrecer respuestas precisas y contextualizadas. La investigación demuestra que técnicas como Fine-Tuning y RAG son fundamentales para casos de uso específicos, lo que coincide con la aplicación de la arquitectura RAG en el desarrollo del presente trabajo.

Lee et al. (2024) desarrollaron un chatbot basado en arquitectura RAG para mejorar la precisión y eficiencia en soporte técnico, dado que los chatbots generales no proporcionaban respuestas suficientemente exactas. Integrando un LLM con un sistema de recuperación de conocimiento y evaluando mediante métricas de *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE) y pruebas de usuario, lograron mejoras de 38% a 188% en precisión y un

aumento en la satisfacción de los usuarios. Este estudio respalda la utilización de RAG para generar respuestas contextuales precisas en mesas de ayuda.

Purewal Martinez & Sobero Rodriguez (2025) abordaron la lentitud y falta de personalización en la atención al cliente mediante un chatbot potenciado con GPT-3.5-turbo, bases de datos vectoriales y RAG, logrando respuestas precisas, fluidas y con alta precisión (0,98). El estudio respalda el uso de RAG como estrategia efectiva para automatizar consultas frecuentes en sistemas de soporte TI.

Swacha & Gracel (2025) realizan una revisión sobre la aplicación de chatbots basados en RAG en educación, destacando que los LLM por sí solos generan información incompleta o incorrecta sin acceso a datos actualizados. El estudio muestra que RAG permite integrar documentos externos, reducir errores de información y aumentar la confiabilidad de los agentes conversacionales, reforzando su uso para garantizar respuestas precisas y actualizadas en sistemas de soporte TI.

Capítulo 2
Marco teórico

Este capítulo aborda los principales conceptos relacionados con la Inteligencia Artificial Generativa (GenAI) y los LLM. Se revisan los fundamentos del Procesamiento de Lenguaje Natural (NLP), la arquitectura RAG como modelo híbrido de recuperación y generación, las bases de datos vectoriales utilizadas para la búsqueda por similitud, y los frameworks de orquestación que permiten integrar y controlar aplicaciones construidas con LLM.

2.1 Fundamentos del procesamiento de lenguaje natural

El NLP constituye una disciplina de la IA, cuyo objetivo principal es capacitar a los sistemas computacionales para comprender, interpretar, generar y manipular el lenguaje humano de una manera que sea tanto significativa como útil. Este campo se sitúa en la intersección de la informática, la lingüística y la IA, y se dedica a modelar los complejos mecanismos a través de los cuales los seres humanos comparten información, con el fin de cerrar la brecha existente entre la comunicación humana y la comprensión de las máquinas (Hasan & Ibrahim, 2025).

El alcance del NLP en la actualidad es muy amplio y su aplicación se ha vuelto indispensable en numerosos campos. Una de sus funciones más relevantes es la extracción de información y conocimiento a partir de datos no estructurados, especialmente en formato de texto libre, que representan la mayor parte de la información digital disponible. Por ejemplo, en el ámbito de la salud, se estima que más del 80% de los datos clínicos en los Registros de Salud Electrónicos (EHR) es de naturaleza no estructurada, incluyendo notas clínicas, informes de alta, narrativas de pacientes e informes de imagenología. Históricamente, este vasto repositorio de conocimiento ha sido un recurso poco aprovechado, ya que su análisis manual es un proceso extremadamente lento, costoso y propenso a errores, lo que limita su utilidad a gran escala (Golder et al., 2025).

El valor del NLP reside en su capacidad para automatizar y escalar la conversión de grandes volúmenes de texto no estructurado en datos analizables, superando la inviabilidad del análisis manual ante los actuales corpus masivos. Estas técnicas permiten extraer información clínica relevante, como historiales de tabaquismo, toxicidades o fechas de síntomas, directamente desde notas en texto libre. Así, el NLP se consolida como una necesidad práctica y económica, al ofrecer una solución escalable para aprovechar el valor oculto del texto no estructurado mediante su integración en modelos de aprendizaje automático (Hasan & Ibrahim, 2025).

El NLP ha evolucionado desde los sistemas basados en reglas lingüísticas, pasando por el aprendizaje automático estadístico, hasta el aprendizaje profundo, que predomina en la actualidad. Los primeros enfoques, fundamentados en reglas definidas por expertos, ofrecían alta interpretabilidad, pero eran difíciles de escalar. Luego, el aprendizaje automático

introdujo modelos como como las Máquinas de Vectores de Soporte (SVM) y regresión logística, capaces de aprender patrones a partir de datos etiquetados. Finalmente, el aprendizaje profundo, mediante arquitecturas como Redes Neuronales Convolucionales (CNN), Redes Neuronales Recurrentes (RNN) y *Transformers*, revolucionó el campo al automatizar la extracción de características y aprender representaciones contextuales complejas directamente de los datos (Hasan & Ibrahim, 2025).

La evolución del NLP no ha sido una sustitución lineal de tecnologías, sino un proceso de integración de distintos enfoques según el equilibrio entre rendimiento, interpretabilidad y disponibilidad de recursos. Los métodos basados en reglas mantienen su relevancia por su transparencia y facilidad de auditoría, especialmente en contextos con limitaciones computacionales o donde la trazabilidad de las decisiones es indispensable. Esta coexistencia se debe a los problemas de interpretabilidad de muchos modelos de aprendizaje profundo, cuya complejidad hace difícil entender sus predicciones, lo cual es crítico en áreas sensibles como la toma de decisiones. En respuesta, el NLP ha adoptado un enfoque pragmático que combina modelos como los *Transformers* para tareas generales con módulos basados en reglas para extraer información precisa y auditada (S. Zhang et al., 2023).

Antes de la aparición de la arquitectura *Transformer*, el NLP se sustentaba en las RNN y sus variantes, como la de Memoria a Corto y Largo Plazo (LSTM). Estas arquitecturas fueron diseñadas para procesar secuencias de datos de manera iterativa, manteniendo una "memoria" del contexto previo mediante conexiones cíclicas. Sin embargo, las RNN tradicionales enfrentaban el problema de la desaparición del gradiente, lo que limitaba su capacidad para aprender dependencias a largo plazo. Las arquitecturas LSTM surgieron como una solución parcial, introduciendo mecanismos de compuertas que regulaban el flujo de información y mejoraban el aprendizaje de secuencias extensas, consolidándose como la arquitectura dominante en múltiples tareas de NLP (Mienye et al., 2024).

Vaswani et al. (2017) revolucionaron el campo del NLP, introduciendo la arquitectura *Transformer*. Este modelo rompió con los enfoques tradicionales basados en recurrencia y convoluciones, reemplazándolos por el mecanismo de autoatención (*self-attention*), capaz de capturar dependencias globales entre palabras sin importar su posición en la secuencia. Al eliminar el procesamiento secuencial de las RNN, el *Transformer* permitió una paralelización masiva del entrenamiento, reduciendo significativamente los tiempos y costos computacionales, y estableció un nuevo estándar en tareas como la traducción automática. Su estructura, basada en pilas de codificadores y decodificadores con capas de autoatención y redes *feed-forward*, sentó las bases de los modelos de lenguaje modernos.

Self-attention es el componente central del *Transformer* y el responsable de su capacidad para capturar relaciones contextuales complejas dentro de una secuencia. Permite que cada palabra (o token) evalúe la relevancia de todas las demás, incluyéndose a sí misma, para construir una representación contextualizada. Cada token se transforma en tres vectores:

Query (consulta), *Key* (clave) y *Value* (valor), obtenidos mediante multiplicaciones con matrices de pesos entrenables. El modelo compara cada *Query* con todas las *Keys* para calcular puntuaciones de similitud, que se normalizan con una función *softmax* y se interpretan como pesos de atención. Finalmente, cada palabra obtiene su nueva representación como una combinación ponderada de los *Values* de toda la secuencia. Este mecanismo elimina la necesidad del procesamiento secuencial de las RNN y otorga al *Transformer* una notable capacidad de paralelización y eficiencia (Kowsher et al., 2025).

Self-attention ha demostrado ser muy eficaz, pero plantea un desafío computacional significativo, ya que requiere comparar cada token con todos los demás, lo que genera una complejidad computacional y de memoria cuadrática con respecto a la longitud de la secuencia. Esta limitación se convierte en un cuello de botella al procesar secuencias largas, reduciendo la capacidad del modelo para manejar grandes contextos de manera eficiente. Como respuesta, han surgido variantes con atención subcuadrática o lineal, como *Linformer* y *Longformer*, que utilizan aproximaciones de bajo rango o patrones de atención dispersos para reducir la carga computacional. Recientemente, los Modelos de Espacio de Estados (SSMs) han emergido como alternativas con complejidad lineal, manteniendo la capacidad de modelar dependencias a largo plazo (Fichtl et al., 2025).

Los modelos de lenguaje preentrenados basados en *Transformer* se dividen en dos enfoques principales, codificadores y decodificadores, orientados a la comprensión y la generación del lenguaje, respectivamente. Las Representaciones de Codificadores Bidireccionales a partir de Transformadores (BERT), se especializan en aprender representaciones contextuales profundas del texto mediante autoatención bidireccional, lo que permite que cada token se relacione con todos los demás en ambas direcciones. BERT se preentrena con el objetivo de Modelado de Lenguaje Enmascarado (MLM), en el que se enmascaran aleatoriamente algunos tokens y el modelo debe predecirlos a partir del contexto. Este enfoque optimiza la comprensión del lenguaje, lo que hace que BERT sobresalga en tareas como clasificación de texto, detección de entidades y respuesta a preguntas e inferencia semántica (S. Zhang et al., 2023).

Las arquitecturas basadas solo en decodificadores, como GPT, se centran en la generación de texto mediante un enfoque autorregresivo o causal, en el que cada token solo puede atender a los que lo preceden, debido a un mecanismo de atención unidireccional con enmascaramiento. Se entrenan con el objetivo de Modelado de Lenguaje Causal (CLM), aprendiendo a predecir el siguiente token a partir del contexto anterior, lo que les permite generar texto coherente de manera secuencial. Este enfoque es la base de los LLM, sobresaliendo en tareas como redacción creativa, traducción automática, resumen de textos y chatbots conversacionales. La diferencia clave entre BERT y GPT radica en sus objetivos, mientras que BERT favorece la comprensión al generar representaciones contextuales completas, GPT se enfoca en la predicción secuencial, simbolizando la distinción técnica entre leer y escribir en los sistemas de IA (Abdulahi Jimale Said & Abdihakin Mohamud Ismail, 2025).

La tokenización es el primer paso en el procesamiento del lenguaje natural, ya que segmenta el texto en unidades mínimas llamadas tokens, que conforman el vocabulario del modelo y determinan la eficiencia y el rendimiento de las representaciones posteriores. La evolución de las técnicas de tokenización ha pasado de enfoques basados en palabras o caracteres hacia métodos de subpalabras, que equilibran el tamaño del vocabulario y la capacidad de generalización. La tokenización por palabras tiene la desventaja de crear vocabularios grandes y dificultades con palabras desconocidas, mientras que la de caracteres genera secuencias demasiado largas. En cambio, la tokenización por subpalabras divide las palabras en unidades significativas como morfemas (Mostafa et al., 2025).

Entre los métodos más destacados de tokenización de subpalabras se encuentran la Codificación por Pares de Bytes (BPE), que fusiona pares de tokens frecuentes; WordPiece, que selecciona fusiones maximizando la probabilidad del corpus; y *Unigram Language Model,* que adopta un enfoque probabilístico para optimizar la verosimilitud del texto. Este enfoque de subpalabras permitió el surgimiento de modelos multilingües más eficientes, especialmente en lenguajes morfológicamente complejos, al ofrecer una representación más universal y consistente que mejora el aprendizaje por transferencia y el desempeño en idiomas con recursos limitados, convirtiéndose así en una piedra angular del NLP moderno (Chintha & Konduru, 2025).

Tras la tokenización, los tokens se convierten en representaciones numéricas a través de *embeddings*, vectores densos que capturan relaciones semánticas y sintácticas, situando palabras con significados similares cerca en un espacio continuo. Los primeros enfoques, como Word2Vec y GloVe, producían *embeddings* estáticos, asignando a cada palabra un único vector sin considerar el contexto, lo que limitaba su capacidad para manejar la polisemia y los matices lingüísticos. Los modelos *Transformer* (BERT, GPT) introdujeron *embeddings* contextualizados, adaptando la representación de cada palabra según su contexto y mejorando la capacidad para capturar fenómenos complejos como la ironía y la ambigüedad, y acercando su procesamiento al de la cognición humana (Alkaabi et al., 2025).

2.2 Inteligencia artificial generativa y LLM

La IA ha experimentado una transformación significativa en los últimos años, impulsada por el surgimiento de la GenAI. Este avance ha ampliado su alcance, llevando sus capacidades más allá de las tareas analíticas y predictivas hacia la creación de contenido novedoso. A diferencia de los modelos predictivos, que se enfocan en analizar datos para clasificar o extrapolar patrones, la GenAI destaca por su capacidad para producir textos, imágenes, audio y código que emulan la creatividad humana mediante instrucciones detalladas conocidas como *prompts*. Este proceso no solo interpreta datos, sino que los utiliza como base para generar artefactos nuevos y únicos, ofreciendo un enfoque más creativo e innovador (Heigl, 2025).

El auge de la GenAI ha ampliado significativamente las capacidades de la IA, transformándola en un aliado activo en lugar de ser únicamente una herramienta analítica orientada a comprender el pasado o predecir el futuro. Hoy, se ha convertido en un motor de innovación en diversas áreas como la investigación médica, la educación personalizada y el avance científico. Modelos como GPT, Llama y Gemini han demostrado una notable capacidad para comprender, generar e interactuar con el lenguaje humano, de manera que simulan procesos cognitivos. Los LLM no solo predicen palabras sucesivas, sino que son capaces de estructurar complejas relaciones lingüísticas y contextuales, lo que les permite generar textos coherentes y relevantes en una amplia gama de tareas (Han et al., 2024).

En la Tabla 1 se muestra la evolución de la GenAI, se puede dividir en cuatro etapas históricas, cada una marcada por un aumento en la complejidad, autonomía y capacidad generativa. Este progreso muestra la transición de sistemas rígidos, basados en conocimiento explícito, a modelos flexibles que aprenden representaciones latentes a partir de grandes volúmenes de datos.

Tabla 1

Etapas Históricas de la Inteligencia Artificial Generativa

Etapa	**Periodo**	**Tecnología**	**Ejemplo**
Sistemas Basados en Reglas	Década de 1950 en adelante	La generación de contenido sigue reglas explícitas diseñadas por expertos humanos y almacenadas en una base de conocimiento.	Sistemas expertos, programas de generación de texto temprana.
Algoritmos Basados en Modelos	Décadas de 1980 a 2000	La generación se basa en modelos estadísticos o físicos. Expansión a campos como el aprendizaje automático y los gráficos por computadora.	Generación de animación por computadora, modelos gráficos estadísticos.
Metodologías Generativas Profundas	Década de 2010	Uso de redes neuronales profundas para aprender distribuciones de datos complejas.	Redes Generativas Antagónicas (GANs), Autoencoders Variacionales (VAEs).
Modelos Fundacionales	Década de 2020 en adelante	Aprovechamiento del escalado masivo en tamaño del modelo (parámetros) y datos de entrenamiento, basado en la arquitectura Transformer.	Modelos de la serie GPT, LLaMA, Gemini, PaLM.

Nota. Adaptado de (He et al., 2025)

El desarrollo de los LLM se organiza en dos etapas complementarias, las cuales son preentrenamiento y ajuste fino. En la primera, el modelo aprende de un corpus masivo mediante aprendizaje auto supervisado, adquiriendo representaciones profundas de sintaxis, semántica y conocimiento general. Esta fase genera un modelo fundacional amplio y adaptable del lenguaje. En la segunda, el ajuste fino adapta el modelo a tareas específicas con datos etiquetados, optimizando su desempeño en dominios como medicina, derecho o programación. Mientras el preentrenamiento requiere gran capacidad computacional, el ajuste fino es más eficiente, facilitando la personalización de modelos fundacionales para aplicaciones especializadas (Anisuzzaman et al., 2025).

El rendimiento de los LLM, según las leyes de escalado, tiende a mejorar con modelos más grandes, mayor cantidad de datos y más recursos computacionales. No obstante, el fenómeno de subescalado revela que estas mejoras se ralentizan debido a redundancia y a asignación ineficiente de recursos entre modelo y datos. Esto implica que el incremento de tamaño no garantiza siempre un desempeño significativamente superior. Por ello, los investigadores se han desplazado hacia la eficiencia de recursos y la calidad de los datos, subrayando la importancia de conjuntos precisos, completos y diversos, así como la eliminación de duplicados. La evolución futura de los LLM dependerá tanto del escalado de modelos como de la ingeniería de datos de alta calidad (Chen et al., 2025).

En los LLM se observa un fenómeno conocido como capacidades emergentes, habilidades que no se presentan en modelos pequeños, pero surgen abruptamente al superar cierto umbral de escala, como la aritmética de varios pasos, el razonamiento simbólico o la resolución de preguntas de sentido común. Estas incluyen el aprendizaje en contexto y el razonamiento en cadena de pensamiento, donde el modelo puede descomponer problemas complejos en pasos intermedios sin entrenamiento específico. Algunos consideran que estas capacidades son genuinamente emergentes por la interacción de miles de millones de parámetros, mientras que otros sugieren que podrían reflejar mejoras graduales ocultas en modelos más pequeños. Este debate es clave para la seguridad y gobernanza de la IA, pues capacidades impredecibles podrían generar comportamientos riesgosos en sistemas de mayor escala (Berti et al., 2025).

Una de las limitaciones más importantes de los LLM es su propensión por alucinar, es decir, generar contenido que, aunque estructuralmente correcto, puede ser factualmente incorrecto o contradictorio. Este fenómeno es un desafío crítico en áreas de alto riesgo, como la medicina o las finanzas. Las alucinaciones pueden deberse a datos de entrenamiento sesgados, conocimiento incompleto o errores en el proceso de inferencia. Para mitigar este problema, se han desarrollado estrategias como la arquitectura RAG, que combina la capacidad generativa del LLM con la recuperación de información de fuentes externas actualizadas, permitiendo producir respuestas más precisas y citables. Otras técnicas incluyen el ajuste fino con datos diseñados para reducir alucinaciones y la incorporación de módulos de verificación de hechos durante la generación de contenido (Kostikova et al., 2025).

Los LLM, al ser entrenados con grandes volúmenes de texto extraídos de internet, pueden absorber y amplificar sesgos sociales, estereotipos y desigualdades presentes en los datos. Estos sesgos pueden manifestarse de manera explícita o implícita, generando resultados injustos como la perpetuación de estereotipos de género o raciales, o discriminación en procesos de selección de personal y evaluación crediticia. La investigación de este fenómeno se centra en tres áreas, detectar los sesgos, evaluarlos y mitigarlos, empleando *benchmarks* específicos como *StereoSet* y el Corpus de Inferencia de Sesgos Sociales (SBIC). Las estrategias de mitigación incluyen mejorar y equilibrar los datos de entrenamiento, el ajuste fino para eliminar asociaciones sesgadas, y técnicas que instruyen al modelo a reconocer y corregir sus propios sesgo (Lin & Li, 2025).

El problema del alineamiento es uno de los desafíos más duraderos en la evolución de la IA, dado que implica asegurar que los sistemas actúen conforme a los valores y fines humanos. Con la autonomía creciente de los modelos, aumenta el riesgo de que sus acciones, aunque coherentes con su programación, deriven en consecuencias no previstas o éticamente problemáticas. Las investigaciones recientes amplían esta preocupación hacia el alineamiento socioafectivo, que busca integrar la empatía y la sensibilidad emocional en las respuestas del modelo, y el alineamiento de valores centrado en el usuario, que permite a las personas ajustar el comportamiento del sistema según sus principios morales. Las fallas de alineamiento, junto con problemas como el sesgo y la alucinación, destacan la brecha entre la habilidad lingüística de los modelos y su comprensión real del mundo, lo que subraya la necesidad de enfoques éticos más profundos (Fan et al., 2025).

2.3 Evolución de los asistentes conversacionales y la arquitectura RAG

La IA conversacional (CAI) se encuentra en una etapa de transformación acelerada que está redefiniendo las dinámicas de interacción entre humanos y máquinas. Este proceso se manifiesta tanto en los avances tecnológicos como en la velocidad sin precedentes con la que el público ha adoptado estas herramientas. El lanzamiento de ChatGPT a finales de 2022, que logró alcanzar los 100 millones de usuarios en apenas dos meses, simbolizó un hito histórico y reveló el enorme potencial de los sistemas conversacionales avanzados. Este acontecimiento impulsó una competencia intensa entre las principales empresas tecnológicas, favoreciendo el desarrollo y despliegue acelerado de modelos como Gemini de Google y otras soluciones especializadas (Sengul et al., 2024).

Antes de la irrupción de los LLM, la CAI se basaba en arquitecturas modulares y orientados a tareas muy específicas. Estos chatbots, fundamentados en técnicas de NLP, relacionaban las consultas del usuario con un conjunto limitado de respuestas predefinidas. Los Sistemas de Diálogo Orientados a Tareas (TOD) eran los que más predominaban, su propósito era asistir al usuario en metas concretas, como realizar reservas o verificar estados de pedidos. Su arquitectura compuesta por módulos de comprensión, seguimiento y decisión requería entrenamiento y ajustes independientes, lo que restringía la escalabilidad. Esta

rigidez se traducía en interacciones poco naturales, motivando la búsqueda de modelos más adaptativos y basados en aprendizaje profundo (Wang et al., 2025).

La evolución hacia la CAI actual fue impulsada por la arquitectura *Transformer* y los mecanismos de atención. Esta innovación dio origen a los LLM, que funcionan como sistemas generativos capaces de predecir con coherencia la siguiente palabra de un texto. A diferencia de los sistemas tradicionales, los LLM integran comprensión, seguimiento contextual y generación de respuestas en una sola red neuronal. Los conocimientos que poseen se encuentran codificados en los parámetros aprendidos durante el preentrenamiento, opera como una base de conocimiento implícita. El éxito de modelos como ChatGPT y Gemini demostró la eficacia de esta arquitectura, eliminando las limitaciones de los sistemas anteriores y ampliando la capacidad de generalización de los sistemas conversacionales (Dam et al., 2024).

Los LLM representan un salto tecnológico sin precedentes, pero su fiabilidad se ve comprometida por la dependencia de su memoria paramétrica. La alucinación, donde los modelos generan respuestas coherentes pero incorrectas, es una debilidad clave, ya que se basan en correlaciones estadísticas en lugar de hechos verificables. También, su conocimiento estático y la falta de trazabilidad interna complican su uso en contextos que requieren información actualizada o verificable. Además, la especialización en dominios concretos sigue siendo arriesgada. La arquitectura RAG, que introduce una memoria externa dinámica para mejorar la precisión, actualidad y verificabilidad de las respuestas, busca superar estas limitaciones (Kostikova et al., 2025).

La arquitectura RAG, propuesta por Lewis et al. (2020), surge como respuesta a las limitaciones de los LLM basados únicamente en memoria paramétrica. Esta arquitectura híbrida combina el conocimiento implícito del modelo, almacenado en sus parámetros, con una memoria no paramétrica proveniente de una base de conocimiento externa, accesible y verificable. Al basar sus respuestas en información recuperada de fuentes autorizadas, la RAG mitiga las alucinaciones, permite el acceso a conocimiento actualizado, mejora la trazabilidad mediante citas y facilita la especialización en dominios concretos sin necesidad de reentrenar el modelo base. Su funcionamiento integra dos fases complementarias, la recuperación de información relevante y la generación de texto contextualizado, estableciendo así un marco más fiable, dinámico y explicable para la inteligencia artificial conversacional.

Recuperación (*Retriever*) es la primera fase del proceso RAG y se encarga de localizar la información más relevante dentro de una base de conocimiento externa. Este proceso comienza con la ingesta y segmentación (*chunking*) de los documentos fuente, que se dividen en fragmentos más pequeños y manejables. Luego, cada fragmento se somete a una vectorización (*embedding*), convirtiéndose en una representación numérica que captura su significado semántico, y se almacena en una base de datos vectorial mediante un proceso de indexación. Cuando el usuario formula una consulta, esta también se vectoriza con el mismo

modelo de *embedding*, permitiendo al recuperador identificar los fragmentos cuyos vectores son más cercanos y relevantes para construir una respuesta contextualizada (Gao et al., 2024).

Generación (*Generator*) es el segundo componente de la arquitectura RAG y suele estar representado por un LLM. A diferencia de los enfoques tradicionales, este módulo no procesa solo la consulta original del usuario, sino una versión aumentada que incluye los fragmentos de texto recuperados por el *retriever*. Estos fragmentos se integran con la pregunta para crear un *prompt* enriquecido, que brinda al modelo un contexto fáctico y específico extraído de la base de conocimiento. Con esta información contextual, el LLM genera respuestas más coherentes, precisas y verificables, minimizando las alucinaciones y permitiendo citar fuentes, lo que aporta trazabilidad y fiabilidad al proceso (Gao et al., 2024).

La investigación en sistemas RAG continúa enfrentando importantes desafíos. Uno de los más relevantes es la calidad de la recuperación, ya que el desempeño del sistema depende de que el *retriever* identifique fragmentos relevantes; una recuperación deficiente puede generar respuestas erróneas. Además, la evaluación de sistemas RAG es una tarea compleja que requiere medir no solo la exactitud de la respuesta final, sino también la relevancia del contexto, la fidelidad de las fuentes y la completitud de la información, lo que impulsa el desarrollo de métricas para una IA auditable y confiable. Otro desafío clave es la eficiencia y escalabilidad, pues la etapa de recuperación agrega latencia que debe optimizarse. Finalmente, la calidad de la base de conocimiento es determinante, ya que fuentes incompletas o desactualizadas degradan el rendimiento del sistema (Gao et al., 2024).

2.4 Bases de datos vectoriales y búsqueda por similitud

Un *embedding* de texto es una representación vectorial densa que proyecta palabras, frases o documentos en un espacio semántico continuo de alta dimensión, donde la proximidad entre vectores refleja la similitud conceptual entre las unidades de texto. Este mapeo permite que elementos lingüísticos con significados similares se ubiquen cerca unos de otros en dicho espacio. La evolución de los modelos de embedding ha sido clave para el progreso del NLP. Los primeros enfoques como Word2Vec generaban *embeddings* estáticos que asignaban un único vector por palabra, sin considerar el contexto en que aparecía. Estos modelos fueron innovadores al capturar relaciones semánticas y analogías, pero presentaban limitaciones, ya que no podían manejar la polisemia ni reflejar los cambios temporales en el significado del lenguaje (Samira Rabinataj, Seyedeh, 2025).

El avance más importante en este campo ha sido la consolidación de los *embeddings* contextuales, impulsados por modelos basados en la arquitectura *Transformer*, como BERT y sus variantes. A diferencia de los *embeddings* estáticos, estos generan representaciones vectoriales dinámicas que se adaptan al contexto completo de cada palabra. Esto permite distinguir con precisión distintos significados de un mismo término, como “banco” en el contexto de una institución financiera o como asiento y capturar matices semánticos con una gran exactitud. Esta innovación no solo es una mejora incremental, sino un salto cualitativo

que fortalece los sistemas RAG, ya que un enfoque basado en *embeddings* estáticos fallaría ante la ambigüedad léxica (Jia & Zhao, 2025).

Las dos métricas más empleadas para cuantificar la proximidad semántica entre los vectores son la Similitud del Coseno y la Distancia Euclidiana. La Similitud del Coseno calcula el coseno del ángulo entre dos vectores, enfocándose en su orientación en lugar de su magnitud, lo que la hace ideal para comparar embeddings de texto, ya que la longitud del vector no tiene relevancia semántica. En cambio, la Distancia Euclidiana evalúa la separación directa entre los puntos en el espacio n-dimensional y es sensible tanto a la magnitud como a la dirección. Sin embargo, cuando los embeddings se normalizan a una longitud unitaria, ambas métricas se vuelven equivalentes en términos de ranking, ya que la distancia euclidiana se convierte en una transformación monotónica de la Similitud del Coseno. Esta equivalencia permite optimizar los cálculos mediante operaciones más simples como el producto punto sin afectar la calidad de los resultados de recuperación (Q. Zhang et al., 2025).

Cuando un sistema gestiona millones de embeddings, la tarea de buscar los vectores más similares se vuelve un desafío computacional debido a la "maldición de la dimensionalidad", que hace que el espacio crezca exponencialmente y degrade la eficacia de los métodos tradicionales. Ante la imposibilidad de realizar una búsqueda exacta del vecino más cercano (KNN) de manera eficiente, se emplean algoritmos de búsqueda aproximada de vecinos más cercanos (ANN). Estos algoritmos sacrifican una mínima parte de la precisión a cambio de mejoras significativas en velocidad y eficiencia. En lugar de garantizar la exactitud de los vecinos, los métodos ANN se enfocan en recuperar puntos con alta probabilidad de relevancia, lo que permite reducir considerablemente el tiempo de respuesta y el consumo de recursos (Zhao, 2024).

Una base de datos vectorial es un sistema diseñado para almacenar, indexar y consultar de manera eficiente grandes colecciones de embeddings de alta dimensionalidad. A diferencia de las bases de datos tradicionales, que optimizan la recuperación de datos mediante coincidencias exactas en campos estructurados o semiestructurados, las bases de datos vectoriales se centran en la búsqueda por similitud, empleando algoritmos ANN para acelerar las consultas. Su arquitectura integra un motor de almacenamiento que persiste los vectores y sus metadatos, y un motor de indexación, que gestiona las estructuras de datos para la recuperación eficiente. Este sistema integra tecnologías de procesamiento de lenguaje natural, algoritmos de búsqueda aproximada y diseño de bases de datos, brindando una solución avanzada para la búsqueda semántica compleja (Zhao, 2024).

En la arquitectura RAG, la base de datos vectorial es responsable de implementar el componente Recuperador. Durante la inferencia, la consulta del usuario se codifica en un vector mediante el mismo modelo de embedding usado para indexar la base de conocimiento. Luego, el sistema ejecuta una búsqueda por similitud ANN en la base de datos vectorial, identificando los k vectores más cercanos según una métrica como la similitud coseno. Los

fragmentos de texto asociados a esos vectores son recuperados y se emplean en la aumentación del *prompt*, concatenándolos con la consulta original para proporcionar contexto relevante. El *prompt* enriquecido es finalmente procesado por el modelo de lenguaje, que genera una respuesta más precisa, contextualizada y fundamentada en datos fácticos, minimizando la posibilidad de errores o alucinaciones (Gao et al., 2024).

2.5 Frameworks de orquestación para aplicaciones LLM

La arquitectura RAG desde un enfoque sistémico representa un mecanismo avanzado de orquestación de LLM. Administra un flujo de trabajo complejo que integra la interacción con bases de conocimiento externas, la ingeniería de *prompts* en tiempo de ejecución mediante la combinación de la consulta con el contexto recuperado y la posterior invocación de la API del LLM. Este enfoque transforma al LLM de un sistema de caja negra con conocimiento estático en un motor de razonamiento de caja abierta, capaz de utilizar información contextual y actualizada, lo que fortalece la fiabilidad y robustez de los sistemas de inteligencia artificial (IBM, 2025b).

Python se ha establecido como el lenguaje principal en el desarrollo de inteligencia artificial, especialmente en la implementación de arquitecturas RAG. La sencillez de su sintaxis facilita la implementación de conceptos complejos mediante código claro y estructurado, lo que impulsa la colaboración entre equipos multidisciplinarios. Además, su amplio ecosistema de bibliotecas como como NumPy, Pandas, PyTorch, TensorFlow, Hugging Face brindan acceso a modelos de *embeddings* y LLMs de vanguardia de forma eficiente. *Frameworks* como LangChain y LlamaIndex simplifican la orquestación de flujos RAG, y plataformas como Streamlit permiten construir interfaces interactivas con rapidez. Finalmente, su activa comunidad global proporciona recursos y soporte continuo, lo que refuerza su papel central en el campo de la inteligencia artificial (Vanishree, K & Ananya, G, 2024).

LlamaIndex es un framework de orquestación de datos que simplifica la creación de aplicaciones basadas en LLM capaces de interactuar con fuentes de información externas, como en los sistemas RAG. Gestiona el flujo de información desde las fuentes originales hasta el índice vectorial y el modelo de lenguaje. Su objetivo es abstraer la complejidad técnica del manejo de datos, permitiendo a los desarrolladores centrarse en la lógica de la aplicación. Sus funciones incluyen la ingesta de datos mediante conectores, capaces de extraer información desde diversas fuentes, lo que facilita la creación de bases de conocimiento. La segmentación automatiza la división de documentos en fragmentos para mejorar la precisión en la recuperación. Finalmente, construye un índice con *embeddings* y bases vectoriales, creando una memoria externa que puede ser consultada eficientemente (IBM, 2025a).

El modelo de *embeddings* es el núcleo del proceso de indexación y recuperación semántica, ya que traduce el lenguaje humano en representaciones numéricas que las máquinas pueden interpretar matemáticamente. El modelo all-MiniLM-L6-v2, de *Sentence-Transformers*, mapea oraciones y párrafos en vectores de 384 dimensiones dentro de un

espacio vectorial denso, donde la distancia entre vectores refleja la similitud semántica entre los textos. Esto permite una búsqueda basada en significado, superando las limitaciones de los enfoques de palabras clave. Modelos más grandes como all-mpnet-base-v2 ofrecen un rendimiento ligeramente superior en algunos *benchmarks*, pero all-MiniLM-L6-v2 proporciona una precisión similar con menor demanda computacional y tiempos de procesamiento más rápidos, lo que lo hace ideal para entornos con recursos limitados y desarrollo ágil de sistemas RAG (Colangelo et al., 2025).

ChromaDB es una base de datos vectorial de código abierto que facilita el almacenamiento, la indexación y la consulta eficiente de los vectores generados por modelos de embeddings en aplicaciones de IA. Su arquitectura ligera y facilidad de uso la hacen ideal para prototipado rápido y proyectos de pequeña y mediana escala. En una arquitectura RAG, su función principal es almacenar los vectores y sus metadatos, lo que permite búsquedas contextuales precisas. Además, utiliza algoritmos ANN para localizar rápidamente los vectores más similares a una consulta y la similitud del coseno se emplea como métrica principal, evitando altos costos. Finalmente, al recibir una consulta, ChromaDB, aplica la búsqueda ANN y devuelve el top-K de fragmentos más relevantes, los cuales enriquecen el prompt del LLM para generar respuestas precisas y contextualizadas (S Kamalov et al., 2025).

La etapa final del *pipeline* RAG es la generación, donde un LLM integra la consulta del usuario con el contexto recuperado para producir una respuesta contextualizada. La elección del LLM es una decisión de diseño importante que requiere equilibrar las ventajas de los modelos propietarios y las alternativas de código abierto. Modelos comerciales como Gemini de Google o GPT de OpenAI, son los líderes en capacidad y rendimiento, con innovaciones como ventanas de contexto de millones de *tokens*, aunque presentan limitaciones en costo, latencia, opacidad y fiabilidad en contextos extensos. Modelos de código abierto, como *Llama*, *Mistral o Gemma*, ofrecen mayor control, transparencia, privacidad y costo-efectividad al permitir su autoalojamiento y ajuste (*fine-tuning*) para tareas específicas, consolidándose como alternativas más adaptables y sostenibles para entornos donde la autonomía y la seguridad son prioritarias (Savage et al., 2025).

El componente final de una arquitectura LLM es la interfaz de usuario (UI), siendo el punto de contacto entre el usuario y el *backend* del sistema RAG. En una investigación, la capacidad de construir y modificar rápidamente esta interfaz es clave para validar hipótesis y presentar resultados. Streamlit destaca como un *framework* de código abierto en Python que transforma scripts de datos o IA en aplicaciones web interactivas con un mínimo esfuerzo. Su principal ventaja es eliminar la complejidad del desarrollo *frontend*, permitiendo crear interfaces funcionales sin usar HTML, CSS o JavaScript. Además, su enfoque reactivo acelera el prototipado, ya que las aplicaciones se actualizan automáticamente al modificar el código. Integrado con Python, facilita la visualización de resultados de los modelos, simplificando la conexión con el pipeline RAG y ofreciendo a los investigadores y científicos de datos una herramienta práctica para construir interfaces interactivas (Snowflake Inc., 2025).

Para la validación cuantitativa del prototipo RAG, se utiliza el framework Ragas (Retrieval Augmented Generation Assessment). Ragas es un marco de evaluación que implementa el paradigma de "LLM-as-a-Judge", el cual utiliza LLM para evaluar la calidad de las salidas de otros modelos sin necesidad de intervención humana constante. A diferencia de las métricas tradicionales de NLP (como BLEU o ROUGE), que se basan en la coincidencia léxica palabra por palabra, Ragas evalúa la coherencia semántica y la corrección factual descomponiendo la evaluación en cuatro métricas que analizan tanto el componente de recuperación como el de generación (Es et al., 2025).

Faithfulness (Fidelidad) evalúa la integridad factual del componente generador, midiendo el grado en que las respuestas producidas se derivan exclusivamente del contexto recuperado, evitando las alucinaciones. El proceso de cálculo inicia cuando el modelo evaluador descompone la respuesta generada en un conjunto de afirmaciones atómicas $S = \{s_1, s_2, \dots, s_n\}$. Luego, se verifica cada afirmación s_i contra el contexto recuperado para determinar si puede ser inferida lógicamente de este. La puntuación de fidelidad F se define como la proporción de afirmaciones verificables sobre el total de afirmaciones producidas:

$$F = \frac{|S_{verificadas}|}{|S_{total}|} = \frac{1}{n}\sum_{i=1}^{n} v_i$$

Donde $v_i \in \{0,1\}$ representa el veredicto de consistencia para cada sentencia. Un valor de 1 garantiza que el sistema no ha inventado información externa al corpus proporcionado (Explodinggradients, 2025).

Answer Relevancy (Relevancia de la Respuesta) mide la pertinencia de la salida del sistema respecto a la intención del usuario. Esta métrica no evalúa la veracidad, sino la alineación semántica, penalizando respuestas que, aunque sean verdaderas, no responden directamente a la pregunta formulada. El algoritmo emplea una técnica de ingeniería inversa: el LLM genera n preguntas sintéticas probables $\{q_1, q_2, \dots, q_n\}$ basándose únicamente en la respuesta obtenida. Luego, se calculan los *embeddings* tanto de la pregunta original del usuario (E_Q) como de las preguntas sintéticas (E_{q_i}). La puntuación final se obtiene mediante el promedio de la similitud coseno entre estos vectores:

$$AR = \frac{1}{n}\sum_{i=1}^{n} \text{sim}(E_Q, E_{q_i})$$

Este enfoque vectorial permite cuantificar la proximidad semántica en un espacio continuo, donde un puntaje cercano a 1 indica una alta correlación entre la respuesta generada y la necesidad de información original (Explodinggradients, 2025).

Context Precision (Precisión del Contexto) analiza la calidad del ranking de los documentos recuperados. Esta métrica es fundamental para determinar si el sistema es capaz de priorizar la información relevante sobre el ruido documental. Se calcula analizando la

posición de los fragmentos relevantes dentro de la lista de resultados recuperados, penalizando aquellos casos en los que la información útil aparece en posiciones inferiores. La formulación matemática corresponde a la media de la precisión a nivel k $(Precision@k)$:

$$CP = \frac{\sum_{k=1}^{K} (Precision@k \times v_k)}{Total\ de\ Items\ Relevantes}$$

En esta ecuación, el término v_k es un indicador binario que señala si el documento en la posición k es relevante (Explodinggradients, 2025).

Context Recall (Exhaustividad del Contexto) mide la capacidad del sistema para recuperar toda la información necesaria para responder a una consulta ideal. A diferencia de la precisión, que mide la pureza del contexto, la exhaustividad mide su completitud. El cálculo se realiza comparando las afirmaciones presentes en la respuesta de referencia o *Ground Truth* (GT) con el contexto recuperado. El modelo evaluador verifica cuántas de las sentencias del *Ground Truth* pueden ser atribuidas a la información encontrada en el contexto. Se formaliza como:

$$CR = \frac{|GT_{atribuidas}|}{|GT_{total}|}$$

Este indicador es crítico para diagnosticar fallos en la etapa de indexación o búsqueda semántica, ya que un valor bajo sugiere que el motor de búsqueda no está logrando encontrar la evidencia necesaria para responder correctamente. Un valor de 1 indica que el contexto recuperado contiene absolutamente toda la información necesaria para responder correctamente a la pregunta ideal (Explodinggradients, 2025).

Capítulo 3
Metodología

En este capítulo se describe la metodología del estudio, presentando el objetivo general, los objetivos específicos, el alcance y las limitaciones, el enfoque de la investigación. También se exponen las fases que conforman el proyecto, desde la construcción del sistema RAG hasta la evaluación su rendimiento, con el propósito de ofrecer una comprensión completa del proceso metodológico aplicado.

3.1 Objetivo general

Implementar un prototipo de chatbot inteligente, basado en una arquitectura RAG para automatizar la resolución de consultas frecuentes en la mesa de ayuda de una empresa de servicios, con el fin de reducir su carga operativa.

3.2 Objetivos específicos

1. Diseñar la arquitectura del sistema RAG, seleccionando y justificando un conjunto de herramientas de código abierto para la orquestación del pipeline, el almacenamiento de vectores y el prototipado de la interfaz (OE1).

2. Implementar un prototipo funcional del chatbot, incluyendo la ingesta y procesamiento de un corpus de conocimiento documental definido (OE2).

3. Evaluar el prototipo con métricas cuantitativas y cualitativas sobre precisión del recuperador y fidelidad de las respuestas generadas (OE3).

3.3 Alcance y limitaciones

Es importante gestionar las expectativas y asegurar la viabilidad de la investigación. El alcance de este trabajo comprende el desarrollo de un prototipo funcional concebido como una prueba de concepto para validar la arquitectura y la metodología propuestas, no como un sistema listo para producción. Además, el sistema operará sobre un corpus de conocimiento delimitado, compuesto por un conjunto específico y acotado de documentos, por lo que la generalización de los resultados a otros dominios o tipos de documentos queda fuera de este estudio. Finalmente, la validación del rendimiento se basará exclusivamente en métricas cuantitativas predefinidas, sin incluir estudios de experiencia de usuario a gran escala ni análisis cualitativos de las interacciones.

Se reconocen limitaciones asociadas al diseño del estudio y a las tecnologías utilizadas. La calidad, coherencia y veracidad de las respuestas dependen directamente del desempeño del LLM empleado, considerado en esta investigación como una caja negra, sin acceso a sus procesos internos. El rendimiento del sistema RAG está condicionado por la calidad, precisión y actualidad del corpus de conocimiento, por lo que errores o sesgos presentes en los documentos fuente se reflejarán en los resultados. La arquitectura basada en ChromaDB está optimizada para entornos de desarrollo y prototipado, pero su capacidad de escalamiento es

limitada y requeriría migrar a soluciones distribuidas para manejar grandes volúmenes de datos. La evaluación es cuantitativa y sistemática, pero depende de juicios generados por un LLM, como en el framework RAGAS, lo que puede introducir sesgos y no sustituye la revisión experta humana.

3.4 Tipo de enfoque de la investigación

Esta investigación se enmarca en un modelo y diseño metodológico ampliamente reconocido en el campo de la ingeniería de software. Se adopta la investigación aplicada en ingeniería de software, que emplea un enfoque sistemático y cuantificable en el desarrollo, operación y mantenimiento del software. Este enfoque tiene como objetivo no solo generar conocimiento, sino también resolver problemas prácticos mediante soluciones tecnológicas. En este marco, se emplea el *Design Science*, apropiado cuando la contribución principal consiste en la creación de un artefacto innovador. El prototipo de chatbot RAG, que busca facilitar el acceso a información contenida en documentos, es un ejemplo de este enfoque. Además, permite evaluar su utilidad y desempeño (Knauss, 2021).

El trabajo es experimental debido a que se construye un sistema controlado, se manipulan variables como las consultas de entrada y se mide el desempeño frente a métricas definidas. El experimento central consiste en aplicar el protocolo de evaluación cuantitativa al prototipo para resolver el tercer objetivo específico. El sistema procesa datos cualitativos, pero el enfoque de investigación es cuantitativo, ya que la evaluación y análisis se sustentan en datos numéricos. Las métricas, incluyendo precisión y fidelidad, se expresan en puntuaciones escalares que permiten un análisis estadístico riguroso y la comparación objetiva de resultados.

3.5 Fases del desarrollo del proyecto

El presente trabajo de investigación se estructura en fases secuenciales resumidas en la Tabla 2, cada una con actividades y entregables definidos, asegurando un progreso sistemático y controlado a lo largo del proyecto.

Tabla 2

Fases del desarrollo del proyecto

Fase	Título	Objetivo
1	Preparación y Base de Conocimiento	Crear la base de conocimiento (limpieza, *chunking*).
2	Implementación del Prototipo	Construir el *pipeline* de indexación (LlamaIndex, ChromaDB).
3	Desarrollo de Interfaz	Crear la aplicación web de prueba (Streamlit).
4	Diseño del Protocolo de Evaluación	Crear el *benchmark* (dataset, *ground truth*).

Fase	Título	Objetivo
5	Recolección de Datos	Ejecutar las pruebas y registrar los resultados (JSON).
6	Análisis Cuantitativo	Medir el rendimiento del sistema con RAGAS.

La fase inicial se enfoca en la preparación del entorno de desarrollo y en la creación de la base de conocimiento del sistema RAG. En esta etapa, se selecciona y recopila el corpus de documentos, los cuales constituirán la base de conocimiento del chatbot. Luego, se preprocesarán los datos, con el objetivo de limpiar los documentos y eliminar elementos irrelevantes, como encabezados, pies de página o formatos anómalos que podrían generar ruido durante el proceso de recuperación. Posteriormente, se implementa una estrategia de *chunking*, dividiendo los documentos procesados en fragmentos más pequeños, conocidos como *chunks*. El tamaño y el solapamiento de los *chunks* afectan directamente el equilibrio entre la conservación del contexto semántico y la precisión en la recuperación de información. Se establecerán valores fijos tanto para el tamaño de los *chunks* como para el porcentaje de solapamiento, definidos según la estructura de los documentos del corpus.

La segunda fase se centra en la implementación del prototipo, utilizando una pila tecnológica de código abierto seleccionada. En esta etapa, se construye el pipeline de indexación y recuperación de información, con LlamaIndex como framework de orquestación, dado su soporte para fuentes de datos diversas, estrategias de indexación flexibles y control detallado sobre los procesos de recuperación y síntesis. Esta herramienta reduce la complejidad técnica, permitiendo enfocar los esfuerzos en la lógica del sistema. Se utiliza ChromaDB como base de datos vectorial, seleccionada por su código abierto, facilidad de instalación local e integración con LlamaIndex, ideal para el desarrollo experimental con un enfoque en rapidez y simplicidad. Durante la indexación, los fragmentos de texto se transforman en *embeddings* vectoriales mediante un modelo preentrenado como all-MiniLM-L6-v2 de *Sentence Transformers*, y se almacenan junto con texto y metadatos en ChromaDB para la búsqueda de similitud semántica.

La tercera fase se enfoca en el desarrollo de la interfaz de usuario, orientada al prototipado y validación funcional del sistema RAG en un entorno interactivo. Se crea una aplicación web que permite al usuario interactuar directamente con el modelo, visualizando de manera dinámica los resultados del proceso de recuperación y generación. Se selecciona Streamlit como herramienta de prototipado, por su capacidad para convertir scripts de Python en aplicaciones web interactivas y visualmente atractivas con poco código, sin necesidad de conocimientos en desarrollo *frontend*. La interfaz incluye un campo de texto donde el usuario puede formular preguntas. Al enviarlas, el motor de consultas de LlamaIndex ejecuta todo el pipeline RAG, convierte la pregunta en un vector, busca los fragmentos relevantes en

ChromaDB, construye un *prompt* enriquecido con ese contexto, lo envía al modelo de lenguaje para generar la respuesta y presenta el resultado de manera clara y accesible.

La cuarta fase se centra en el diseño del protocolo de evaluación sistemática, con la finalidad es establecer una metodología cuantitativa rigurosa para medir el desempeño del prototipo y servir como puente hacia la etapa de análisis de datos. En esta fase se construye un *dataset* de validación que actúa como referencia o *benchmark*, conformado por un conjunto representativo de preguntas que un usuario típico podría plantear al sistema. Para cada pregunta, se redacta manualmente una respuesta ideal y verificada, denominada *ground truth*, que sirve de base para calcular métricas dependientes de una referencia, como el *Context Recall*. Además, se definen las métricas de evaluación, seleccionando aquellas que permiten cuantificar las principales dimensiones del rendimiento del sistema RAG. Estas métricas, que se detallan en la siguiente sección, fueron elegidas por su capacidad para evaluar de manera integral la precisión, coherencia y efectividad del prototipo desarrollado.

La fase de recolección de datos utiliza el sistema RAG como herramienta principal, configurado para registrar sus entradas y salidas durante la ejecución del protocolo de evaluación. El sistema se ejecuta y almacena cuatro elementos clave en un formato estructurado JSON, para cada pregunta del *dataset* de validación. Estos elementos son la *question* (consulta de entrada del conjunto de validación), los *retrieved contexts* (fragmentos de texto identificados como relevantes por el recuperador mediante ChromaDB y LlamaIndex), la *generated answer* (respuesta generada por el modelo de lenguaje tras procesar la consulta y los contextos recuperados) y la *ground truth* (respuesta ideal anotada manualmente y asociada a cada pregunta del conjunto de validación). Este conjunto de registros, formado por tuplas de (*question, retrieved contexts, generated answer, ground truth*), constituye la base empírica para el análisis cuantitativo del desempeño del sistema.

El análisis cuantitativo de rendimiento se realizará con RAGAS, un framework de código abierto reconocido por su eficacia en evaluación de pipelines RAG. Esta herramienta ofrece métricas que permiten evaluar de forma independiente y conjunta los componentes de recuperación y generación del sistema. Se han seleccionado cuatro métricas clave que permiten una visión completa de la calidad del sistema. Esto permite identificar las fortalezas y debilidades del prototipo, considerando que la arquitectura RAG se basa en la calidad de la información recuperada y la de la respuesta generada. Un bajo desempeño en el recuperador, como una recuperación insuficiente de contexto, puede afectar la efectividad del generador, y viceversa. Por ello, es esencial evaluar ambos componentes de manera independiente para garantizar un diagnóstico completo del sistema (Zhou et al., 2024).

La matriz de evaluación del sistema RAG se basa en cuatro métricas clave que ofrecen una visión integral del rendimiento del prototipo. En el componente de recuperación, *Context Precision* mide la proporción de fragmentos realmente relevantes entre los recuperado y *Context Recall* evalúa la capacidad del sistema para recuperar toda la información necesaria

para una respuesta completa. En el componente de generación, Faithfulness indica qué tan bien se fundamenta la respuesta en el contexto recuperado, evitando la invención de información, y *Answer Relevancy* analiza si la respuesta se ajusta con la intención del usuario. Estas métricas con valores entre 0 y 1, permite diagnosticar con precisión la calidad del sistema, identificando posibles deficiencias en recuperación o generación.

Capítulo 4
Resultados y discusión

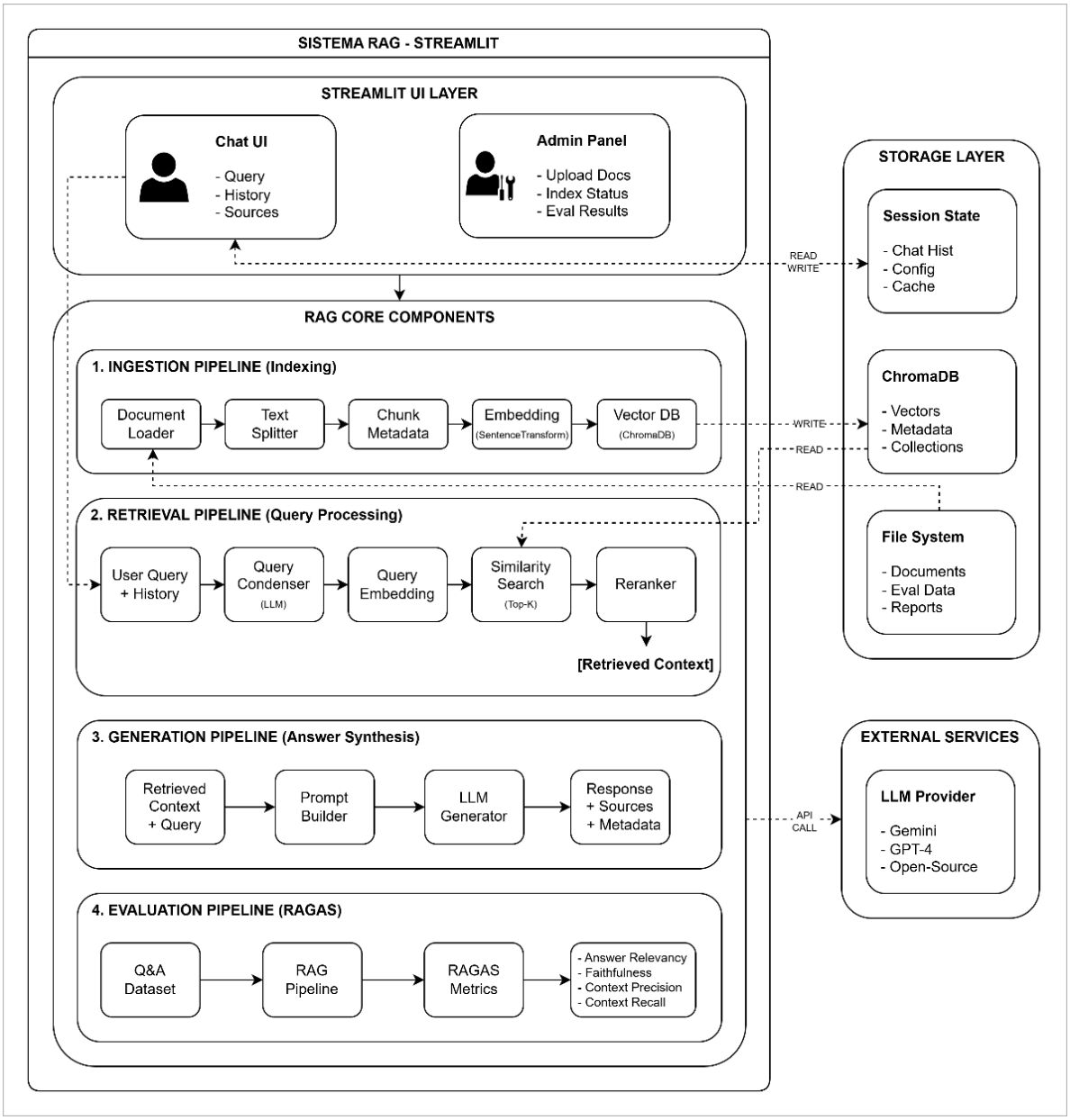
El capítulo presenta los resultados de la implementación y evaluación de un prototipo de chatbot inteligente basado en la arquitectura RAG. Primero, se describe el diseño e implementación del prototipo; luego, se muestran los resultados cuantitativos del protocolo de evaluación; y finalmente, se analizan e interpretan dichos resultados.

4.1 Diseño del prototipo

Se presenta la arquitectura del prototipo en la Figura 1, la cual está conformada por cuatro capas principales que incluyen la interfaz de Streamlit, los componentes centrales RAG, la capa de almacenamiento y los servicios externos, junto con las interconexiones entre ellas.

Figura 1

Diagrama de Arquitectura



Este diagrama de arquitectura presenta una visión global y funcional del prototipo RAG, dividiéndolo en cuatro componentes lógicos principales.

1. *Streamlit UI* actúa como la capa de presentación. Es el único punto de entrada para los dos actores del sistema: el Usuario (que interactúa con el chat) y el Administrador (que carga documentos y revisa evaluaciones).
2. *RAG Core Components* es el cerebro lógico del sistema. Agrupa los cuatro pipelines de procesamiento que definen la funcionalidad completa: Ingestión (aprender), Recuperación (buscar), Generación (responder) y Evaluación (validar).
3. *Storage Layer* es la memoria del sistema. Centraliza todos los componentes de almacenamiento, tanto persistentes (como ChromaDB para vectores y el *File System* para documentos) como volátiles (como el *Session State* para el historial del chat).
4. *External Services* aísla las dependencias externas, en este caso, el LLM Provider, que es consumido por el núcleo para tareas de generación y evaluación.

La decisión de diseño más relevante en el código fue adoptar la arquitectura limpia, la cual garantiza una estructura modular y de fácil mantenimiento. Esta arquitectura establece una separación de responsabilidades entre las capas de interfaz, casos de uso, dominio e infraestructura, lo que resultó fundamental para el desarrollo del prototipo. Esta estructura permitió implementar y probar de manera independiente cada componente del chatbot, como los módulos de ingestión y evaluación. Además, la arquitectura limpia proporciona independencia tecnológica, ya que el núcleo del sistema, representado por los casos de uso implementados en LlamaIndex, no depende directamente de la base de datos ni de la interfaz de usuario, sino de abstracciones definidas en el dominio. Esta característica asegura la viabilidad y escalabilidad de la propuesta, al permitir sustituir las tecnologías empleadas en el prototipo por soluciones más robustas en el entorno de producción, sin necesidad de modificar la lógica de negocio.

Antes de detallar los componentes de software, es fundamental definir el entorno de hardware sobre el cual se construyó y evaluó el prototipo. Las especificaciones del sistema en la Tabla 3 no solo garantizan la replicabilidad del estudio, sino que también contextualizan y justifican las decisiones de diseño tomadas en los siguientes pilares (OE1). La arquitectura fue diseñada para operar eficientemente dentro de las limitaciones de una estación de trabajo accesible, en lugar de depender de infraestructura en la nube de alto costo.

Tabla 3

Especificaciones del entorno de ejecución

Componente	Especificación	Justificación e Impacto en el Diseño
CPU	AMD Ryzen 9 8940HX	Utilizada para el procesamiento general de datos, la orquestación del *pipeline* y la ejecución de componentes no-neuronales.
GPU	NVIDIA GeForce RTX 5060	Componente crítico para la inferencia del LLM y los modelos de *embedding* y *reranking*.
VRAM (GPU)	8 GB	Factor de restricción principal. La limitación de 8 GB de VRAM justificó la elección de modelos ligeros (gemma3:4b, all-MiniLM-L6-v2) que pudieran operar localmente sin sacrificar rendimiento.
RAM (Sistema)	32 GB DDR5	Proporciona el soporte necesario para el manejo en memoria del corpus de conocimiento, la base de datos vectorial (ChromaDB) y los procesos de inferencia.
Entorno	Python 3.12, CUDA 12.9	Define el software base para asegurar la compatibilidad de las bibliotecas y la correcta ejecución de las operaciones en la GPU.

Definido el hardware, el segundo pilar corresponde a la pila tecnológica. Con el fin de cumplir el OE1, la selección de herramientas se basó en tres criterios fundamentales. Se consideró su naturaleza de código abierto. Se valoró la facilidad de integración que ofrecen para el prototipado rápido. Finalmente, se tomó en cuenta su nivel de madurez dentro del ecosistema RAG.

La Tabla 4 presenta de manera detallada los componentes de software seleccionados para las capas de interfaz, orquestación, almacenamiento y evaluación, junto con la justificación de su función dentro de la arquitectura propuesta.

Tabla 4

Justificación de la Pila Tecnológica

Componente	Herramienta	Justificación de la elección
Interfaz de Usuario (UI)	Streamlit	Facilita el prototipado rápido al permitir la creación de una interfaz web interactiva y funcional directamente desde Python, sin requerir desarrollo *frontend*.

Componente	Herramienta	Justificación de la elección
Orquestación RAG	LlamaIndex	Resulta especialmente adecuada para el alcance de una Prueba de Concepto (PoC). Ofrece una abstracción de alto nivel mediante componentes como ChatEngine y VectorStoreIndex, lo que simplifica la lógica del pipeline, reduce el código repetitivo y permite concentrar el desarrollo en el flujo de datos.
Base de Datos Vectorial	ChromaDB	Brinda simplicidad y portabilidad al ser una solución de código abierto que se ejecuta localmente (en proceso). Su uso elimina la complejidad asociada a un servidor de base de datos, por lo que resulta ideal para la experimentación y el prototipado rápido.
Modelo de Embedding	SentenceTransformers (all-MiniLM-L6-v2)	Ofrece un equilibrio óptimo entre rendimiento y eficiencia. Este modelo de código abierto se ejecuta localmente, proporciona representaciones semánticas de alta calidad sin costo de API y mantiene una baja latencia, garantizando una búsqueda de similitud eficiente.
Framework de Evaluación	RAGAS	Constituye la herramienta estándar de la industria para la evaluación cuantitativa de sistemas RAG. De código abierto, proporciona las métricas necesarias requeridas por la metodología de la tesis.

El segundo pilar de la implementación es la configuración del experimento. Para asegurar la reproducibilidad del estudio y un rendimiento consistente, se definió un conjunto de parámetros fijos que no variaron durante las pruebas.

La Tabla 5 detalla estos valores, explicando el racional detrás de cada elección y su impacto en los pipelines de Ingesta, Recuperación y Generación.

Tabla 5

Justificación de Parámetros de Configuración

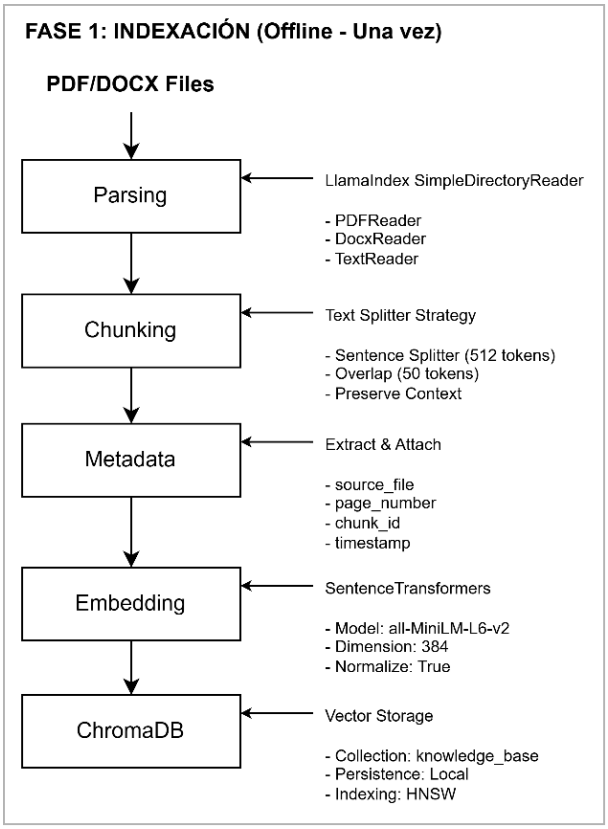
Parámetro	Fase	Valor	Justificación
chunk_size	Ingesta	512 (*tokens*)	Representa un equilibrio entre contexto y precisión. El tamaño es lo suficientemente pequeño para generar vectores semánticamente densos, pero

Parámetro	Fase	Valor	Justificación
			también lo bastante grande para conservar el contexto local de cada oración.
chunk_overlap	Ingesta	50 (tokens)	Mitiga los cortes de información al garantizar que las ideas o frases que se ubican en el límite de un fragmento no se pierdan, manteniendo así la coherencia semántica entre los distintos chunks.
similarity_top_k	Recuperación	10 (*chunks*)	Favorece la maximización del *recall* o tasa de recuperación al obtener un conjunto amplio de fragmentos potencialmente relevantes, asegurando que la información necesaria esté disponible para la siguiente fase del proceso.
rerank_top_n	Recuperación	3 (*chunks*)	Contribuye a mejorar la precisión del sistema. El *reranker* selecciona los tres fragmentos más relevantes entre los inicialmente recuperados, lo que permite construir un contexto más denso y eliminar información irrelevante antes de la generación.
temperature	Generación	0.1	Favorece la generación de respuestas fácticas y deterministas. Un valor bajo reduce la aleatoriedad del modelo, promoviendo respuestas basadas estrictamente en el contexto disponible y disminuyendo la probabilidad de alucinaciones.

El pipeline de ingesta es el proceso *offline* responsable de enseñar al sistema. Su objetivo es convertir el corpus de documentos no estructurados (PDF/DOCX) en una base de conocimiento vectorial consultable. El flujo de este proceso, detallado en la Figura 2, se ejecuta una vez por cada documento que se añade al sistema.

Figura 2

Fase 1: Indexación



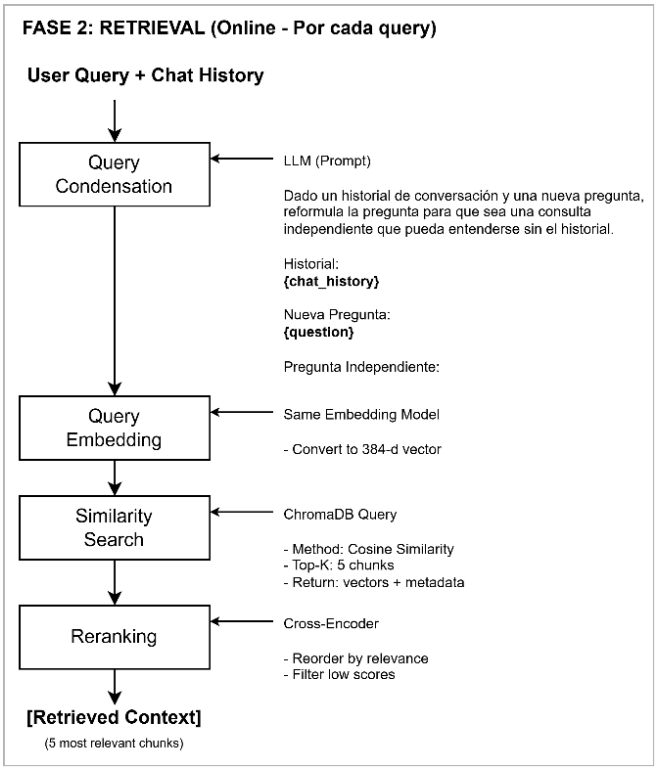
El proceso sigue los siguientes pasos:

1. *Parsing*: El *Document Loader* (utilizando LlamaIndex SimpleDirectoryReader) lee los archivos fuente. Este componente utiliza internamente lectores específicos, como PdfReader o DocxReader, para extraer el texto crudo.
2. *Chunking*: El texto extraído pasa al *Text Splitter* (implementado como *SentenceSplitter*). Este divide el texto en fragmentos (chunks) más pequeños, con un tamaño y solapamiento predefinidos (512 tokens con 50 de solapamiento), un paso crucial para balancear el contexto y la precisión.
3. Metadata: A cada *chunk* se le adjuntan metadatos relevantes, como el *source_file* (nombre del archivo), *chunk_id* y *timestamp*, que serán útiles posteriormente para la citación de fuentes.
4. *Embedding*: Cada chunk de texto es procesado por el modelo de *embedding* SentenceTransformers (all-MiniLM-L6-v2). Este modelo convierte el contenido semántico del texto en un vector numérico de alta dimensionalidad.
5. *Storage*: Finalmente, el *Vector Storage* (nuestra interfaz con ChromaDB) almacena cada *chunk*. Guarda el texto, sus metadatos y su vector (*embedding*) en una colección persistente, completando la indexación.

El pipeline de recuperación es un proceso en línea que, con cada consulta, busca la información más relevante en la base de conocimiento. El flujo, visible en la Figura 3, es el núcleo de la "R" (*Retrieval*) en RAG.

Figura 3

Fase 2: Retrieval



El procedimiento se desarrolla a través de las siguientes etapas:

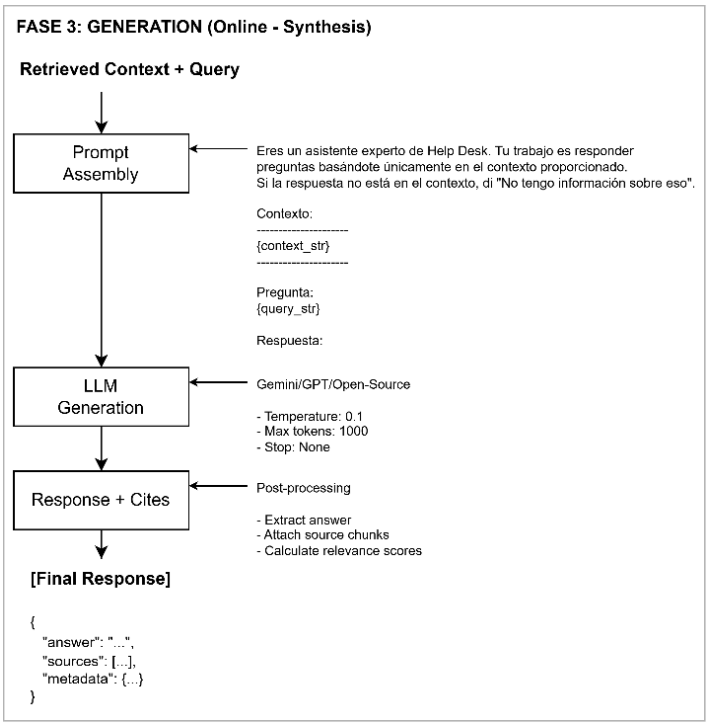
1. *Query Condensation*: El sistema recibe la consulta (*User Query*) y el historial de la conversación (*Chat History*). Para manejar el contexto conversacional, un LLM reformula la consulta (usando el *prompt* de condensación) para crear una "pregunta independiente" que no dependa del historial.
2. *Query Embedding*: La pregunta condensada (ahora una cadena de texto semánticamente completa) se pasa por el mismo modelo SentenceTransformers para convertirla en un vector de consulta.
3. *Similarity Search*: Este vector de consulta se usa para realizar una búsqueda de similitud (similitud de coseno) en ChromaDB. La base de datos devuelve los 10 *chunks* cuyos vectores son más cercanos al vector de la consulta.
4. *Reranking*: Los 3 *chunks* recuperados, que son semánticamente similares, pero no necesariamente la respuesta más precisa, se pasan a un *Cross-Encoder (Reranker).* Este modelo, más costoso, pero más preciso, reordena los *chunks* por relevancia real a la pregunta, filtrando el ruido.

5. *Context Retrieval*: El sistema selecciona los 3 *chunks* mejor clasificados por el *Reranker*. Este conjunto final de texto se denomina *[Retrieved Context]* y es la salida de este pipeline.

La fase de recuperación, detallada en la Figura 4, se lleva a cabo en línea inmediatamente después de la recuperación de datos. Su objetivo es utilizar el contexto relevante para generar una respuesta coherente y precisa.

Figura 4

Fase 3: Generation



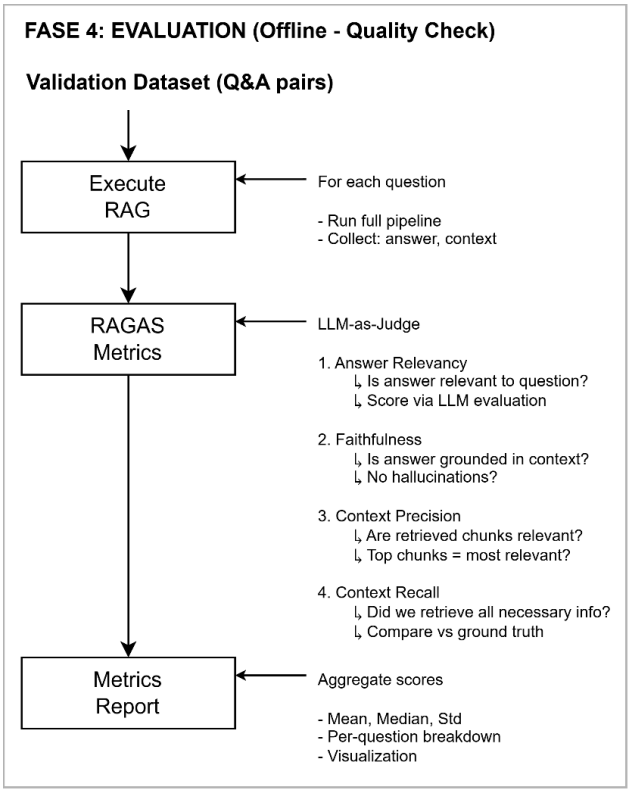
El procedimiento se lleva a cabo en las siguientes etapas:

1. *Prompt Assembly*: Los [*Retrieved Context*] (*chunks* de la Fase 2) y la *Query* (pregunta condensada) se insertan en una plantilla de *prompt* personalizada. Como se muestra en la figura, esta plantilla instruye al LLM sobre su rol ("Eres un asistente experto...") y sus restricciones ("basándote únicamente en el contexto...").
2. *LLM Generation*: El *prompt* aumentado final se envía al LLM. Se configuran parámetros clave como temperatura (0.1, para respuestas objetivas) y tokens máximos para controlar la salida.
3. *Response + Cites*: El LLM genera la respuesta en lenguaje natural. El sistema recibe este texto y lo empaqueta con los metadatos de los *chunks* que se usaron como contexto (las fuentes). Esta es la [*Final Response*] que se devuelve al usuario.

La fase final consiste en el proceso de validación cuantitativa del prototipo, que se realiza a través de un pipeline offline diseñado para evaluar el rendimiento del sistema en comparación con un *benchmark*. El flujo de evaluación se muestra en la Figura 5.

Figura 5

Fase 4: Evaluation



El flujo se organiza en los siguientes pasos:

1. *Validation Dataset*: El proceso se inicia cargando un conjunto de datos de validación (Q&A *pairs*) creado manualmente, que contiene pares de pregunta, contextos más adecuados y respuesta ideal (*ground truth*).
2. *Execute RAG*: Por cada pregunta en el *dataset*, el sistema ejecuta los pipelines de Recuperación (Fase 2) y Generación (Fase 3) de forma completa. Esto recolecta el *generated_answer* (la respuesta del bot) y los *retrieved_contexts* (los *chunks* que usó).
3. *RAGAS Metrics*: El framework RAGAS recibe la tupla completa (*question, generated_answer, retrieved_contexts, ground_truth*).
4. *LLM as Judge*: RAGAS utiliza un LLM como "juez" para puntuar cada respuesta. Como se detalla en la figura, el LLM evalúa las 4 métricas clave:
    - *Faithfulness*: ¿La respuesta se basa en el contexto?
    - *Answer Relevancy*: ¿La respuesta contesta la pregunta?

- *Context Precision*: ¿El contexto recuperado fue relevante?
- *Context Recall*: ¿Se recuperó todo el contexto necesario del *ground truth*?

5. *Metrics Report*: RAGAS agrega las puntuaciones de todas las preguntas y genera un reporte final con las puntuaciones promedio (de 0 a 1) para cada métrica, proporcionando un diagnóstico cuantitativo del rendimiento del prototipo.

4.2 Presentación de resultados de la evaluación

En el subcapítulo anterior se presentó el diseño de la arquitectura y la implementación detallada de los pipelines, lo que permitió dar cumplimiento al primer objetivo específico. En este subcapítulo se aborda la validación empírica del sistema y la exposición de sus resultados. Se demuestra que el artefacto desarrollado constituye un prototipo funcional capaz de realizar la ingesta de un corpus y procesar consultas, cumpliendo así con el segundo objetivo específico. Finalmente, se ejecuta el protocolo de evaluación con el fin de obtener métricas cuantitativas que permitan medir la precisión del recuperador y la fidelidad del generador, respondiendo de este modo al tercer objetivo específico.

Para ejecutar la validación cuantitativa, se preparó un entorno de prueba siguiendo el protocolo de evaluación descrito en la metodología. A continuación, se reporta la configuración específica y los artefactos de datos exactos que se utilizaron para generar los resultados.

Corpus de Conocimiento. En la Fase 1, el prototipo fue indexado utilizando el corpus previamente definido, compuesto por 10 documentos internos de una organización de servicios ficticia llamada Innovatech Solutions. Estos documentos, que se detallan en la Tabla 6, simulan una base de conocimiento realista que estaría disponible para los empleados de un departamento de Soporte de TI y Recursos Humanos.

Tabla 6

Resumen del Corpus de Conocimiento

Categoría	Propósito y Contenido Clave	Cantidad
Políticas y Gobernanza	Establecer reglas corporativas (misión, cultura, horario) y directrices de seguridad de TI (contraseñas, clasificación de datos).	2
Procedimientos Formales (SOPs)	Describir flujos de trabajo con pasos obligatorios, como la solicitud de nuevo software (que requiere aprobación gerencial) o el autoservicio de contraseñas.	2
Guías de Configuración	Proveer instrucciones paso a paso para configurar servicios (correo en móviles, impresoras de red, respuestas automáticas).	3

Categoría	Propósito y Contenido Clave	Cantidad
Guías de Solución de Problemas	Ayudar a los usuarios a diagnosticar y resolver fallos comunes por sí mismos (problemas de VPN, fallos de audio/video en Teams, disco duro lleno).	3

Dataset de Validación: De acuerdo con lo estipulado en el protocolo, se construyó un *dataset* de validación (qa_dataset.json) compuesto por 50 entradas. La estructura detallada del *dataset* se presenta en la Tabla 7 y los pasos de su construcción en la Tabla 8.

Tabla 7

Estructura del dataset de validación

Clave JSON	Descripción del Contenido	Propósito en la Evaluación (RAGAS)
question	Una consulta de usuario simulada, formulada en lenguaje natural.	Representa la entrada (input) que recibe el chatbot para ser evaluada.
contexts	Una lista de fragmentos de texto exactos (citas) extraídos de los 10 documentos	Es la evidencia que el sistema debería recuperar. Se usa para medir la calidad de la recuperación (*Retrieval*).
ground_truth	La respuesta "perfecta" y verificable, sintetizada manualmente solo a partir de los *contexts*.	Es la verdad fundamental. Se usa para medir la calidad de la generación.

Tabla 8

Pasos de creación del dataset

Paso	Tarea	Objetivo
1. Simulación de Consultas	Se analizaron los 10 documentos para diseñar 50 preguntas realistas que un empleado haría.	Asegurar la cobertura total del corpus y simular casos de uso auténticos.
2. Extracción de Evidencia	Se localizaron y copiaron los fragmentos de texto exactos del corpus que contenían la respuesta a cada pregunta.	Establecer el contexto necesario para la evaluación objetiva de la recuperación.
3. Síntesis de la Verdad	Se redactó una respuesta *ground_truth* ideal, basándose *estrictamente* en la evidencia del Paso 2.	Crear la respuesta "perfecta" y fáctica contra la cual medir la generación del modelo.

Paso	Tarea	Objetivo
4. Estructuración JSON	Se ensamblaron los 3 elementos (*question, contexts, ground_truth*) en el formato JSON requerido por RAGAS.	Producir el archivo de validación final listo para la evaluación automatizada.

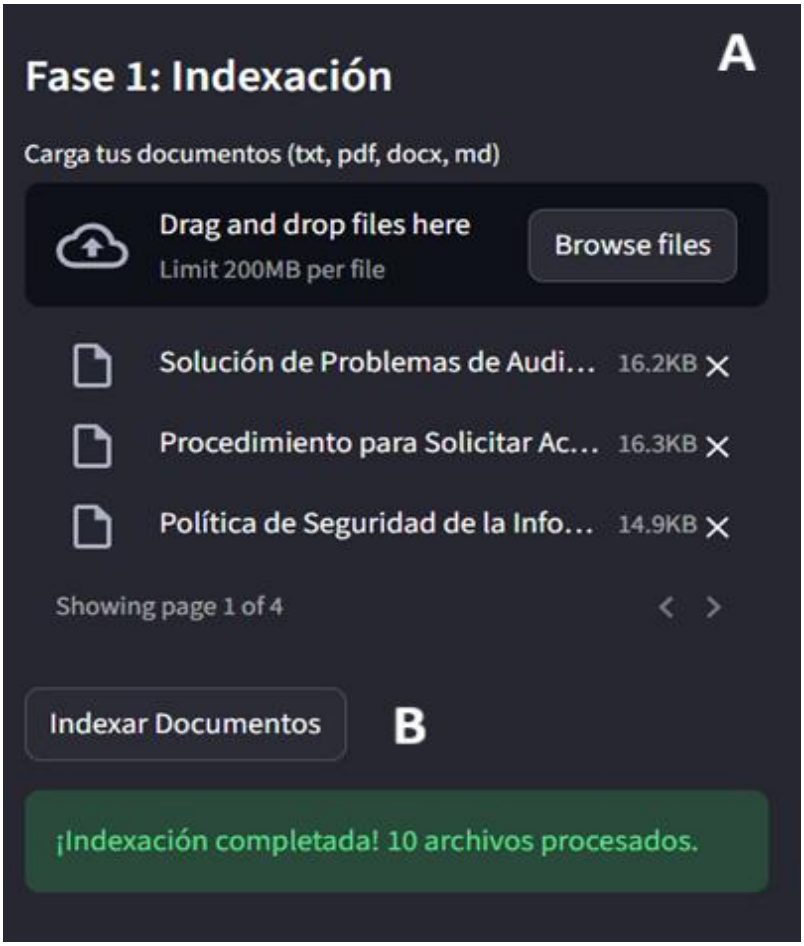
Métricas. El análisis de resultados se centrará en las métricas ya justificadas en la metodología: *Faithfulness, Answer Relevancy, Context Precision y Context Recall*.

Esta sección presenta la evidencia cualitativa de que el sistema implementado es un prototipo funciona. Las siguientes figuras demuestran los flujos de trabajo clave en operación.

La Figura 6 muestra el panel de administración, que permite ejecutar el pipeline de ingesta. Esta funcionalidad es la prueba de la capacidad del sistema para "la ingesta y procesamiento de un corpus", como lo exige el segundo objetivo específico. El panel de administración (A) permite al usuario seleccionar y cargar el corpus de conocimiento. Al ejecutar la indexación (B), el sistema procesa los archivos (Fase 1) y los almacena en ChromaDB, dejando el prototipo listo para las consultas.

Figura 6

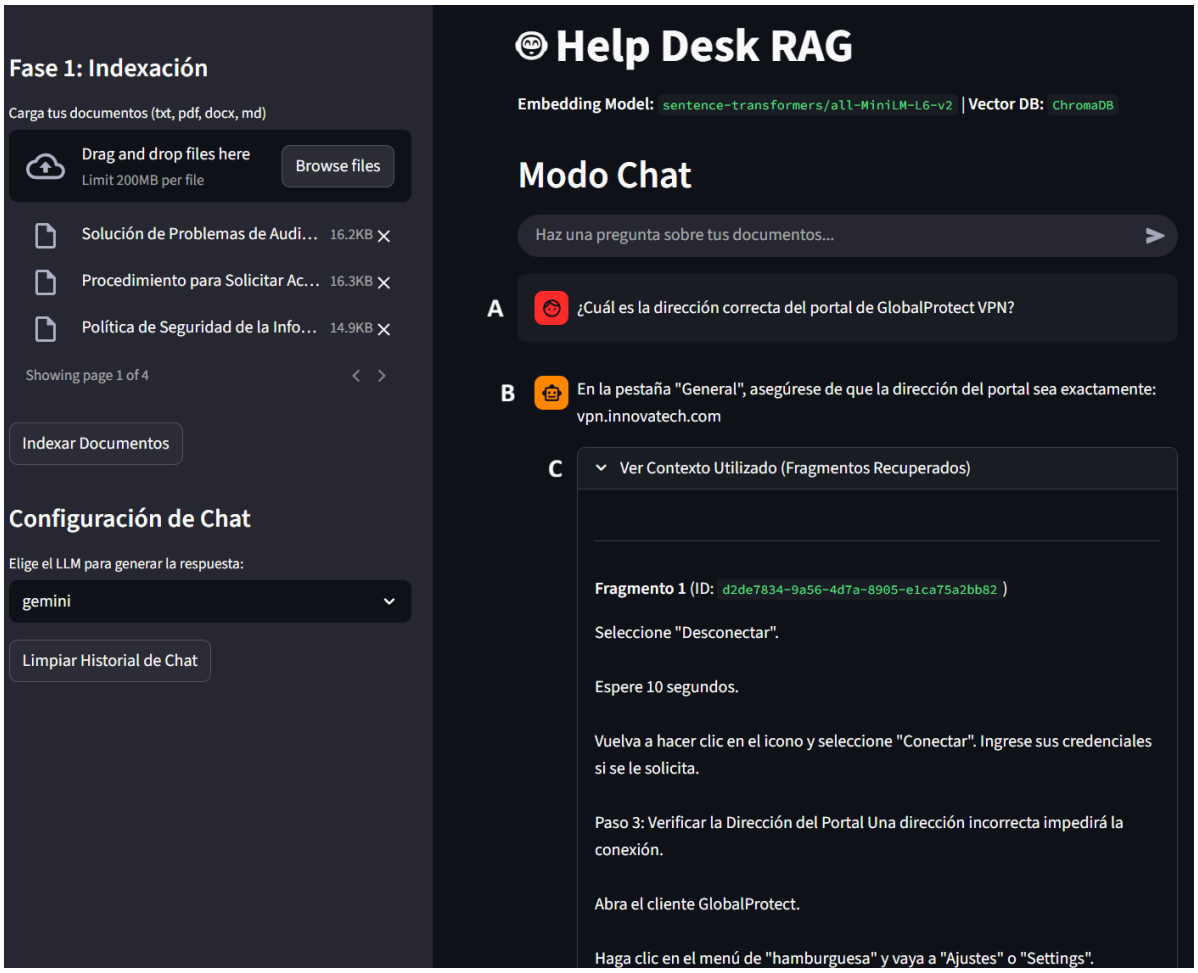
Interfaz de ingesta de documentos del prototipo



La Figura 7 demuestra la funcionalidad principal del chatbot. Se presenta una interacción real donde el sistema automatiza la resolución de una consulta. Un usuario realiza una consulta (A) y el sistema, ejecutando los pipelines de Recuperación (Fase 2) y Generación (Fase 3), devuelve una respuesta sintetizada (B). Fundamentalmente, el sistema expone el contexto recuperado (C), demostrando transparencia y validando el núcleo del mecanismo RAG. Esta funcionalidad cumple con la implementación del prototipo funcional del chatbot.

Figura 7

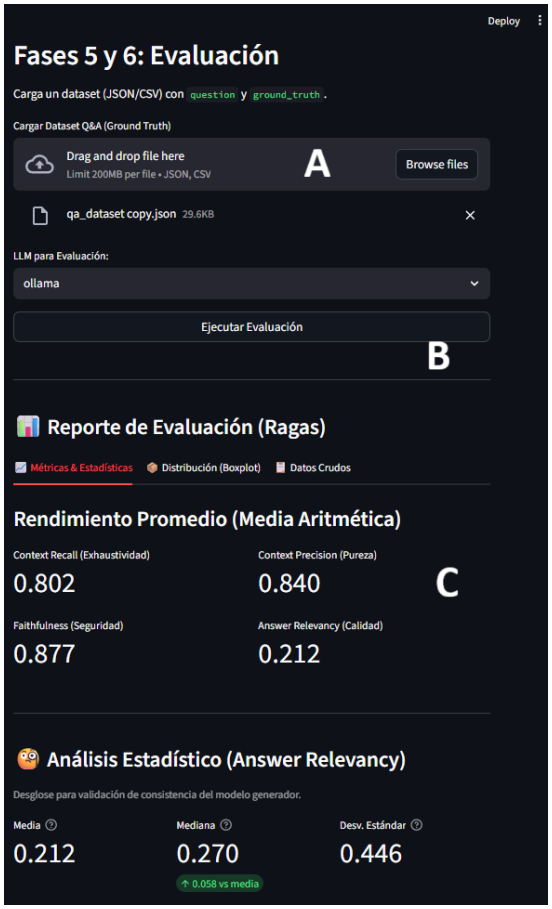
Demostración del prototipo funcional de chat (RAG)



Finalmente, para validar el sistema como una herramienta de investigación completa, la Figura 8 demuestra el flujo de evaluación en la interfaz. Esta funcionalidad es la implementación del Pipeline de Evaluación (Fase 4) y sirve como la herramienta para ejecutar el experimento cuantitativo y cumplir con el OE3. El panel permite al administrador cargar el *Dataset* Q&A (A) e iniciar el proceso de validación (B). El sistema ejecuta automáticamente el pipeline RAG para cada pregunta del *dataset* y presenta el Reporte de Desempeño (C) directamente en la interfaz. Esta implementación garantiza la transparencia y reproducibilidad de los resultados que se analizan a continuación.

Figura 8

Interfaz del pipeline de evaluación cuantitativa (RAGAS)



Esta sección presenta los resultados cuantitativos obtenidos tras ejecutar el Pipeline de Evaluación (Fase 4) sobre el *dataset* de validación (50 preguntas y respuestas). El objetivo es medir el rendimiento del sistema RAG. La ejecución del protocolo de evaluación arrojó los resultados consolidados que se presentan en la Tabla 9. Los valores representan la puntuación promedio (en una escala de 0 a 1) para cada una de las métricas clave.

Tabla 9

Resultados cuantitativos iniciales del prototipo (Preajuste)

Métrica	Componente	Puntuación Promedio
Context Precision	Recuperación	0.802
Context Recall	Recuperación	0.840
Faithfulness	Generación	0.877
Answer Relevancy	Generación	0.212

La primera ejecución de la evaluación arrojó resultados contradictorios. Mientras que métricas como *faithfulness* y *context_recall* alcanzaron valores altos, la métrica *answer_relevancy* presentó un resultado anómalo de 0.212. Este dato contradecía la inspección cualitativa manual, donde las respuestas del chatbot se percibían coherentes y pertinentes. El análisis técnico del código fuente del framework RAGAS permitió identificar un sesgo lingüístico en la evaluación. La métrica *Answer Relevancy* incluye un mecanismo de penalización estricta (*kill switch*) definido como:

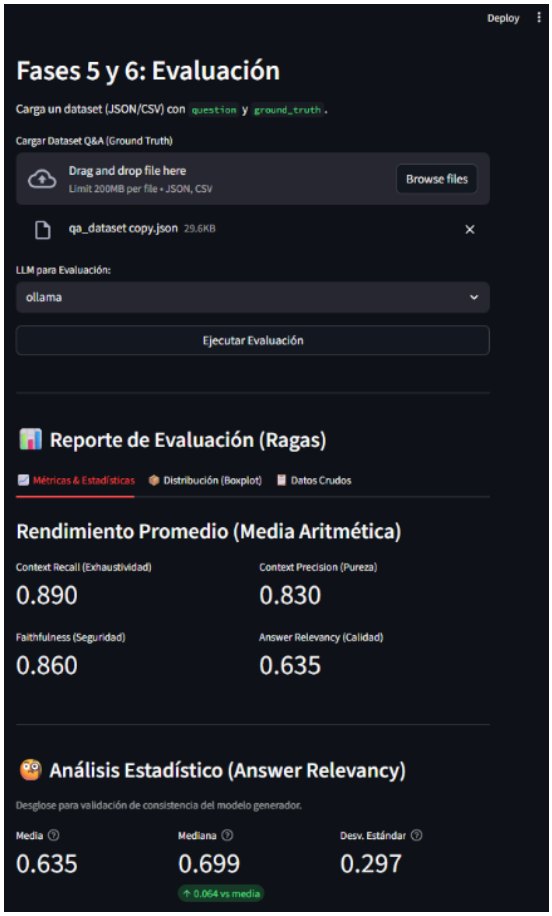
$$\text{score} = \text{valor\_inicial} \times \text{int}(\text{not all\_noncommittal})$$

Este mecanismo anula la puntuación (multiplicándola por cero) si el modelo evaluador considera que la respuesta es evasiva o no comprometida (*noncommittal*). Se validó la hipótesis de que el modelo gemma3:4b, al recibir el *prompt* de evaluación original en inglés para juzgar respuestas en español, clasificaba erróneamente respuestas válidas como evasivas, activando incorrectamente la penalización.

Para corregir este sesgo, se realizó la localización del *prompt* de evaluación al idioma español. Tras este ajuste metodológico, se repitió el experimento, obteniendo métricas que reflejan con fidelidad el desempeño real del sistema, como se muestra en la Figura 9.

Figura 9

Evaluación del chatbot postajuste



Tras corregir el instrumento de evaluación, se obtuvieron los resultados finales. Para ofrecer una lectura más completa que vaya más allá del promedio simple, en la Tabla 10 se presentan medidas que describen el comportamiento de los datos, como la media, la mediana y la desviación estándar.

Tabla 10

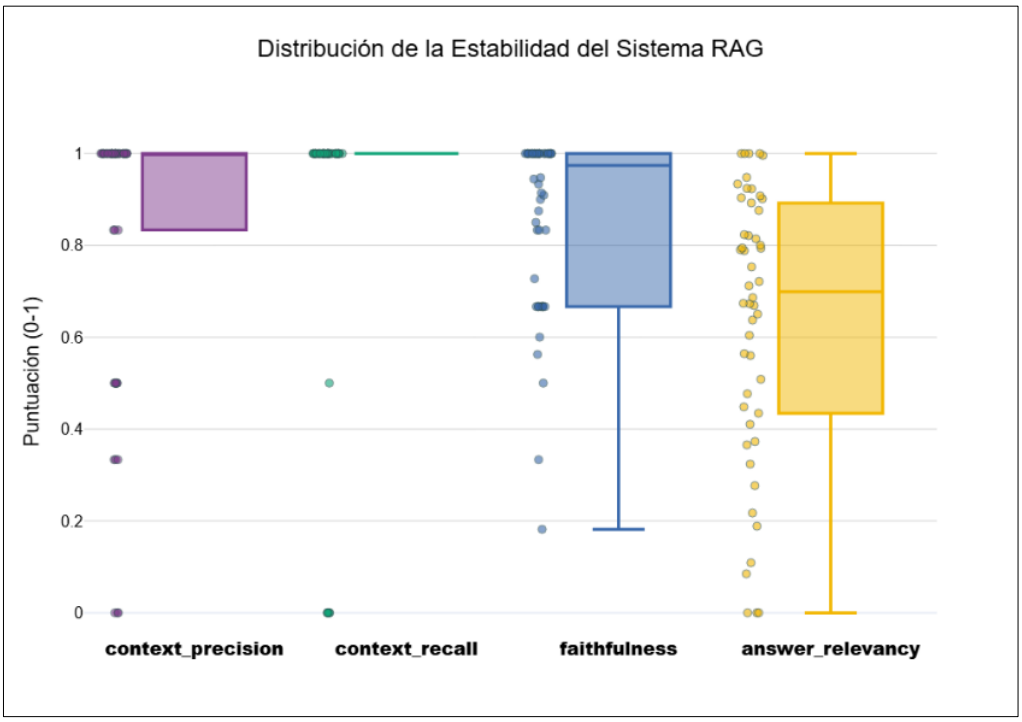
Resultados cuantitativos finales del prototipo

Métrica	Media	Mediana	Desv. Estándar	Interpretación Estadística
Context Recall	0.890	1.000	0.308	Dispersión por polarización. Comportamiento "todo o nada" (éxito total o fallo de recuperación).
Context Precision	0.830	1.000	0.302	Dispersión moderada. Afectada por casos puntuales de ruido documental.
Faithfulness	0.860	0.974	0.195	Alta estabilidad. La métrica con menor varianza; el sistema es consistentemente fiel.
Answer Relevancy	0.635	0.699	0.297	Dispersión estocástica. Alta variabilidad distribuida a lo largo de todo el espectro (0-1).

La métrica *Answer Relevancy* alcanzó un promedio final de 0.635, un valor que refleja un desempeño general sólido. Sin embargo, es importante destacar que la mediana (0.699) supera a la media. Estadísticamente, esta diferencia indica una distribución con sesgo negativo, lo que significa que el rendimiento típico del sistema es mejor que el promedio reportado. Esto indica que el resultado global se ve afectado por un pequeño conjunto de casos extremos (cercanos a 0) en los que el modelo no logró inferir la respuesta. Para ilustrar la estabilidad del sistema y la dispersión de los puntajes, la Figura 10 muestra la distribución mediante diagramas de caja obtenido del módulo de evaluación.

Figura 10

Distribución de puntajes por métrica evaluada



El análisis gráfico confirma tres hallazgos clave:

- Robustez en Recuperación: La caja de *Context Recall* está colapsada en el valor 1.0, confirmando que la arquitectura de búsqueda vectorial es altamente consistente.
- Seguridad en Generación: La métrica *Faithfulness* muestra que, en la inmensa mayoría de los casos, el sistema no alucina (mediana en 1.0), siendo los errores casos aislados.
- Volatilidad del LLM: La caja de *Answer Relevancy* presenta la mayor dispersión. Esto identifica al modelo generador local como el componente más sensible a la complejidad de la consulta.

4.3 Discusión de los resultados y limitaciones del estudio

Esta sección presenta el análisis e interpretación de los resultados con el propósito de validar los objetivos planteados. La discusión aborda dos hallazgos principales. el desempeño real del prototipo RAG en la automatización de consultas y la identificación de un artefacto metodológico crítico en el framework de evaluación, cuya corrección resultó indispensable para garantizar la validez de los resultados obtenidos.

El objetivo general de este trabajo fue implementar un prototipo RAG capaz de automatizar consultas frecuentes para reducir la carga operativa de una mesa de ayuda. Se partió de la hipótesis de que un sistema basado en herramientas de código abierto podría gestionar un volumen significativo de dichas consultas con una precisión aceptable.

Los resultados cuantitativos respaldan esta hipótesis, aunque con matices importantes derivados de la arquitectura local. La métrica clave, *answer_relevancy*, alcanzó un promedio final de 0.635 sobre el conjunto de 50 muestras de prueba. Dado que la métrica de relevancia es un valor continuo (0-1), resulta necesario establecer un criterio de corte para estimar la utilidad práctica del sistema en un entorno real. Para este análisis, se establece un umbral de referencia de (0.70). Este criterio toma como base los estándares de alineación semántica validados en arquitecturas recientes como TweakLLM, donde se demuestra experimentalmente que una similitud coseno de 0.70 es el límite inferior para garantizar una correspondencia temática fuerte en espacios vectoriales. Se extrapola este valor para la interpretación de la métrica *Answer Relevancy*, considerando que puntuaciones muy inferiores a este umbral evidencian una desconexión semántica significativa entre la intención de la consulta y la respuesta generada por el sistema (Cheema et al., 2025).

Al aplicar este criterio sobre la distribución de los resultados, se observa que, pese a tener un promedio inferior, la mediana del sistema (0.699) se sitúa en el límite de este umbral. Esto implica que el sistema logra resolver satisfactoriamente cerca del 50% de las incidencias con un nivel de calidad alto, validando su utilidad como herramienta de soporte de primer nivel, aunque evidenciando la necesidad de supervisión en casos complejos.

Este resultado principal se ve reforzado por el sólido desempeño en las métricas auxiliares del OE3:

1. Robustez del recuperador (*context_recall*: 0.890). Este valor evidencia que el sistema de recuperación es altamente eficaz, identificando la información correcta en el 89% de los casos. El análisis de distribución mostró un "efecto techo" con baja dispersión, confirmando que la estrategia de *chunking* y *embeddings* es sólida y no representa un cuello de botella.
2. Eficiencia del filtrado (*context_precision*: 0.830). Este resultado, optimizado por el módulo de *reranking*, indica que el 83% del contexto enviado al LLM fue útil y relevante. Demuestra la eficiencia del pipeline para filtrar el ruido y enviar al generador una base de conocimiento limpia.
3. Seguridad de la respuesta (*faithfulness*: 0.860). Esta métrica es crucial para la viabilidad corporativa. Demuestra que el sistema es robusto frente a las alucinaciones, adhiriéndose estrictamente a los hechos recuperados. Su comportamiento bimodal sugiere que el sistema es generalmente seguro, fallando solo en casos aislados de ruptura de contexto.

El análisis conjunto revela una dicotomía en la arquitectura: mientras que el componente de recuperación opera con niveles de excelencia (0.89), el desafío reside en la capacidad de generación. La brecha entre la disponibilidad de la información y la relevancia final (0.635) sugiere que la limitación no es de conocimiento, sino de la capacidad de síntesis

del modelo gemma3:4b. En conjunto, los resultados permiten concluir que los objetivos específicos 1 (diseñar), 2 (implementar) y 3 (evaluar) fueron alcanzados satisfactoriamente.

Para comprender mejor lo que reflejan las métricas globales, se realizó un análisis cualitativo detallado sobre casos concretos en los que la evaluación automática no coincidía con la calidad percibida por un revisor humano. Este examen permitió identificar varios patrones técnicos que explican por qué el modelo se comporta como lo hace, más allá de los promedios estadísticos.

El primero de estos patrones corresponde al fenómeno del conocimiento paramétrico dominante. En consultas como "¿Qué debo hacer si tengo problemas para configurar mi correo en el móvil?", el sistema obtuvo un *Context Recall* de 0.000, lo que indica que no recuperó el documento relevante de la base de conocimiento. Aun así, generó una respuesta correcta y con alta fidelidad (0.833). Esto ocurre porque el modelo gemma3:4b, al no recibir contexto externo, recurre a su conocimiento interno adquirido durante el preentrenamiento y produce instrucciones genéricas pero funcionales. Aunque el resultado es útil para el usuario, desde la perspectiva de una arquitectura RAG constituye un falso positivo, lo que refuerza la importancia de realizar auditorías periódicas para detectar lagunas en la documentación disponible.

Un segundo patrón tiene que ver con la penalización por verbosidad y cortesía. En preguntas breves y directas, como "Mi micrófono no funciona en Teams, ¿qué es lo primero que debo revisar?", el modelo tendió a generar respuestas extensas, estructuradas y muy corteses. Esta diferencia en la longitud y el estilo introduce "relleno conversacional" que reduce la similitud semántica vectorial y hace que la métrica de *Answer Relevancy* (0.277) subestime la calidad real de la respuesta, que en contenido era completamente fiel (1.000) y correcta.

Finalmente, se identificaron casos de lo que podría describirse como alucinación elocuente. El modelo generó respuestas coherentes, detalladas y aparentemente bien fundamentadas (*Answer Relevancy*: 0.821), pero sin respaldo factual en el contexto recuperado (*Faithfulness*: 0.182). Esto ocurrió especialmente en consultas sobre direcciones o configuraciones de red, por ejemplo, impresoras, donde el modelo rellenó vacíos de evidencia con supuestos plausibles. Este comportamiento muestra que, cuando el contexto es insuficiente, los modelos generativos pueden priorizar la forma por encima del fondo, lo que subraya la importancia de la métrica de fidelidad como mecanismo central de control de calidad y seguridad del sistema.

Un aporte significativo de este estudio fue la detección de un artefacto metodológico en la herramienta RAGAS. La evaluación inicial arrojó un *answer_relevancy* anómalo de 0.212, lo cual contradecía la inspección cualitativa. El análisis detallado reveló que el fallo no era del prototipo, sino de la métrica, afectada por tres factores concurrentes:

1. Lógica de penalización estricta: La métrica activa un mecanismo de anulación (*kill switch*) ante respuestas clasificadas como evasivas.
2. Sesgo lingüístico: El *prompt* original en inglés introdujo un sesgo al evaluar respuestas en español.
3. Limitación del juez (LLM): El modelo evaluador local mostró dificultades para realizar juicios translingüísticos, disparando falsos negativos.

La localización del *prompt* al español se convirtió, por tanto, en una intervención metodológica necesaria e indispensable para mejorar la herramienta de evaluación y poder cumplir con el OE3.

El rendimiento de 0.635 obtenido resulta notable si se consideran las limitaciones del estudio, las cuales explican el margen de error restante y marcan las posibles líneas de trabajo futuro:

1. Limitación del LLM: La principal restricción fue la elección estratégica de un LLM local. Se optó por gemma3:4b (4B parámetros) para garantizar la protección de los datos, evitar costos y eludir las restricciones de API. Este modelo, aunque funcional, tiene una capacidad de razonamiento y síntesis inferior a la de modelos de API a gran escala. La alta desviación estándar en *Answer Relevancy* (0.297) es atribuible directamente a la volatilidad estocástica del modelo.
2. Limitación metodológica en la evaluación: Esta limitación se debe a una restricción física de la infraestructura de hardware empleada. Ejecutar Ragas con el valor estándar de (n=3) habría triplicado la carga computacional y el tiempo de inferencia, superando la memoria necesaria para mantener el modelo cuantizado funcionando de forma estable. Por ello, se optó por una ejecución secuencial (n=1) como compromiso entre la viabilidad técnica del experimento local y la robustez estadística, asumiendo una mayor sensibilidad a la estocasticidad del modelo.
3. Limitación de optimización del pipeline: El prototipo utiliza componentes genéricos (*embeddings* y *reranker*); el uso de modelos especializados podría mejorar la precisión del recuperador (OE1) y, en consecuencia, facilitar la tarea del generador.
4. Ingeniería de *prompts*: El *prompt* del sistema, no fue sometido a un proceso iterativo de optimización avanzada. Un refinamiento en la ingeniería de *prompts* podría mejorar la coherencia y precisión de las respuestas.

A pesar de estas limitaciones, este estudio realiza aportes significativos en tres dimensiones:

- Práctica: Valida que un sistema RAG 100% *open source* puede alcanzar una eficacia de recuperación cercana al 90% y una relevancia operativa (mediana) de 0.70, suficiente para automatizar el primer nivel de soporte.

- Metodológica: Identifica y corrige sesgos críticos en la evaluación automatizada (RAGAS) para el idioma español, proporcionando una advertencia crucial para futuras investigaciones en contextos no angloparlantes.
- Académica: Establece un *benchmark* realista para sistemas RAG bajo restricciones de hardware, contrastando con estudios que dependen de recursos computacionales masivos. Para fomentar la replicabilidad, el código fuente y el pipeline de evaluación corregido se publican como un repositorio de acceso abierto en https://github.com/vladbin/tesis-rag-helpdesk.

Conclusiones

Se confirma la hipótesis de que es técnicamente viable implementar un sistema de automatización de consultas utilizando un *stack* 100% local y de código abierto (LlamaIndex, ChromaDB, Gemma3:4b). El prototipo alcanzó una mediana de relevancia operativa de 0.699, lo que indica que, en su comportamiento típico, el sistema es capaz de articular respuestas coherentes y útiles para el usuario final. Este resultado valida su potencial para reducir la carga operativa en un primer nivel de soporte sin comprometer la privacidad de los datos, demostrando que la soberanía tecnológica es alcanzable con recursos limitados.

El análisis desagregado de las métricas revela una dicotomía en el comportamiento estadístico de la arquitectura. Por un lado, el subsistema de recuperación muestra un desempeño robusto (*Context Recall*: 0.890; *Context Precision*: 0.830) con desviación estándar $\sigma \approx 0.308$, evidenciando un comportamiento polarizado: es eficaz en la inmensa mayoría de los casos, con fallos atribuibles a vacíos puntuales de información. En contraste, el subsistema generativo (*Answer Relevancy*: $\mu = 0.635;\ \sigma \approx 0.297$) se identifica como cuello de botella, mostrando variabilidad estocástica derivada de la limitación de un modelo de 4 billones de parámetros, cuya capacidad de inferencia fluctúa según la complejidad de la consulta.

Se demostró que los *frameworks* de evaluación estándar (RAGAS), diseñados predominantemente para el idioma inglés, introducen sesgos críticos al aplicarse en contextos hispanohablantes. La puntuación inicial anómala de 0.212 en relevancia no reflejaba un fallo del prototipo, sino un artefacto métrico provocado por la interpretación errónea de respuestas en español como evasivas. La corrección metodológica mediante la localización de *prompts* resultó indispensable, estableciendo que la validación lingüística del instrumento de evaluación es un prerrequisito científico para estudios de esta naturaleza.

Finalmente, la Tabla 11 sitúa el aporte de esta investigación frente al estado del arte. El análisis comparativo evidencia que, si bien las plataformas SaaS ofrecen una capacidad de razonamiento superior, el prototipo desarrollado ocupa un nicho estratégico insustituible: garantiza la soberanía total de los datos eliminando los costos variables por token y licenciamiento de software, lo que permite escalar el volumen de consultas sin incrementar el presupuesto operativo directo.

Tabla 11

Análisis Comparativo: Prototipo Local vs. Soluciones de Mercado

Criterio	**Prototipo Desarrollado (Tesis)**	**Soluciones SaaS (ej. OpenAI, Azure)**	**RAG Enterprise On-Premise**
Arquitectura	100% Local (*Open Source*)	Nube Propietaria (Caja Negra)	Servidores Dedicados

Criterio	Prototipo Desarrollado (Tesis)	Soluciones SaaS (ej. OpenAI, Azure)	RAG Enterprise On-Premise
Privacidad de Datos	Soberanía Total (No sale de la red)	Datos viajan a terceros	Alta (Requiere auditoría)
Costo Variable por Inferencia	Nulo (0 USD/token)	Variable (Pago por token/uso)	Alto (Licencias y Mantenimiento)
Hardware Requerido	Estación de Trabajo (GPU 8GB VRAM)	Ninguno (Solo acceso a Internet)	Clúster de GPUs
Dependencia	Independiente	*Vendor Lock-in*	*Vendor Lock-in*
Calidad de Respuesta	Operativa (Aprox. 70% éxito)	Estado del Arte (>95% éxito)	Alta (>90% éxito)
Caso de Uso Ideal	Datos Confidenciales / PyMEs	Prototipado rápido / Público general	Grandes Corporaciones

Recomendaciones

Dado que la recuperación ya es óptima, se recomienda priorizar la sustitución del modelo gemma3:4b por modelos locales de mayor densidad de parámetros, como Llama-3-8B o Mistral-7B. Se estima que incrementar la capacidad de razonamiento atacará directamente la varianza detectada en la métrica de relevancia (0.297), mejorando la síntesis de respuestas sin alterar la arquitectura de búsqueda.

La precisión actual es alta (0.830), sin embargo, se sugiere explorar técnicas de *Fine-Tuning* sobre el modelo de *Reranking* utilizando pares de preguntas reales de la organización. Esto permitirá al sistema discriminar con mayor agudeza entre documentos técnicos con alta similitud semántica, elevando la pureza del contexto entregado al LLM.

Se recomienda implementar un protocolo de auditoría documental para las consultas con *Context Recall* nulo. Dado que la desviación estándar de esta métrica (0.308) es atribuible a casos de fallo total (0.0), estas incidencias deben tratarse como "señales de vacío" para ingerir nueva documentación o ajustar los metadatos, convirtiendo fallos binarios en aciertos futuros.

Iniciar la implementación productiva en un entorno controlado (ej. consultas de políticas internas) manteniendo la opción de escalar a un agente humano. La mediana de relevancia (0.699) valida que el desempeño es suficiente para filtrar consultas repetitivas (Tier 1), pero el sistema aún requiere supervisión para incidencias críticas o de alta ambigüedad.

Se insta a no asumir la neutralidad de herramientas estándar como RAGAS. Es imperativo auditar y localizar (traducir) los *prompts* de evaluación antes de medir el desempeño en español para evitar artefactos métricos (falsos negativos) que invaliden los resultados científicos.

Evitar la evaluación basada en una única métrica. Se recomienda contrastar métricas de diagnóstico como *context_recall* frente a métricas de resultado como *answer_relevancy*. Esta diferencia es la herramienta clave para aislar cuellos de botella técnicos, diferenciando fallos de recuperación de fallos de razonamiento.

Fomentar la creación de líneas base de rendimiento centradas en arquitecturas de código abierto y hardware restringido. La investigación debería reducir la dependencia de APIs propietarias para ofrecer referencias realistas. Se recomienda utilizar el repositorio y el pipeline publicados en este estudio como base para construir un cuerpo de conocimiento sobre la viabilidad del RAG local en organizaciones con restricciones de privacidad.

Referencias

Abdulahi Jimale Said & Abdihakin Mohamud Ismail. (2025). Trends in natural language processing for text classification: A comprehensive survey. *International Journal of Science and Research Archive*, *14*(2), 1540-1547. https://doi.org/10.30574/ijsra.2025.14.2.0518

Alkaabi, H., Jasim, A. K., & Darroudi, A. (2025). From Static to Contextual: A Survey of Embedding Advances in NLP. *PERFECT: Journal of Smart Algorithms*, *2*(2), 57-66. https://doi.org/10.62671/perfect.v2i2.77

Anisuzzaman, D. M., Malins, J. G., Friedman, P. A., & Attia, Z. I. (2025). Fine-Tuning Large Language Models for Specialized Use Cases. *Mayo Clinic Proceedings: Digital Health*, *3*(1), 100184. https://doi.org/10.1016/j.mcpdig.2024.11.005

Berti, L., Giorgi, F., & Kasneci, G. (2025). *Emergent Abilities in Large Language Models: A Survey* (Versión 2). arXiv. https://doi.org/10.48550/ARXIV.2503.05788

Bhat, V., Cheerla, S. D., Mathew, J. R., Pathak, N., Liu, G., & Gao, J. (2024). Retrieval Augmented Generation (RAG) Based Restaurant Chatbot with AI Testability. *2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)*, 1-10. https://doi.org/10.1109/BigDataService62917.2024.00008

Chakraborty, A., Dahal, C., & Gupta, V. (2025). *Federated Retrieval-Augmented Generation: A Systematic Mapping Study* (No. arXiv:2505.18906). arXiv. https://doi.org/10.48550/arXiv.2505.18906

Cheema, M. T., Aamir, A., Muhammad, K. G., Bhatti, N. A., Qazi, I. A., & Qazi, Z. A. (2025). *TweakLLM: A Routing Architecture for Dynamic Tailoring of Cached Responses* (No. arXiv:2507.23674). arXiv. https://doi.org/10.48550/arXiv.2507.23674

Cheerla, C. (2025). *Advancing Retrieval-Augmented Generation for Structured Enterprise and Internal Data* (No. arXiv:2507.12425). arXiv. https://doi.org/10.48550/arXiv.2507.12425

Chen, Z., Wang, S., Xiao, T., Wang, Y., Chen, S., Cai, X., He, J., & Wang, J. (2025). Revisiting Scaling Laws for Language Models: The Role of Data Quality and Training Strategies. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 23881-23899. https://doi.org/10.18653/v1/2025.acl-long.1163

Chintha, D., & Konduru, N. V. (2025). *Survey of Tokenization Mechanisms in Multilingual Large Language Models with a Focus on Indian Languages*. *12*(4).

Colangelo, M. T., Meleti, M., Guizzardi, S., Calciolari, E., & Galli, C. (2025). *A Comparative Analysis of Sentence Transformer Models for Automated Journal Recommendation Using PubMed Metadata*. Biology and Life Sciences. https://doi.org/10.20944/preprints202501.1334.v1

Dam, S. K., Hong, C. S., Qiao, Y., & Zhang, C. (2024). *A Complete Survey on LLM-based AI Chatbots* (No. arXiv:2406.16937). arXiv. https://doi.org/10.48550/arXiv.2406.16937

Ernst & Young. (2024). *Nuevos horizontes de la madurez digital en el Perú*. https://www.ey.com/content/dam/ey-unified-site/ey-com/es-pe/insights/technology/documents/ey-nuevos-horizontes-madurez-digital-en-peru-2024.pdf

Espinosa-Luna, B. H., Castillo-Oliva, J., Montañez-Díaz, B. A., & Mendoza-De-los-Santos, A. (2023). Implementación de un chatbot basado en modelo de lenguaje de inteligencia artificial para responder preguntas frecuentes de estudiantes universitarios. *Revista Científica de Sistemas e Informática*, *3*(2), e570. https://doi.org/10.51252/rcsi.v3i2.570

Explodinggradients. (2025). *List of available metrics*. https://docs.ragas.io/en/latest/concepts/metrics/available_metrics/

Fan, X., Xiao, Q., Zhou, X., Pei, J., Sap, M., Lu, Z., & Shen, H. (2025). *User-Driven Value Alignment: Understanding Users' Perceptions and Strategies for Addressing Biased and Discriminatory Statements in AI Companions* (No. arXiv:2409.00862). arXiv. https://doi.org/10.48550/arXiv.2409.00862

Fichtl, A. M., Bohn, J., Kelber, J., Mosca, E., & Groh, G. (2025). *The End of Transformers? On Challenging Attention and the Rise of Sub-Quadratic Architectures* (No. arXiv:2510.05364). arXiv. https://doi.org/10.48550/arXiv.2510.05364

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). *Retrieval-Augmented Generation for Large Language Models: A Survey* (No. arXiv:2312.10997). arXiv. https://doi.org/10.48550/arXiv.2312.10997

Garzón-Quiroz, M., Del Campo-Saltos, G., & Loor-Ávila, B. (2025). Análisis sistemático sobre la eficiencia comunicativa entre chatbots basados en reglas y modelos de lenguaje natural. *Universitas*, *42*, 167-192. https://doi.org/10.17163/uni.n42.2025.07

Gill, G., Gupta, R., Lusson, D., Chandrashekar, A., & Nguyen, D. (2025). *From Search to Reasoning: A Five-Level RAG Capability Framework for Enterprise Data* (No. arXiv:2509.21324). arXiv. https://doi.org/10.48550/arXiv.2509.21324

Golder, S., Xu, D., O'Connor, K., Wang, Y., Batra, M., & Hernandez, G. G. (2025). Leveraging Natural Language Processing and Machine Learning Methods for Adverse Drug Event Detection in Electronic Health/Medical Records: A Scoping Review. *Drug Safety*, *48*(4), 321-337. https://doi.org/10.1007/s40264-024-01505-6

Han, S., Wang, M., Zhang, J., Li, D., & Duan, J. (2024). A Review of Large Language Models: Fundamental Architectures, Key Technological Evolutions, Interdisciplinary Technologies Integration, Optimization and Compression Techniques, Applications, and Challenges. *Electronics*, *13*(24), 5040. https://doi.org/10.3390/electronics13245040

Hasan, K. I., & Ibrahim, I. M. (2025). A Review of Natural Language Processing for Structured and Unstructured Data in Electronic Health Records. *Engineering and Technology Journal*, *10*(05). https://doi.org/10.47191/etj/v10i05.51

He, R., Cao, J., & Tan, T. (2025). Generative artificial intelligence: A historical perspective. *National Science Review*, *12*(5), nwaf050. https://doi.org/10.1093/nsr/nwaf050

Heigl, R. (2025). Generative artificial intelligence in creative contexts: A systematic review and future research agenda. *Management Review Quarterly*. https://doi.org/10.1007/s11301-025-00494-9

IBM. (2023). *New IBM study reveals how AI is changing work and what HR leaders should do about it*. https://www.ibm.com/think/insights/new-ibm-study-reveals-how-ai-is-changing-work-and-what-hr-leaders-should-do-about-it

IBM. (2025a). *What is LlamaIndex?* https://www.ibm.com/think/topics/llamaindex

IBM. (2025b). *What is LLM orchestration?* https://www.ibm.com/think/topics/llm-orchestration

INEI. (2023). *PRINCIPALES INDICADORES MACROECONÓMICOS*. https://m.inei.gob.pe/estadisticas/indice-tematico/economia/

IPSOS. (2024). *LA IA Y EL FUTURO DE CX*. https://www.ipsos.com/sites/default/files/AI%20and%20the%20Future%20of%20CX%20POV_v8%20ESP.pdf

Ivanti. (2025). *Employee Experience Management*. https://www.ivanti.com/resources/research-reports/2025-digital-employee-experience-report

James, A., Trovati, M., & Bolton, S. (2025). Retrieval-Augmented Generation to Generate Knowledge Assets and Creation of Action Drivers. *Applied Sciences*, *15*(11), 6247. https://doi.org/10.3390/app15116247

Jia, X., & Zhao, Z. (2025). *The Emergence of Social Science of Large Language Models* (Versión 1). arXiv. https://doi.org/10.48550/ARXIV.2509.24877

Knauss, E. (2021). *Constructive Master's Thesis Work in Industry: Guidelines for Applying Design Science Research* (No. arXiv:2012.04966). arXiv. https://doi.org/10.48550/arXiv.2012.04966

Kostikova, A., Wang, Z., Bajri, D., Pütz, O., Paaßen, B., & Eger, S. (2025). *LLLMs: A Data-Driven Survey of Evolving Research on Limitations of Large Language Models* (No. arXiv:2505.19240). arXiv. https://doi.org/10.48550/arXiv.2505.19240

Kowsher, M., Prottasha, N. J., Yu, C.-N., Garibay, O. O., & Yousefi, N. (2025). *Does Self-Attention Need Separate Weights in Transformers?* (No. arXiv:2412.00359). arXiv. https://doi.org/10.48550/arXiv.2412.00359

Lee, H.-C., Hung, K., Man, G. M.-T., Ho, R., & Leung, M. (2024). Development of an RAG-Based LLM Chatbot for Enhancing Technical Support Service. *TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON)*, 1080-1083. https://doi.org/10.1109/TENCON61640.2024.10902801

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks* (No. arXiv:2005.11401). arXiv. https://doi.org/10.48550/arXiv.2005.11401

Lin, X., & Li, L. (2025). *Implicit Bias in LLMs: A Survey*.

Mahajan, V. F. (2025). The Evolution of AI Support: How RAG is Transforming Customer Experience. *European Journal of Computer Science and Information Technology*, *13*(14), 115-126. https://doi.org/10.37745/ejcsit.2013/vol13n14115126

Marcel, M., & Aotearoa, G. H. (2025). Enhancing IT Service Desk for Hybrid Work: Insight from a TOE and TTF Case Study. *Journal of Information Systems and Informatics*, *7*(1), 848-869. https://doi.org/10.51519/journalisi.v7i1.971

MetricNet. (2021). *The ROI of Benchmarking | The Business Case for Benchmarking IT Service and Support*. https://www.metricnet.com/benchmarkingroi/

Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information*, *15*(9), 517. https://doi.org/10.3390/info15090517

Mostafa, A., Nahid, R. A., & Mulder, S. (2025). How Different Tokenization Algorithms Impact LLMs and Transformer Models for Binary Code Analysis. *Proceedings 2025 Workshop on Binary Analysis Research*. Workshop on Binary Analysis Research, San Diego, CA, USA. https://doi.org/10.14722/bar.2025.23013

Ordóñez-Camacho, D., Melgarejo-Heredia, R., Abbasi, M., & González-Solis, L. (2024). Aurel_AI: Automating an Institutional Help Desk Using an LLM Chatbot. *Journal of Systemics, Cybernetics and Informatics*, *22*(5), 77-87. https://doi.org/10.54808/JSCI.22.05.77

Purewal Martinez, B., & Sobero Rodriguez, F. Y. (2025). Chatbot basado en un Modelo Grande de Lenguaje para la atención al cliente. *Revista Científica: BIOTECH AND ENGINEERING*, *5*(1). https://doi.org/10.52248/eb.Vol5Iss1.196

Rick, V. B., Brandl, C., Mertens, A., & Nitsch, V. (2024). Work interruptions of office workers: The influence of the complexity of primary work tasks on the perception of interruptions. *Work*, *77*(1), 185-196. https://doi.org/10.3233/WOR-220684

S Kamalov, D Absalamova, G Absalamova, J Kamalova, F Tengelova, & M Makhamedova. (2025). *SEMANTIC SEARCH THROUGH VECTOR STORES: SIGNIFICANCE IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS OF NLP MODELS*. *3*(3).

Salinas Santiago, J., García Gutiérrez, W. F., Ordoñez Reyes, A. B., & Mendoza De Los Santos, A. C. (2024). Implementación de un chatbot inteligente en la gestión de las mesas de ayuda. *Revista Científica: BIOTECH AND ENGINEERING*, *4*(1). https://doi.org/10.52248/eb.Vol4Iss1.103

Samira Rabinataj, Seyedeh. (2025). *Analyzing the Evolution of Social Concepts Using Temporal Embedding Models*.

Sanugommula, H. (2024). Optimizing Service Desk Operations: Enhancing Customer Support in the Digital Age. *Journal of Mathematical & Computer Applications*, 1-3. https://doi.org/10.47363/JMCA/2024(3)E150

Savage, C. H., Kanhere, A., Parekh, V., Langlotz, C. P., Joshi, A., Huang, H., & Doo, F. X. (2025). Open-Source Large Language Models in Radiology: A Review and Tutorial for Practical Research and Clinical Deployment. *Radiology*, *314*(1), e241073. https://doi.org/10.1148/radiol.241073

Sengul, C., Neykova, R., & Destefanis, G. (2024). Software engineering education in the era of conversational AI: Current trends and future directions. *Frontiers in Artificial Intelligence*, *7*, 1436350. https://doi.org/10.3389/frai.2024.1436350

Snowflake Inc. (2025). *Streamlit documentation*. https://docs.streamlit.io/

Sosa Erazo. (2024). *DISEÑO DE UN SISTEMA DE ASISTENCIA VIRTUAL BASADO EN INTELIGENCIA ARTIFICAL (IA) EN LA MESA DE SERVICIOS DE TECNOLOGIAS DE LA INFORMACIÓN DE LA EMPRESA DATAFAST S.A.* PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR.

Swacha, J., & Gracel, M. (2025). Retrieval-Augmented Generation (RAG) Chatbots for Education: A Survey of Applications. *Applied Sciences*, *15*(8), 4234. https://doi.org/10.3390/app15084234

Vanishree, K & Ananya, G. (2024). RAG based Chatbot using LLMs. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, *08*(06), 1-5. https://doi.org/10.55041/IJSREM35600

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (No. arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

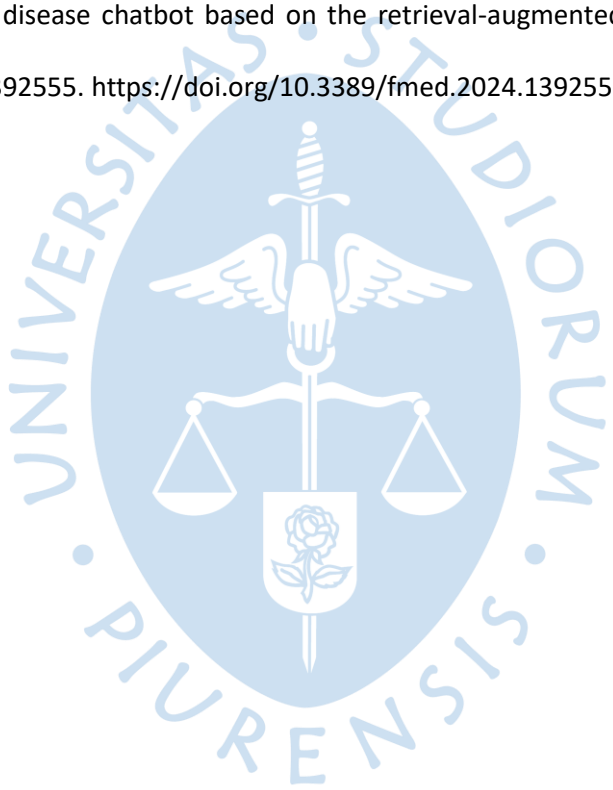
Wang, H., Wang, L., Du, Y., Chen, L., Zhou, J., Wang, Y., & Wong, K.-F. (2025). *A Survey of the Evolution of Language Model-Based Dialogue Systems: Data, Task and Models* (No. arXiv:2311.16789). arXiv. https://doi.org/10.48550/arXiv.2311.16789

Zhang, Q., Fang, C., Gu, S., Shang, Y., Chen, Z., & Xiao, L. (2025). *Large Language Models for Unit Testing: A Systematic Literature Review* (No. arXiv:2506.15227). arXiv. https://doi.org/10.48550/arXiv.2506.15227

Zhang, S., Fan, R., Liu, Y., Chen, S., Liu, Q., & Zeng, W. (2023). Applications of transformer-based language models in bioinformatics: A survey. *Bioinformatics Advances*, *3*(1), vbad001. https://doi.org/10.1093/bioadv/vbad001

Zhao, D. (2024). *FRAG: Toward Federated Vector Database Management for Collaborative and Secure Retrieval-Augmented Generation* (No. arXiv:2410.13272). arXiv. https://doi.org/10.48550/arXiv.2410.13272

Zhou, Q., Liu, C., Duan, Y., Sun, K., Li, Y., Kan, H., Gu, Z., Shu, J., & Hu, J. (2024). GastroBot: A Chinese gastrointestinal disease chatbot based on the retrieval-augmented generation. *Frontiers in Medicine*, *11*, 1392555. https://doi.org/10.3389/fmed.2024.1392555



Apéndice

Apéndice A. Tabla 12 Resultados de evaluación

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
1	¿Cuál es la aplicación recomendada para configurar el correo de Innovatech en Android?	De acuerdo a la documentación disponible, la aplicación recomendada para configurar el correo de Innovatech en Android es Microsoft Outlook. Se recomienda usarla sobre la app de Gmail. Para Android, se puede utilizar la app de Gmail o Microsoft Outlook.	1.000	0.667	1.000	1.000
5	¿Qué tipo de cuenta debo seleccionar en la app de Gmail para añadir mi correo de Innovatech?	Para añadir tu correo de Innovatech utilizando la app de Gmail, debes seleccionar "Exchange y Office 365". Después de seleccionar esta opción, se te pedirá que ingreses tu dirección de correo completa (nombre.apellido@innovatech.com) y procederás con el inicio de sesión, siguiendo los pasos indicados en la guía. Si tienes algún problema durante la configuración, te recomiendo crear un ticket en http://support.innovatech.com en la categoría "Dispositivos Móviles".	1.000	1.000	0.604	1.000
6	¿Cuál es el primer paso recomendado si no me puedo conectar a la VPN?	El primer paso para solucionar problemas de conexión a la VPN es verificar la conexión a Internet básica. Por favor, abra un navegador web e intente acceder a un sitio público como google.com o bbc.com. Si no puede navegar, el problema reside en su red local (Wi-Fi o cable). Reinicie su router o contacte a su proveedor de internet. La VPN no puede funcionar sin una conexión a internet estable.	1.000	1.000	0.366	1.000
7	¿Cuál es la dirección correcta del portal de GlobalProtect VPN?	La dirección correcta del portal de GlobalProtect VPN es vpn.innovatech.com. Por favor, asegúrese de que esta dirección esté configurada correctamente en su cliente GlobalProtect para asegurar una conexión exitosa.	1.000	0.667	1.000	1.000
8	¿Qué información debo incluir en un ticket de soporte si mi VPN sigue fallando después de seguir la guía?	Entiendo que tu VPN sigue presentando problemas a pesar de haber seguido los pasos de la guía. Para ayudarte a acelerar la resolución, te recomiendo incluir la siguiente información en tu ticket de soporte: 1. **Título del Ticket:** Sea descriptivo. Ej: "Problema de Conexión VPN" 2. **En el cuerpo del ticket, incluye:** * Una captura de pantalla del mensaje de error exacto que muestra el cliente GlobalProtect. * Una breve descripción de los pasos de esta guía que ya ha intentado. * Su ubicación física (ej: "Trabajando desde casa en Madrid", "En la oficina de Londres"). * ¿El problema es constante o intermitente? Para crear el ticket, por favor, contacta a la mesa de ayuda a través de: * **Portal de Soporte:** http://support.innovatech.com (Categoría: "Red y Conectividad") * **Teléfono (para casos urgentes):** Extensión 5555 (Opción 2). Si tienes alguna otra pregunta, no dudes en consultarme.	1.000	0.900	0.373	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
2	¿Qué debo hacer si tengo problemas para configurar mi correo en el móvil?	Entiendo que está teniendo problemas para configurar su correo electrónico de Innovatech Solutions en su teléfono móvil. Aquí le indico los pasos que puede seguir, basados en la documentación disponible: 1. **Verifique los requisitos de seguridad:** Asegúrese de que su dispositivo móvil tenga configurado un código de acceso, PIN, patrón o biometría (huella dactilar, reconocimiento facial). 2. **Siga las instrucciones de configuración para iOS (iPhone/iPad):** * Vaya a "Ajustes" > "Mail" > "Cuentas". * Seleccione "Añadir cuenta". * Elija "Microsoft Exchange". * Ingrese su dirección de correo completa. * Toque "Siguiente". * Seleccione "Iniciar sesión". * Apruebe el inicio de sesión a través de su método de autenticación multifactor (MFA). * Seleccione los servicios que desea sincronizar (Mail, Contactos, Calendarios, Recordatorios). * Toque "Guardar". 3. **Si aún tiene problemas, complete el desafío de CAPTCHA** (las letras y números distorsionados) para demostrar que no es un robot. 4. **Si los pasos anteriores no funcionan, contacte a la Mesa de Ayuda de TI:** * **Portal de Soporte:** http://support.innovatech.com (Crear un ticket en la categoría "Cuentas y Accesos"). * **Teléfono:** Extensión interna 5555 (Opción 1). Recuerde que el departamento de TI nunca le pedirá su contraseña por correo electrónico o por teléfono. Si recibe una solicitud de este tipo, repórtela inmediatamente a security@innovatech.com.	0.333	0.833	0.217	0.000
11	Mi micrófono no funciona en Teams, ¿qué es lo primero que debo revisar?	Para solucionar problemas de audio en Microsoft Teams, primero debe verificar lo siguiente: 1. **Que sus dispositivos estén conectados correctamente:** Asegúrese de que sus auriculares, micrófono externo o cámara web estén correctamente conectados a su ordenador (USB o Bluetooth). 2. **Que el micrófono no esté en silencio:** Verifique el botón de silencio físico en sus auriculares o micrófono. 3. **Que la configuración de dispositivos dentro de Teams esté correcta:** Antes de unirse a una reunión, haga clic en el icono de engranaje (Configuración de dispositivos) en la pantalla de vista previa. Si ya está en una reunión, haga clic en los tres puntos (...) en la parte superior y seleccione "Configuración" -> "Configuración de dispositivos". Dentro de la configuración, revise que el "Dispositivo de audio" seleccionado coincida con su dispositivo principal (ej: "Logitech USB Headset", "Jabra Evolve 75"). Si estos pasos no resuelven el problema, puede intentar actualizar los controladores de sus dispositivos.	0.500	1.000	0.277	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
3	Al configurar mi correo de Innovatech en el móvil, ¿qué políticas de seguridad estoy aceptando?	Claro, puedo ayudarte con eso. Al configurar su cuenta de correo electrónico de Innovatech Solutions en un dispositivo móvil, usted debe cumplir con las siguientes políticas de seguridad: 1. **Código de acceso/Autenticación biométrica:** El dispositivo debe tener configurado un código de acceso, PIN, patrón o biometría (huella dactilar, reconocimiento facial). 2. **Borrado Remoto:** El departamento de TI tiene la capacidad de borrar remotamente los datos corporativos del dispositivo en caso de pérdida o robo. 3. **Autenticación de Dos Factores (2FA):** Se requiere el uso de autenticación de dos factores (2FA) en todas las cuentas compatibles. 4. **Requisitos de la Contraseña:** * Mínimo 12 caracteres. * Debe contener al menos una letra mayúscula (A-Z), una letra minúscula (a-z), un número (0-9) y un símbolo (!, @, #, $, etc.). * No puede ser una de sus últimas 5 contraseñas utilizadas. Además, al completar el proceso de recuperación de contraseña, usted puede seleccionar uno de los siguientes métodos de verificación de identidad: * **Opción A: Aplicación de Autenticación:** Usar la aplicación Microsoft Authenticator (o similar) y aprobar la notificación de inicio de sesión. * **Opción B: Código por SMS:** Recibir un código de 6 dígitos en el número de teléfono registrado e ingresar el código en el portal. * **Opción C: Preguntas de Seguridad:** Responder correctamente a las preguntas de seguridad configuradas. ¿Hay algo más que pueda aclarar o en lo que pueda ayudarle?	1.000	0.909	0.800	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
4	¿Cómo configuro mi correo Innovatech en un iPhone?	Claro, con gusto te ayudaré a configurar tu correo Innovatech en un iPhone. Aquí están los pasos a seguir: 1. Ve a "Ajustes" > "Mail" > "Cuentas". 2. Seleccione "Añadir cuenta". 3. En la lista de proveedores, elija "Microsoft Exchange". 4. En el campo "Email", ingrese su dirección de correo completa (ej: nombre.apellido@innovatech.com). 5. Puede darle una descripción opcional (ej: "Innovatech Trabajo"). 6. Toque "Siguiente". Aparecerá una ventana emergente. Seleccione "Iniciar sesión". 7. Será redirigido a la página de inicio de sesión de Innovatech. Ingrese su contraseña y toque "Iniciar sesión". 8. Apruebe el inicio de sesión a través de su método de autenticación multifactor (MFA). 9. En la siguiente pantalla, elija los servicios que desea sincronizar (Mail, Contactos, Calendarios, Recordatorios). Se recomienda dejarlos todos activados. 10. Toque "Guardar". Su cuenta ya está configurada. Si tienes algún problema, puedes crear un ticket en http://support.innovatech.com (Categoría: "Software y Aplicaciones"), incluyendo: * El modelo de sus auriculares/cámara web. * Una captura de pantalla de su "Configuración de dispositivos" en Teams. * Los resultados de la llamada de prueba.	1.000	1.000	0.508	1.000
9	¿Para qué sistemas operativos es la guía de solución de problemas de la VPN?	Esta guía de solución de problemas de la VPN cubre los sistemas operativos Windows y macOS. El documento especifica que la guía está diseñada para empleados que utilizan el cliente GlobalProtect VPN en estos sistemas operativos para acceder a los recursos de la red interna de Innovatech Solutions.	1.000	1.000	0.448	1.000
12	¿Cómo puedo hacer una llamada de prueba en Teams para verificar mi audio?	Para verificar tu audio en Teams, puedes realizar una llamada de prueba. En el panel de configuración, haz clic en "Hacer una llamada de prueba". Teams llamará a un bot que te pedirá que grabe un mensaje corto y luego lo reproducirá. Esto confirma si tu micrófono funciona correctamente dentro de la aplicación.	0.833	1.000	0.435	1.000
10	¿Cómo reinicio el cliente GlobalProtect VPN?	Claro, puedo ayudarte con eso. Para reiniciar el cliente GlobalProtect VPN, por favor, siga estos pasos: 1. Busque el icono de GlobalProtect (un globo terráqueo) en su barra de tareas (Windows) o barra de menú (macOS). 2. Haga clic en el icono y luego en el menú de "hamburguesa" (tres líneas horizontales). 3. Seleccione "Desconectar". 4. Espere 10 segundos. 5. Vuelva a hacer clic en el icono y seleccione "Conectar". Ingrese sus credenciales si se le solicita. Si tiene problemas, recuerde incluir una captura de pantalla del mensaje de error exacto que muestra el cliente GlobalProtect, una breve descripción de los pasos que ya ha intentado y su ubicación física en su ticket de soporte.	1.000	1.000	1.000	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
15	¿Cómo actualizo los controladores (drivers) de mi cámara web en Windows?	Para actualizar los controladores de su cámara web en Windows, siga estos pasos: 1. Vaya a Inicio (en Windows) y escriba "Administrador de dispositivos". 2. Expanda las secciones "Controladoras de sonido y video y dispositivos de juego" y "Cámaras". 3. Haga clic derecho en su dispositivo (ej: su cámara web) y seleccione "Actualizar controlador".	1.000	1.000	0.901	1.000
13	Teams no detecta mi cámara. ¿Qué configuración del sistema operativo debo revisar en macOS?	Para solucionar problemas de detección de cámara en macOS con Teams, debe revisar los siguientes ajustes: 1. Vaya a Menú Apple > Ajustes del Sistema > Privacidad y seguridad. 2. Seleccione "Cámara" en la lista. 3. Asegúrese de que la casilla junto a Microsoft Teams esté marcada. Si el problema persiste, cree un ticket en http://support.innovatech.com (Categoría: "Software y Aplicaciones"), incluyendo: * El modelo de su cámara web. * Una captura de pantalla de su "Configuración de dispositivos" en Teams. * Los resultados de la llamada de prueba.	1.000	1.000	0.788	1.000
14	¿Cómo reviso qué micrófono está usando Teams durante una reunión?	Entiendo que quieres saber cómo puedes ver qué dispositivo de audio está seleccionando Microsoft Teams durante una reunión en curso. Para diagnosticar este problema, te sugiero los siguientes pasos: 1. **Dentro de la Reunión:** Mientras estás en una reunión de Teams, haz clic en los tres puntos (...) en la parte superior y selecciona "Configuración" -> "Configuración de dispositivos". 2. **Revisar la Configuración de Audio:** En el menú desplegable "Dispositivos de audio", asegúrate de que tu dispositivo principal (ej: "Logitech USB Headset", "Jabra Evolve 75") esté seleccionado. 3. **Realizar una Llamada de Prueba:** En este mismo panel de configuración, haz clic en "Hacer una llamada de prueba". Teams llamará a un bot que te pedirá que grabe un mensaje corto y luego lo reproducirá. Esto confirma si tu micrófono está funcionando correctamente dentro de la aplicación. Si sigues teniendo problemas después de realizar estos pasos, te recomiendo crear un ticket en http://support.innovatech.com (Categoría: "Software y Aplicaciones"), incluyendo: el modelo de tus auriculares/cámara web y una captura de pantalla de tu "Configuración de dispositivos" en Teams.	0.833	0.944	0.189	1.000
16	¿Cuál es el código de vestimenta en Innovatech Solutions?	Entiendo que estás preguntando sobre el código de vestimenta en Innovatech Solutions. Nuestro código de vestimenta es "Business Casual". Se espera que los empleados vistan de manera profesional y apropiada para un entorno de oficina.	1.000	0.667	1.000	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
17	¿Cuál es la misión de Innovatech Solutions?	Entiendo que estás buscando información sobre los objetivos de Innovatech Solutions. Innovatech Solutions busca proporcionar soluciones tecnológicas innovadoras y de alta calidad para sus clientes, impulsando el crecimiento y la eficiencia de sus negocios a través de soluciones tecnológicas a medida. Según nuestro manual, nuestra misión es desarrollar soluciones tecnológicas innovadoras que empoderen a nuestros clientes y simplifiquen la complejidad del mundo digital. Además, nuestro valor de "Enfoque en el Cliente" se centra en el éxito de nuestros clientes como nuestro propio éxito. ¿Hay algo más específico que te gustaría saber sobre nuestros objetivos o cómo los alcanzamos?	1.000	0.667	0.934	1.000
18	¿Cuál es el horario laboral estándar?	El horario laboral estándar en Innovatech Solutions es de lunes a viernes, de 9:00 a.m. a 6:00 p.m., con una hora para el almuerzo. Se espera flexibilidad según las necesidades del proyecto y el rol.	1.000	1.000	0.674	1.000
19	¿Innovatech permite el trabajo remoto?	Innovatech ofrece un modelo de trabajo híbrido. Los detalles y la elegibilidad se definen a nivel de equipo y deben ser aprobados por el gerente directo.	1.000	1.000	0.823	1.000
20	¿Cuáles son los valores fundamentales de Innovatech?	Entiendo que desea conocer los valores fundamentales que guían a Innovatech Solutions. Según el Manual del Empleado, los valores fundamentales son: * **Innovación Constante:** Abrazamos el cambio y buscamos nuevas y mejores formas de hacer las cosas. * **Integridad Absoluta:** Actuamos con honestidad y transparencia en todas nuestras interacciones. * **Colaboración Radical:** Creemos que los mejores resultados surgen del trabajo en equipo y la diversidad de pensamiento. * **Enfoque en el Cliente:** El éxito de nuestros clientes es nuestro éxito. * **Pasión por la Excelencia:** Nos esforzamos por la más alta calidad en todo lo que hacemos. No tengo información sobre eso en la documentación disponible. Te recomiendo contactar al soporte en http://support.innovatech.com	1.000	0.667	0.669	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
21	¿Cómo agrego una impresora de red en Windows 10 u 11?	Claro, con gusto te ayudaré con eso. Para agregar una impresora de red a un sistema operativo Windows 10 o 11, siga estos pasos: 1. Asegúrese de estar conectado a la red de la oficina (ya sea por cable Ethernet o por la red Wi-Fi "Innovatech_Corp"). 2. Abra el Explorador de Archivos (la carpeta amarilla en la barra de tareas o presione Tecla de Windows + E). 3. En la barra de direcciones en la parte superior, escriba la siguiente ruta y presione Enter: \\printserv01 4. Aparecerá una ventana con la lista de todas las impresoras de red disponibles. 5. Busque la impresora que desea agregar basándose en la nomenclatura descrita anteriormente. 6. Haga doble clic sobre el nombre de la impresora. 7. Windows comenzará a instalar los controladores automáticamente. Este proceso puede tardar uno o dos minutos. 8. Una vez finalizado, la impresora estará disponible en su lista de impresoras. 9. Para establecerla como predeterminada, vaya a Inicio > Configuración > Bluetooth y dispositivos > Impresoras y escáneres, seleccione la impresora y haga clic en "Establecer como predeterminada". Si tiene problemas, revise el apartado 5 de la guía de usuario.	1.000	0.933	0.924	1.000
22	¿Cómo agrego una impresora de red en una Mac (macOS)?	Claro, puedo ayudarte con eso. Aquí tienes los pasos para agregar una impresora de red a tu Mac (macOS) según la documentación disponible: Para agregar una impresora de red a tu Mac, sigue estos pasos: 1. Asegúrate de estar conectado a la red de la oficina. 2. Abre “Ajustes del Sistema” (desde el menú Apple o el Dock). 3. En la barra lateral, selecciona “Impresoras y escáneres”. 4. Haz clic en el botón “Añadir impresora, escáner o fax…”. 5. Aparecerá una ventana. Las impresoras de red de Innovatech deberían aparecer automáticamente en la pestaña "Por omisión". 6. Selecciona la impresora que deseas agregar de la lista. macOS detectará y seleccionará automáticamente el controlador correcto (generalmente “AirPrint”). 7. Haz clic en el botón “Añadir”. La impresora se agregará a tu lista y estará lista para usar. Si sigues teniendo problemas, por favor, crea un ticket en http://support.innovatech.com (Categoría: "Hardware e Impresoras"), especificando el nombre de la impresora que intenta instalar y el mensaje de error que recibe.	0.500	1.000	0.892	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
23	Escribí \\printserv01 en Windows pero no veo ninguna impresora. ¿Cuál puede ser el problema?	Entiendo que al intentar acceder a una impresora escribiendo "\\printserv01" en Windows, no la encuentra. Para solucionar este problema, te sugiero los siguientes pasos: 1. **Verificar la dirección:** Asegúrate de que la dirección "\\printserv01" sea correcta. En la documentación de Innovatech Solutions, no se menciona directamente problemas con direcciones de impresión incorrectas. Te recomiendo contactar al soporte en http://support.innovatech.com para confirmar que esa es la dirección correcta de la impresora compartida. 2. **Verificar la configuración de red:** Asegúrate de que tu ordenador esté conectado a la red de Innovatech Solutions. 3. **Consultar el soporte:** Dado que esta es una situación específica, te recomiendo crear un ticket de soporte en http://support.innovatech.com (Categoría: "Red y Conectividad"). Incluye en el ticket: * Una descripción detallada del problema. * La información de tu ubicación. * Los pasos que ya ha intentado.	0.000	0.182	0.821	0.000
24	¿Qué significa el nombre de impresora "NYC-FIN-BW-01"?	El nombre de impresora "NYC-FIN-BW-01" significa que es una impresora blanco y negro de Finanzas en Nueva York. Esta nomenclatura sigue el estándar de Innovatech Solutions para identificar sus impresoras de red.	1.000	1.000	0.996	1.000
25	¿En qué categoría del portal de soporte debo crear un ticket si no puedo instalar una impresora?	Para reportar un problema con la instalación de una impresora a través del portal de soporte de Innovatech Solutions, por favor, siga estos pasos: 1. **Crear un Ticket de Solicitud:** * Acceda al Portal de Soporte de TI en: http://support.innovatech.com * Inicie sesión con sus credenciales corporativas. * Haga clic en "Crear Nuevo Ticket" o "Catálogo de Servicios". * Seleccione la categoría: "Software y Aplicaciones". 2. **Completar el Formulario de Solicitud:** * Título del Ticket: Sea descriptivo. Ej: "Problema al instalar la impresora [Modelo de la Impresora]". * Proporcione detalles sobre el problema, incluyendo el modelo de la impresora y cualquier mensaje de error que esté viendo. * Adjunte una captura de pantalla del mensaje de error. Si estos pasos no resuelven el problema, por favor, cree un ticket en http://support.innovatech.com (Categoría: "Software y Aplicaciones"), incluyendo el modelo de sus auriculares/cámara web (si aplica) y los resultados de la llamada de prueba.	0.000	0.667	0.564	0.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
26	¿Cuál es la herramienta integrada de Windows recomendada para eliminar archivos temporales de forma segura?	Para eliminar archivos temporales de forma segura, se recomienda utilizar la herramienta integrada de Windows llamada "Liberador de espacio en disco". Para utilizarla: 1. Haga clic en el botón de Inicio y escriba "Liberador de espacio en disco". 2. Seleccione la aplicación de la lista. 3. Asegúrese de que la unidad seleccionada sea (C:) y haga clic en "Aceptar". 4. La herramienta calculará el espacio que se puede liberar. 5. En la ventana que aparece, marque las casillas de los archivos que desea eliminar. Se recomienda marcar: Archivos de programa descargados, Archivos temporales de Internet, Archivos de optimización de distribución, Papelera de reciclaje (si no lo hizo en el paso 1), Archivos temporales y Miniaturas. 6. Haga clic en "Aceptar" y luego en "Eliminar archivos" para confirmar.	1.000	1.000	0.908	1.000
27	¿De qué carpetas del sistema NO debo eliminar archivos manualmente?	Entiendo que desea saber cuáles carpetas del sistema de Windows no debe eliminar manualmente para evitar problemas. No tengo información sobre eso en la documentación disponible. Te recomiendo contactar al soporte en http://support.innovatech.com.	1.000	0.667	0.000	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
28	¿Cómo puedo hacer que mis archivos de OneDrive dejen de ocupar espacio en mi disco local?	Para optimizar el almacenamiento de tus archivos de OneDrive en tu disco local, tienes varias opciones: 1. **Liberar espacio en OneDrive (Archivos a petición):** Puedes usar la función "Liberar espacio" en OneDrive. Busca el icono de la nube en tu barra de tareas, haz clic derecho sobre él y selecciona "Configuración". Ve a la pestaña "Configuración" y asegúrate de que la opción "Ahorrar espacio y descargar archivos al usarlos" esté marcada. Luego, abre tu carpeta de OneDrive en el Explorador de Archivos, busca carpetas o archivos grandes que no necesites tener disponibles sin conexión (identificados por un icono de marca de verificación verde) y haz clic derecho sobre ellos y selecciona "Liberar espacio". El icono cambiará a una nube azul, indicando que el archivo está solo en la nube y no ocupa espacio en tu disco. 2. **Sincronización Selectiva (No disponible a través de la interfaz de usuario de OneDrive):** No existe una función directa de "sincronización selectiva" a través de la interfaz de usuario de OneDrive para controlar qué archivos se descargan o sincronizan. Sin embargo, puedes lograr un efecto similar al utilizar la función de "Archivos a petición" (descrita anteriormente). Al seleccionar los archivos para "Liberar espacio", estás efectivamente indicándole a OneDrive que no descargue esos archivos a tu disco local. Para solicitudes de acceso a software o sistemas, por favor, crea un ticket en http://support.innovatech.com (Categoría: "Hardware y Rendimiento") con la siguiente información: * Título del Ticket: Sea descriptivo. Ej: "Solicitud de Acceso a Salesforce para". * Nombre del Software/Sistema: Especifique el nombre exacto de la aplicación o recurso. * Tipo de Acceso Requerido: Describa el nivel de permiso necesario (ej: Lectura, Edición, Administrador). * Justificación del Negocio: Este es el campo más importante. Explique por qué necesita este acceso para realizar sus funciones laborales. Describa las tareas que no puede realizar sin él. * Nombre del Gerente Aprobador: Ingrese el nombre completo de su gerente directo. * Usuario de Referencia (Opcional): Ingrese el nombre de un colega con los permisos que necesita.	1.000	0.914	0.794	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
29	¿Qué debo hacer si mi disco duro sigue peligrosamente lleno después de seguir todos los pasos de la guía?	Comprendo que estás buscando información adicional sobre cómo diagnosticar un disco duro que sigue lleno después de haber seguido los pasos de optimización que te proporcioné. Aunque la documentación disponible no incluye los pasos adicionales que mencionas (análisis de malware, revisión de carpetas del sistema, deduplicación de archivos, etc.), sí puedo ayudarte con lo que sí está incluido en la guía. Para ayudarte a diagnosticar la situación, te recomiendo seguir estos pasos: 1. **Vaciar la Papelera de Reciclaje:** Los archivos eliminados permanecen en la Papelera de Reciclaje y siguen ocupando espacio. Busca el icono de la Papelera de Reciclaje en tu Escritorio y selecciona "Vaciar papelera de reciclaje". 2. **Utilizar la Herramienta "Liberador de espacio en disco":** Puedes encontrar esta herramienta haciendo clic en el botón de Inicio y escribiendo "Liberador de espacio en disco". Selecciona la unidad (C:) y marca las casillas de los archivos que deseas eliminar (como Archivos de programa descargados, Archivos temporales de Internet, Archivos de optimización de distribución, Papelera de reciclaje si no la vaciaste, archivos temporales y miniaturas). 3. **Limpiar la Carpeta de Descargas:** Abre el Explorador de Archivos (Windows + E), ve a la carpeta "Descargas" y elimina archivos que ya no necesites, moviendo archivos importantes a OneDrive o carpetas de proyectos. 4. **Gestionar Archivos de OneDrive:** Asegúrate de que la opción "Ahorrar espacio y descargar archivos al usarlos" esté marcada en la configuración de OneDrive. Si después de seguir estos pasos tu disco duro sigue peligrosamente lleno (menos del 10% de espacio libre), por favor, crea un ticket en http://support.innovatech.com (Categoría: "Hardware y Rendimiento") para que un técnico pueda investigar más a fondo. No tengo información sobre eso en la documentación disponible. Te recomiendo contactar al soporte en http://support.innovatech.com	1.000	0.850	0.686	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
30	¿Qué tipos de archivos se recomienda marcar en el Liberador de espacio en disco?	Claro, puedo ayudarte con eso. Según la guía de usuario, para liberar el mayor espacio en tu disco duro utilizando la herramienta "Liberador de espacio en disco" de Windows, se recomienda marcar las siguientes casillas de archivos: * Archivos de programa descargados * Archivos temporales de Internet * Archivos de optimización de distribución * Papelera de reciclaje (si no lo hizo en el paso 1) * Archivos temporales * Miniaturas Es importante recordar que este documento se basa únicamente en la información proporcionada en la guía de usuario. Si tienes alguna otra pregunta, no dudes en consultarme.	0.500	0.727	0.712	1.000
31	¿Cómo configuro mi respuesta de "Fuera de la Oficina" en la versión web de Outlook?	Claro, puedo ayudarte con eso. Para configurar tu respuesta automática de "Fuera de la Oficina" en la versión web de Outlook: 1. Inicia sesión en http://outlook.office.com. 2. Haz clic en el icono de engranaje (Configuración) en la esquina superior derecha. 3. Selecciona "Correo" -> "Respuestas automáticas". 4. Activa el interruptor "Activar respuestas automáticas". 5. Sigue los pasos 6, 7 y 8 del procedimiento anterior para establecer el intervalo de tiempo y redactar tus mensajes. 6. Haz clic en "Guardar" en la parte inferior. Si tienes problemas durante la configuración, por favor, crea un ticket en http://support.innovatech.com en la categoría "Dispositivos Móviles".	0.500	0.875	0.948	1.000
34	¿Cómo activo las respuestas automáticas en la aplicación de escritorio de Outlook?	Para activar las respuestas automáticas en la aplicación de escritorio de Outlook, siga estos pasos: 1. Abra la aplicación de Outlook. 2. Haga clic en "Archivo" en la esquina superior izquierda. 3. En la pantalla de "Información de la cuenta", haga clic en el botón "Respuestas automáticas". 4. En la ventana que aparece, seleccione la opción "Enviar respuestas automáticas". 5. (Recomendado) Marque la casilla "Enviar solo durante este intervalo de tiempo". 6. Establezca la Fecha y Hora de inicio y la Fecha y Hora de finalización. Outlook activará y desactivará automáticamente las respuestas en las fechas que defina. 7. Redacte sus Mensajes: Hay dos pestañas: "Dentro de mi organización" y "Fuera de mi organización". 8. Toque "Guardar". Su cuenta ya está configurada.	0.500	1.000	0.477	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
32	¿Puedo configurar un mensaje de fuera de oficina diferente para mis colegas y para personas externas?	Sí, el sistema de Outlook web permite la configuración de mensajes de "Fuera de la Oficina" distintos para destinatarios internos y externos. Puedes redactar dos mensajes separados utilizando las pestañas "Dentro de mi organización" y "Fuera de la organización". El mensaje "Dentro de mi organización" se enviará a tus colegas de Innovatech Solutions, mientras que el mensaje "Fuera de la organización" se enviará a cualquier persona externa. Puedes editar el contenido de cada mensaje de forma independiente. Para hacerlo, sigue estos pasos: 1. Inicia sesión en http://outlook.office.com. 2. Haz clic en el icono de engranaje (Configuración) en la esquina superior derecha. 3. Selecciona "Correo" -> "Respuestas automáticas". 4. Utiliza las pestañas "Dentro de mi organización" y "Fuera de la organización" para redactar tus mensajes de respuesta automática. 5. Haz clic en "Guardar" en la parte inferior.	1.000	1.000	0.650	1.000
33	¿Activar las respuestas automáticas también rechaza las invitaciones a reuniones que reciba?	No, activar las respuestas automáticas en Outlook no rechaza automáticamente las invitaciones a reuniones. Recibirás las invitaciones a reuniones como si no tuvieras la respuesta automática habilitada. Es tu responsabilidad revisar tu bandeja de entrada y aceptar o rechazar las invitaciones según tu disponibilidad.	1.000	0.333	0.904	1.000
35	¿Qué pasa si olvido poner una fecha de finalización a mi respuesta automática?	Si no establece un intervalo de tiempo para su respuesta automática, Outlook le mostrará un recordatorio prominente en la parte superior de la aplicación para que la desactive manualmente a su regreso. Por favor, contacte a la Mesa de Ayuda de TI a través de http://support.innovatech.com si no puede completar este proceso.	1.000	1.000	0.721	1.000
36	¿Cuáles son los requisitos para una contraseña segura en Innovatech?	Las contraseñas deben cumplir con los siguientes requisitos: * Tienen una longitud mínima de 12 caracteres. * Deben incluir una combinación de letras mayúsculas, minúsculas, números y símbolos. * Está prohibido reutilizar contraseñas o compartirlas con otras personas. * Se requiere el uso de autenticación de dos factores (2FA) en todas las cuentas compatibles.	1.000	1.000	0.673	1.000
37	¿Qué debo hacer si sospecho que he recibido un correo de phishing?	Si sospechas que recibiste un correo electrónico que podría ser un intento de phishing, debes reportarlo inmediatamente al Help Desk de TI a través del correo security@innovatech.com. Es crucial reportar cualquier sospecha de incidente de seguridad lo antes posible.	1.000	1.000	0.876	1.000
38	Según la política de seguridad, ¿qué es la clasificación de datos "Interno"?	Según la Política de Seguridad de la Información de Innovatech Solutions, la clasificación de datos "Interno" se refiere a información para uso exclusivo de los empleados de Innovatech (ej. manuales, guías).	1.000	1.000	0.794	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
39	¿Está permitido instalar software no autorizado en mi laptop corporativa?	No tengo información sobre eso en la documentación disponible. Te recomiendo contactar al soporte en http://support.innovatech.com	0.333	1.000	0.000	0.000
40	Según la política de seguridad, ¿qué debo hacer siempre que me ausente de mi puesto de trabajo?	Para asegurar la seguridad de tus datos y sistemas cuando te ausentes temporalmente de tu puesto de trabajo, debes seguir estos pasos: 1. **Bloquear tu dispositivo:** Debes bloquear inmediatamente tu dispositivo (laptop, móvil) proporcionado por la empresa cada vez que te ausentes de tu puesto de trabajo. Esto ayuda a proteger tu información confidencial y evitar el acceso no autorizado.	1.000	0.833	0.085	1.000
42	¿Cuál es el campo más importante que debo llenar al solicitar un nuevo acceso a un sistema?	Comprendo que has identificado correctamente el campo más importante para completar al solicitar acceso a un sistema a través del portal de soporte de Innovatech Solutions. De acuerdo con el Procedimiento Operativo Estándar SOP-TI-004, el campo más importante es la **Justificación del Negocio**. Esta sección es crucial porque permite al equipo de TI comprender claramente la necesidad del acceso, la dependencia del usuario y el impacto de la falta del acceso en el desempeño de sus funciones laborales. Recuerda, como se indica en el documento, “Este es el campo más importante. Explique por qué necesita este acceso para realizar sus tareas laborales. Describa las tareas que no puede realizar sin él.”	0.500	0.667	0.790	1.000
48	¿Cuáles son los requisitos que debe cumplir mi nueva contraseña corporativa?	De acuerdo con la Política de Seguridad de la Información (IS-POL-001), las contraseñas corporativas de Innovatech Solutions deben cumplir con los siguientes criterios: * Una longitud mínima de 12 caracteres. * Deben incluir al menos una letra mayúscula, una letra minúscula, un número y un símbolo. * Está prohibido reutilizar contraseñas o compartirlas con otras personas. * Se requiere el uso de autenticación de dos factores (2FA) en todas las cuentas compatibles.	1.000	1.000	0.411	0.500
50	¿El departamento de TI alguna vez me pedirá mi contraseña por correo electrónico?	No tengo información sobre eso en la documentación disponible. Te recomiendo contacte a la Mesa de Ayuda de TI: Portal de Soporte: http://support.innovatech.com (Crear un ticket en la categoría "Cuentas y Accesos"). Teléfono: Extensión interna 5555 (Opción 1).	0.833	0.833	0.000	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
41	¿Cómo solicito acceso a un nuevo software, como Salesforce?	Para solicitar acceso a un nuevo software como Salesforce en Innovatech Solutions, siga estos pasos: 1. **Identifique la Necesidad:** Antes de crear la solicitud, asegúrese de identificar claramente el sistema o recurso al que necesita acceso (ej: "Acceso de solo lectura a la carpeta de Marketing en el servidor \\FILESERV01", "Licencia para Adobe Creative Cloud", "Acceso al entorno de pruebas de JIRA"). 2. **Crear un Ticket de Solicitud:** * Acceda al Portal de Soporte de TI en: http://support.innovatech.com * Inicie sesión con sus credenciales corporativas. * Haga clic en "Crear Nuevo Ticket" o "Catálogo de Servicios". * Seleccione la categoría: "Solicitudes de Servicio" -> "Cuentas y Accesos". * Complete el formulario: * **Título del Ticket:** Sea descriptivo. Ej: "Solicitud de Acceso a Salesforce para". * **Nombre del Software/Sistema:** Especifique el nombre exacto de la aplicación o recurso (ej: Salesforce). * **Tipo de Acceso Requerido:** Describa el nivel de permiso necesario (ej: Lectura, Edición, Administrador). * **Justificación del Negocio:** Este es el campo más importante. Explique por qué necesita este acceso para realizar sus funciones laborales. Describa las tareas que no puede realizar sin él. * **Nombre del Gerente Aprobador:** Ingrese el nombre completo de su gerente directo. * **Usuario de Referencia (Opcional):** Ingrese el nombre de un colega con los permisos que necesita. 3. **Aprobación por el Gerente:** Una vez enviado el ticket, el sistema enviará automáticamente una notificación por correo electrónico a su gerente directo para que apruebe la solicitud. El gerente deberá hacer clic en el enlace del correo y aprobar o rechazar la solicitud. 4. **Procesamiento por TI:** Una vez aprobada, el ticket se asignará a la cola del equipo de TI correspondiente, quien procesará la solicitud (típicamente en 2-3 días hábiles). 5. **Notificación de Cumplimiento:** Una vez aprobada, recibirá una notificación por correo electrónico confirmando que la solicitud ha sido completada.	1.000	1.000	0.560	1.000
43	¿Quién debe aprobar mi solicitud de software antes de que TI la procese?	Tu gerente directo debe aprobar tu solicitud de software antes de que el equipo de TI pueda procesarla. Según el Procedimiento Operativo Estándar (SOP-TI-004), una vez que hayas creado el ticket en el Portal de Soporte de TI (http://support.innovatech.com) y hayas completado los campos requeridos, el sistema enviará automáticamente una notificación a tu gerente directo para que apruebe o rechace la solicitud. El ticket permanecerá en estado "Esperando Aprobación" hasta que tu gerente lo haga. Para garantizar un acceso seguro y el cumplimiento de la Política de Seguridad de la Información, el principio de mínimo privilegio se aplica, y los accesos a sistemas críticos son revisados trimestralmente.	1.000	1.000	0.814	1.000
44	La política de acceso menciona el "Principio de Mínimo Privilegio". ¿Qué significa eso?	El "Principio de Mínimo Privilegio" significa que cada usuario, empleado o sistema solo debe tener acceso a la información y los recursos necesarios para realizar sus tareas laborales específicas. Esto limita la exposición a riesgos de seguridad en caso de que un usuario sea comprometido o si un sistema es vulnerado.	1.000	0.500	0.923	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
45	¿Cómo puedo ayudar a TI a saber qué permisos exactos necesito para un software?	Entiendo. Para asegurar que TI pueda asignar los permisos de software correctos para tu función, es fundamental proporcionar detalles específicos sobre tus necesidades. En base a la documentación disponible, te recomiendo que completes el formulario de solicitud de acceso a sistemas siguiendo los siguientes pasos: 1. **Describe tus Tareas:** Describe con precisión las tareas específicas que realizarás con el software. Por ejemplo, podrías indicar: “Necesito acceso de lectura a los datos de clientes en Salesforce para poder gestionar las relaciones con los clientes.” 2. **Especifica los Datos Necesarios:** Indica qué tipos de datos necesitas acceder y manipular. Por ejemplo, “Necesito acceder a los registros de clientes, incluyendo información de contacto, historial de compras y oportunidades de ventas.” 3. **Indica el Nivel de Acceso:** Sé claro sobre tu necesidad de acceso. Esto podría ser: “Necesito acceso de solo lectura para evitar modificaciones accidentales” o “Necesito acceso de escritura para poder crear y modificar registros”. 4. **Justifica tu Necesidad:** Explica por qué necesitas ese nivel de acceso. Por ejemplo, “Necesito acceso de escritura porque soy el responsable de crear nuevos prospectos y gestionar la información de contacto de los clientes.” Al proporcionar esta información detallada, nos aseguramos de que TI pueda evaluar tu solicitud y asignarte los permisos necesarios siguiendo el principio de "mínimo privilegio”. Recuerda que el objetivo es garantizar que solo se te otorguen los permisos que *realmente* necesitas para llevar a cabo tus funciones laborales.	1.000	0.600	0.324	1.000
46	¿Cuál es la URL para restablecer mi contraseña olvidada?	Para restablecer tu contraseña olvidada en Innovatech Solutions, por favor, crea un ticket en la categoría “Cuentas y Accesos” en nuestro Portal de Soporte: http://support.innovatech.com. También puedes contactar a la Mesa de Ayuda de TI a través del teléfono de extensión interna 5555 (Opción 1).	0.000	1.000	0.637	0.000
49	¿Qué hago si no he configurado mis métodos de recuperación (MFA, preguntas) y no puedo usar el autoservicio de contraseña?	Entiendo que no ha configurado previamente los métodos de recuperación de contraseña (autenticación multifactorial o preguntas de seguridad) y por lo tanto, no puede utilizar el autoservicio de restablecimiento de contraseña. Para restablecer su contraseña en esta situación, siga estos pasos: 1. **Contacte a la Mesa de Ayuda de TI:** Debido a que no ha configurado los métodos de recuperación necesarios, deberá contactar directamente a la mesa de ayuda para que puedan asistirlo. 2. **Portal de Soporte:** Puede acceder al portal de soporte en: http://support.innovatech.com. 3. **Crear un Ticket:** Dentro del portal, cree un nuevo ticket en la categoría "Cuentas y Accesos". Al crear el ticket, describa su situación y que no ha configurado los métodos de recuperación. 4. **Alternativas:** La mesa de ayuda podrá guiarlo a través del proceso para configurar los métodos de recuperación, o, si es necesario, solicitará el restablecimiento de su contraseña directamente. Es importante recordar que el departamento de TI nunca solicitará su contraseña por correo electrónico o teléfono. Si recibe una solicitud de este tipo, repórtela inmediatamente a security@innovatech.com.	0.333	0.563	0.109	1.000

ID	Pregunta (Input)	Respuesta Generada (Output)	CP	F	AR	CR
47	¿Cuántos métodos de verificación debo completar para restablecer mi contraseña usando el autoservicio?	Para restablecer tu contraseña a través del autoservicio, sigue estos pasos específicos y métodos de verificación: **1. Acceder al Portal de Autoservicio:** * Abre un navegador web (Chrome, Edge, Firefox) y dirígete a la siguiente URL: https://passwordreset.innovatech.com **2. Proceso de Recuperación (Pasos Específicos):** * **Iniciar el Proceso de Recuperación:** Debajo de los campos de inicio de sesión, haz clic en el enlace "¿No puede acceder a su cuenta?" o "He olvidado mi contraseña". * **Verificación de Identidad - Paso 1 (Usuario):** * Ingresa tu ID de Usuario (tu correo electrónico corporativo completo, ej: nombre.apellido@innovatech.com). * Completa el desafío de CAPTCHA. * Haz clic en "Siguiente". * **Verificación de Identidad - Paso 2 (Método de Autenticación):** El sistema te presentará las opciones de verificación que configuraste: * **Opción A: Código vía App de Autenticación:** * Selecciona "Usar mi aplicación de autenticación". * Abre la aplicación Microsoft Authenticator (o similar) en tu teléfono móvil. * Aprueba la notificación de inicio de sesión. * **Opción B: Código vía SMS:** * Selecciona "Enviar un mensaje de texto a mi teléfono móvil". * Recibirás un código de 6 dígitos. * Ingresa ese código en el portal y haz clic en "Siguiente". * **Opción C: Preguntas de Seguridad:** * Selecciona "Responder mis preguntas de seguridad". * Responde correctamente a las preguntas que configuraste (no distinguen entre mayúsculas y minúsculas). * **Establecer tu Nueva Contraseña:** * Una vez verificada tu identidad, serás redirigido a la página para crear una nueva contraseña. * Ingresa tu nueva contraseña en el primer campo. * Confirma la nueva contraseña en el segundo campo. * **Requisitos de la Contraseña:** Asegúrate de cumplir con los siguientes requisitos: * Mínimo 12 caracteres. * Debe contener al menos una letra mayúscula (A-Z), una letra minúscula (a-z), un número (0-9) y un símbolo (!, @, #, $, etc.). * No puede ser una de tus últimas 5 contraseñas utilizadas. Ház clic en "Finalizar". Si el proceso fue exitoso, verás un mensaje de confirmación. Tu contraseña ha sido cambiada. Espera entre 2 y 5 minutos para que el cambio se sincronice completamente. ¿Necesitas ayuda con algo más?	1.000	0.947	0.753	1.000