UNIVERSIDAD CATÓLICA DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

FACULTAD DE INFORMÁTICA, CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA

CARRERA DE SOFTWARE

Asistente inteligente para consulta asistida por IA y recuperación de información en documentos PDF académicos

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN SOFTWARE

AUTORA: CARLA ANDREA ENCALADA ARÉVALO

DIRECTOR: ING. MILTON ALFREDO CAMPOVERDE MOLINA, PHD

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

FACULTAD DE INFORMÁTICA, CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA

CARRERA DE SOFTWARE

Asistente inteligente para consulta asistida por IA y recuperación de información en documentos PDF académicos

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN SOFTWARE

AUTORA: CARLA ANDREA ENCALADA ARÉVALO

DIRECTOR: ING. MILTON ALFREDO CAMPOVERDE MOLINA, PHD

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO

Universidad Católica de Cuenca

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD

Declaratoria de Autoría y Responsabilidad

Carla Andrea Encalada Arévalo portador(a) de la cédula de ciudadanía Nº **0106515018**. Declaro ser el autor de la obra: "**Asistente inteligente para consulta asistida por IA y recuperación de información en documentos PDF académicos**", sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, **09 de febrero de 2026**

Carla Encalada.

F: .....

Carla Andrea Encalada Arévalo

C.I. **0106515018**

Universidad Católica de Cuenca	**CERTIFICADO DEL TUTOR**

Yo, Milton Alfredo Campoverde Molina, certifico que el presente trabajo de investigación, con el título **"Asistente inteligente para consulta asistida por IA y recuperación de información en documentos PDF académicos"**, fue desarrollado por Carla Andrea Encalada Arévalo, con número de cédula 0106515018, bajo mi supervisión.

MILTON ALFREDO CAMPOVERDE MOLINA

Firmado digitalmente por MILTON ALFREDO CAMPOVERDE MOLINA
Fecha: 2026.02.09 18:55:48 -05'00'

F: .....

Ing. Milton Campoverde Molina PhD.

C.I. **0103902532**

Dedicatoria

Dedico este trabajo a mi mami Sandra, por no soltarme nunca y por ser refugio, abrazo y fortaleza cuando sentía que no podía más. Gracias por celebrar cada logro, por pequeño que haya sido, y por estar siempre presente. Eres mi mayor referente femenino y un ejemplo de vida lleno de cariño, responsabilidad, resiliencia y alegría. Este logro es totalmente tuyo.

A mi papi Carlos, por cuidarme y acompañarme siempre con muchísimo amor. Aunque te fuiste muy pronto, tus enseñanzas, tus consejos y los recuerdos que compartimos permanecen conmigo. Por eso, mi bello ser de luz, este logro lleva tu nombre y tu esencia. Un abrazo de oso hasta el cielo.

A mi Nena Blanca, por cuidarme con un amor sincero que no pide nada, solo acompaña. Gracias por su bondad, por creer en mí incluso cuando yo no podía hacerlo, y por cuidarme como una segunda mamá. Este logro también lleva su amor.

A mis ñaños mayores, Mercedes y Andrés, por sostenerme entre risas, lágrimas y conversaciones hasta el amanecer. Ustedes son una gran inspiración en mi vida. Giba, gracias por tu resiliencia, empatía, esfuerzo y lealtad. Mi estrellita, gracias por tu bondad, responsabilidad y autenticidad. Sin ustedes, esta meta no habría sido posible.

A mis abuelitos, por enseñarme que la ternura también es fuerza. Abuelito Vichi, gracias por cuidarme como si aún tuviera siete años y devolverme luz en momentos oscuros. Abue Nelly, gracias por sus abrazos, sus oraciones y platos llenos de amor. Abue César, gracias por su fuerza silenciosa y cariño sincero. Y a mi otro ángel, mi abuelita Mari, gracias por hacerme sentir siempre amada y por su ejemplo de sabiduría, que espero algún día alcanzar. Este logro también es de ustedes cuatro, porque mi camino se ha construido sobre los cimientos de su esfuerzo y sacrificio.

Finalmente, dedico esta meta a mi Bruno, quien, sin una palabra y solo con la mirada, me enseñó tantas cosas de la vida. Con un amor fiel, estuvo ahí cuando más triste me sentía. Gracias por llegar a mi vida como el mejor regalo de Dios.

Agradecimientos

A Dios, por su guía constante y por acompañarme en cada etapa de este proceso, incluso en los momentos de mayor dificultad.

A mi familia, por ser un pilar fundamental a lo largo de este camino. Su apoyo, sus consejos y, sobre todo, su amor han sido una fuente permanente de fortaleza. De manera especial, agradezco a mi mami por su respaldo incondicional durante esta etapa; a mi papi, por cuidarme desde arriba; a mi segunda mamá, por creer en mí cuando más lo necesitaba; a mis ñaños, por su compañía y ayuda constante; y a mis abuelitos, por su amor, sus oraciones y el acompañamiento que me brindaron durante todo este recorrido.

A mis profesores y profesoras, por su compromiso y por las enseñanzas que aportaron tanto a mi formación profesional como personal. De manera particular, expreso mi agradecimiento a mi tutor, el ingeniero Milton Campoverde, por su guía y orientación desde el primer día. Asimismo, agradezco a los ingenieros Junior, Jhon y Jheison por su apoyo en los momentos más exigentes de este trabajo y por compartir sus conocimientos con generosidad. A la ingeniera Natalia Peralta, por sus consejos oportunos para culminar esta meta. Finalmente, agradezco a Marietita por su apoyo constante, el cual hizo posible cerrar esta etapa con éxito.

A mis mejores amigas del colegio, quienes, incluso a la distancia, me alentaron a no rendirme mediante su ejemplo y palabras de apoyo. Asimismo, a mis amigas y amigos que me acompañaron a lo largo de mi etapa universitaria, por los momentos compartidos que hicieron más llevadero este proceso.

A todos ustedes, mi más sincero agradecimiento por ser parte esencial de este logro.

Resumen

El objetivo del trabajo fue desarrollar y verificar localmente un asistente inteligente para consultar documentos académicos de un repositorio de la Universidad Católica de Cuenca. En este contexto, el problema consiste en que los archivos extensos dificultan la localización rápida y precisa de información cuando se depende únicamente de la búsqueda tradicional por palabras clave. Para lo cual, implementamos un prototipo que convierte el contenido en representaciones vectoriales, búsqueda semántica con la biblioteca Facebook AI Similarity Search (FAISS) y respuestas en ejecución local con Ollama. El prototipo tiene una arquitectura cliente–servidor, integra la carga de documentos, segmentación por fragmentos, recuperación semántica y persistencia del historial mediante generación aumentada con recuperación (Retrieval Augmented Generation, RAG). Además, el prototipo incorpora la referencia del documento analizado e indica con claridad cuando no existe evidencia suficiente para responder. La metodología utilizada fue Scrum y realizamos pruebas de extremo a extremo en carga individual y por carpeta. También, aplicamos una encuesta de percepción a cuatro participantes para valorar facilidad, claridad, utilidad y experiencia de usuario. En cuanto a los resultados de rendimiento, se obtuvo que la versión actual del prototipo aumenta el tiempo de ingesta e indexación de documentos frente a la versión inicial. Esto se debe a que la segunda versión genera más *embeddings*, pero mejora la recuperación de evidencias y la calidad de respuesta. Se concluye que un asistente documental inteligente optimiza el proceso de consulta al reducir los tiempos de búsqueda y eliminar la dependencia de servicios externos.

Palabras clave: *asistente inteligente,* Embeddings, *FAISS, Ollama, RAG.*

Abstract

The objective of this work was to develop and locally verify an intelligent assistant to query academic documents from a repository of the Catholic University of Cuenca. In this context, the problem is that lengthy documents make it difficult to quickly and accurately locate information when relying solely on traditional keyword search. To address this, we implemented a prototype that converts content into vector representations, performs semantic search using the Facebook AI Similarity Search (FAISS) library, and generates locally executed responses with Ollama. The prototype has a client-server architecture and integrates document loading, text chunking, semantic retrieval, and history persistence through Retrieval Augmented Generation (RAG). Additionally, the prototype incorporates references to the analyzed document and clearly indicates when there is insufficient evidence to answer. The methodology used was Scrum, and we conducted end-to-end tests on single-file and batch loading. We also applied a perception survey to four participants to assess ease of use, clarity, usefulness, and user experience. Regarding performance results, the current version of the prototype increases document ingestion and indexing time compared to the initial version. This is because the second version generates more embeddings but improves evidence retrieval and response quality. It is concluded that an intelligent document assistant optimizes the query process by reducing search times and eliminating dependence on external services.

Keywords: *Intelligent Assistant, Embeddings, FAISS, Ollama, RAG.*

Asistente inteligente para consulta asistida por IA y recuperación de información en documentos PDF académicos

Intelligent assistant for AI-assisted query and information retrieval in academic PDF documents

Introducción

A lo largo del tiempo, los asistentes conversacionales han experimentado una transformación sin precedentes, desde chatbots con respuestas preestablecidas a sistemas complejos basados en modelos de lenguaje de gran escala (Large Language Models, LLM). Esta transformación ha impulsado su incorporación en distintos entornos educativos. Dentro de este ámbito, herramientas como ChatGPT representan un cambio significativo gracias a su capacidad de generar textos y asistir en tareas académicas (Tramallino & Marize Zeni, 2024). No obstante, su implementación trae consigo debates sobre ética, transparencia y su uso apropiado (Bustamante Bula & Camacho Bonilla, 2024). En la enseñanza universitaria, en estudios recientes han analizado cómo se emplea ChatGPT, señalando ventajas, desventajas e indicando que las instituciones deben crear normas o políticas claras de uso (McGrath et al., 2025). También, estas herramientas se pueden utilizar como apoyo para la escritura de trabajos y la investigación (Khalifa & Albadawy, 2024). Sin embargo, se debe revisar con expertos para minimizar el sesgo y alucinación del contenido.

En Ecuador, considerando la necesidad de pautas para un uso responsable, se han realizado investigaciones acerca de cómo emplear asistentes conversacionales e Inteligencia Artificial (IA) en ámbitos de enseñanza. La incorporación de la IA en la formación universitaria es un reto para las organizaciones educativas. Se recalca la importancia de capacitar al personal, considerando aspectos morales y legales en caso de un mal uso (Campuzano-Vásquez et al., 2025). Asimismo, se presenta un desarrollo de un asistente de IA destinado a la orientación vocacional, que depende del perfil de competencias de los estudiantes de educación media (Montoya Naguas et al., 2025). En síntesis, la IA permite el autoaprendizaje y es un desafío su uso correcto (Anchapaxi-Díaz et al., 2024).

La inclusión de ChatGPT en la educación superior desde enfoques complementarios en Cuenca (Ecuador), ha permitido evaluar la opinión de 63 docentes de la Universidad Politécnica Salesiana. Para lo cual, se ha aplicado un instrumento de pruebas de validez y confiabilidad, que revela su uso como una gran herramienta para organizar actividades de clase, pero no se mide su impacto en el proceso educativo (Sigüenza Orellana et al., 2024). De forma similar, en la Universidad Católica de Cuenca se evaluó a 61 docentes de Odontología con un cuestionario validado. El 75% conoce IA, pero solo el 20% la usa por falta de capacitación e infraestructura limitada (Ortiz Vázquez & Marín Guamán, 2025). Asimismo, Carvallo y Erazo-Garzón (2023) evidencian el uso de IA como un soporte en los procesos de enseñanza y aprendizaje, en este caso específico en la materia de ingeniería de requisitos de la Universidad del Azuay. Además, en el Instituto Tecnológico del Azuay se analizan las implicaciones y dificultades de la utilización de ChatGPT, no solo en clases, si no para gestionar procesos de docentes, como planificación académica o registros de evaluación (Terreros- Pesantez et al., 2025). En conjunto, se demuestra la aceptación de la IA por parte de los profesores, aunque aún no se puede predecir las posibles consecuencias a largo plazo.

Por lo tanto, la investigación sobre chatbots basados en LLM resulta relevante por su capacidad para mejorar el acceso a la información académica en la educación superior. Esta necesidad se vuelve más evidente cuando la información institucional se almacena en repositorios que contienen numerosos documentos en formato PDF, a menudo extensos y heterogéneos, lo que dificulta la localización rápida de información pertinente. En este contexto, la optimización de la búsqueda en repositorios universitarios, en una era de constante desarrollo tecnológico, se vincula con el

principio de accesibilidad equitativa, entendido como la igualdad de oportunidades para acceder a los servicios institucionales (Universidad Católica de Cuenca, 2020).

En un estudio se evidencia que ayudar a usuarios a encontrar información relevante en bibliotecas universitarias es más difícil, por el crecimiento de recursos electrónicos (Dragon et al., 2025). En consecuencia, este trabajo desarrolla y evalúa un modelo inicial de asistente inteligente que permite consultar y gestionar documentos académicos en formato PDF, basado en IA y base de datos vectoriales.

El artículo se organiza de la siguiente manera: primero se describen los materiales y métodos empleados para el desarrollo y la evaluación del prototipo. Luego se presentan los resultados y su discusión. Finalmente, se exponen las conclusiones y los trabajos futuros.

Materiales y Métodos

El marco de trabajo Scrum guía la ejecución del proyecto mediante procesos definidos para planificar, desarrollar y validar incrementos funcionales (Schwaber & Sutherland, 2012). La gestión del trabajo se organizó en iteraciones tipo Sprint, con una duración base de dos semanas, que variaba cuando la complejidad de las tareas lo requerían. La organización de roles se aplicó de manera unipersonal porque la desarrolladora asumió los roles de Product Owner y Equipo de Desarrollo, encargándose de priorizar funcionalidades, planificar actividades e implementar los módulos del prototipo. El tutor académico mientras tanto fue el stakeholder principal, revisando los entregables al final de cada Sprint y proporcionando retroalimentación para ajustes técnicos. El proceso general se resume en la figura 1.

Figura 1: Metodología Scrum para el desarrollo del asistente inteligente

Fuente: Elaboración propia (2026)

DELIMITACIÓN DEL PROBLEMA

El primer paso fue delimitar la problemática del proyecto. En el ámbito académico encontrar información útil y rápido continúa siendo complicado, porque se encuentra repartida en distintos PDFs académicos. El problema crece cuando los documentos son extensos, porque

exige dedicar más tiempo en buscar información importante mediante palabras clave o por lecturas completas. Con el problema definido, se revisaron fuentes bibliográficas para seleccionar un LLM e IA necesario para definir el alcance del trabajo.

ANÁLISIS DE REQUERIMIENTOS

En esta etapa se definieron requerimientos funcionales (RF) y requerimientos no funcionales (RFN), los cuales orientan el desarrollo del prototipo del asistente inteligente. Los requerimientos funcionales delimitan las funcionalidades que el prototipo debe ofrecer, mientras que los no funcionales establecen parámetros de calidad y restricciones basándose en el alcance del prototipo.

Los requerimientos funcionales especificaron módulos esenciales del programa, como chats, documentos, consultas e interfaz. Además, se priorizó el flujo de carga y procesamiento de PDFs, la recuperación de fragmentos con FAISS, la construcción de contexto para el LLM y el registro del historial de chats. En cambio, los requerimientos no funcionales delimitaron que el asistente se ejecute en un entorno local, priorizando rendimiento, usabilidad y mantenibilidad por módulos.

La revisión de requerimientos fue al inicio de cada Sprint, por ende, después de cada retroalimentación se ajustaron prioridades y se añadieron mejoras futuras. A partir de esto se elaboró y gestionó el backlog en Jira, mediante historias de usuario y tareas priorizadas. Como consecuencia, el seguimiento de los requerimientos definidos y los incrementos entregados en cada iteración permaneció claro y ordenado.

Requerimientos principales:

- RF1: Cargar y almacenar documentos PDF.
- RF2: Extraer texto, fragmentar e indexar con embeddings.
- RF3: Recuperar fragmentos relevantes con RAG.
- RF4: Generar respuestas con un LLM mediante Ollama.
- RF5: Registrar y consultar historial de chats en MongoDB.
- RNF1: Ejecutar la inferencia en el mismo equipo, en un entorno local.

DISEÑO DE ARQUITECTURA

El diseño de arquitectura del asistente inteligente se orientó a servicios, el cual se compone en un cliente web y un backend. Además, el prototipo se organizó por capas para establecer responsabilidades claras entre los módulos. El diagrama de diseño se elaboró como soporte para su uso como referencia para la implementación incremental del prototipo (ver figura 2).

- Capa de Presentación: Interfaz web donde el usuario carga PDFs y consulta respuestas o historial de chats.
- Capa de Servicios: Backend con endpoints REST que gestiona la ingesta de PDFs, las consultas y la comunicación entre módulos.
- Capa de Procesamiento y Recuperación: Módulos que procesan el PDF (lectura y extracción de texto), fragmentan el contenido (chunking), generan embeddings y buscan en FAISS los fragmentos más relevantes para construir el contexto de respuesta.
- Capa de Datos: MongoDB guarda el historial de chats y metadatos de documentos para permitir retomar conversaciones.
- Capa de IA: Ollama ejecuta el modelo de lenguaje y genera la respuesta final a partir del contexto recuperado.

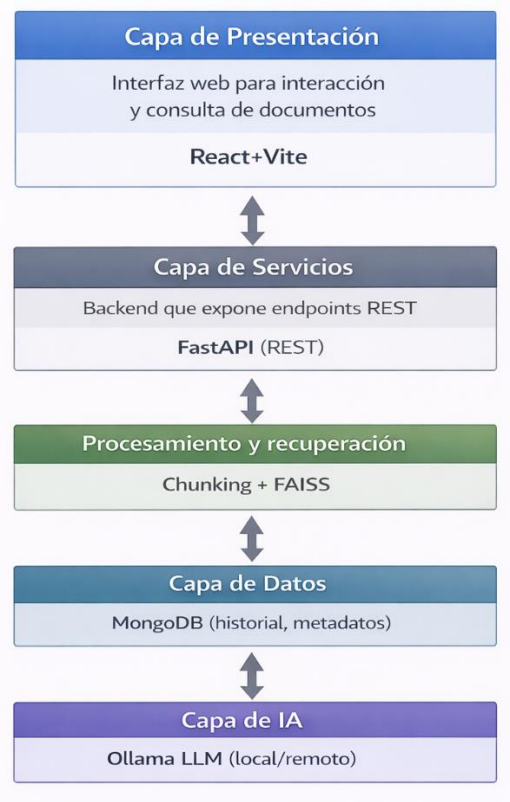

Figura 2: Diagrama del diseño de arquitectura del asistente inteligente de documentos

Fuente: Elaboración propia (2026)

Una solución basada en IA depende de la calidad, consistencia e integridad de los datos, así como de la infraestructura necesaria para probarla, entrenarla y desplegarla (Campoverde-Molina & Luján-Mora, 2025). Por ello, en esta etapa del trabajo se estableció una estructura mínima de datos, la cual garantiza la vinculación entre documentos, fragmentos recuperados y respuestas generadas. A continuación, se describe el flujo de trabajo de ingesta, recuperación semántica y generación aumentada con recuperación (Retrieval-Augmented Generation, RAG) del asistente inteligente.

- PDF (Entrada): El prototipo recibe el archivo PDF, valida su formato y lo registra como documento para iniciar la ingesta.
- Extracción y Chunking: El texto se extrae del PDF y se divide en fragmentos manejables para facilitar la indexación y la búsqueda.
- Embeddings: Cada fragmento se transforma en un vector numérico que representa su significado y permite comparar contenido por similitud.
- FAISS (Indexación): Los vectores se almacenan en un índice vectorial que permite ubicar rápidamente los fragmentos más cercanos a una consulta.
- Recuperación y RAG: Los fragmentos más relevantes se recuperan ante una pregunta y se usan como contexto para que el LLM genere una respuesta sustentada.
- Registro (MongoDB): Se almacena el historial de la consulta y los metadatos del documento.

IMPLEMENTACIÓN DEL BACKEND

Para el desarrollo del backend del prototipo se utilizó Python con FastAPI para exponer la API y Uvicorn como servidor. El flujo RAG se integró con LangChain y PyPDFLoader para extraer el texto de los PDFs y segmentarlos en fragmentos. Los embeddings se generaron con Sentence-Transformers, lo que permitió la indexación y búsqueda semántica mediante FAISS. La generación de respuestas se realizó con Gemma 3 (27B) ejecutado localmente a través de Ollama. La persistencia de datos se gestionó con MongoDB. En síntesis, en esta fase se implementaron los módulos y servicios necesarios para que al final de cada Sprint se entregara un incremento funcional.

IMPLEMENTACIÓN DEL FRONTEND

El cliente web consume los endpoints expuestos por el backend para construir las interfaces y los flujos de interacción del prototipo. Por tal motivo, se utilizó React como biblioteca de interfaz de usuario y Vite como herramienta de construcción. El diseño responsivo se implementó con CSS personalizado y Flexbox, mientras que la comunicación HTTP con el backend se realizó mediante Fetch API. Finalmente, npm se empleó para la administración de paquetes y dependencias del proyecto. En consecuencia, la implementación del frontend permitió proseguir con la validación y evaluación del prototipo.

EVALUACIÓN DEL PROTOTIPO

La validación del asistente inteligente se apoyó en pruebas end-to-end (E2E) y pruebas de uso. Con las pruebas E2E se comprobó, mediante

escenarios preestablecidos, el flujo completo del prototipo. Por ejemplo, se validó el flujo de la carga de un documento PDF o la carga de varios documentos mediante "Conectar carpeta". También, se aplicaron pruebas de uso con cuatro participantes. Su percepción de uso se recolecta mediante una encuesta tipo Likert (1–5), ya que esta escala permite cuantificar actitudes y percepciones de los participantes (Joshi et al., 2015). Dicha encuesta se enfocó en facilidad, claridad, utilidad del historial y comprensión de evidencias del prototipo.

Resultados y Discusión

En esta sección se presentan los resultados del desarrollo y evaluación del asistente inteligente. El cual fue propuesto para apoyar la consulta y gestión de PDFs académicos con IA y bases de datos vectoriales.

DESARROLLO DEL PROTOTIPO

Sprint 1: Estructura del backend

En este Sprint se levantó la API en FastAPI y se dejó operativa en un entorno local. También se definieron modelos Pydantic para estandarizar la estructura de entrada y salida de datos de la API. El resultado fue una base estable para crecer por módulos.

Retroalimentación del Sprint: Se acordó priorizar primero estabilidad y claridad de endpoints antes de agregar funciones avanzadas.

Sprint 2: Gestión de chats e historial

En este Sprint se desarrollaron endpoints para crear, listar, consultar y eliminar chats. Además, el historial se configuró para que se almacene en MongoDB con metadatos. El propósito fue mantener una organización modular basada en servicios.

Retroalimentación del Sprint: Se recomendó trabajar con MongoDB para asegurar persistencia real del historial y trazabilidad de interacciones.

Sprint 3: Ingesta documental y preparación para indexación

En este Sprint se habilitó la carga de PDFs por chat. En primer lugar, se almacena el archivo y se valida su extensión (.pdf) antes de la extracción. Después, el contenido del documento se obtiene con PyPDFLoader y se fragmenta con CharacterTextSplitter. El chunking se parametriza mediante la definición del tamaño de fragmento y solapamiento (overlap) en la configuración del prototipo. En este prototipo se definió el tamaño de chunk con 1000 tokens y overlap con 150 tokens. Además, la normalización de texto se realiza mediante la aplicación de expresiones regulares y también se detecta encabezados repetitivos.

Retroalimentación del Sprint: Se sugirió incorporar límites para reforzar el control de carga con el propósito de mejorar la estabilidad.

Sprint 4: Indexación vectorial, recuperación semántica y RAG con persistencia

La generación de embeddings se integró con HuggingFaceEmbeddings, habilitando búsqueda por similitud. Se construyeron índices FAISS por documento, guardados en un vector store por chat y almacenados con metadatos del procesamiento. Además, se desarrolló una funcionalidad que permite agregar una carpeta común desde configuración. Esta carpeta actúa como una base opcional con varios documentos PDF, disponible para cualquier chat creado. En el caso que el usuario carga otros PDF, estos se priorizan, sin descartar el contexto aportado por la carpeta común.

También, habilitamos el RAG para recuperar fragmentos relevantes y poder consultarlos al LLM con Ollama. Por ello, incorporamos un timeout configurable y excepciones en la llamada al LLM. Además, utilizamos una configuración conservadora para reducir alucinaciones, con temperatura baja (0.1) y un límite de 800 tokens. Como resultado, VectorStoreManager administra los índices vectoriales, ChatManager coordina el flujo del chat, y ChatDocumentsService selecciona los fragmentos de documentos usados como evidencia para responder.

Retroalimentación del Sprint: Se solicitó optimizar el chunking para reducir tiempos y aumentar la calidad de la recuperación.

Sprint 5: Conexión de carpeta y preselección documental para consultas

Incorporamos la opción Conectar carpeta para asociar a un chat hasta 20 PDFs extensos, de más de 100 páginas y solo texto. Los documentos se cargan desde una ruta local, lo cual elimina la necesidad de subirlos uno a uno en la interfaz. A partir de los índices FAISS por documento del anterior sprint, se implementó un mejor proceso de recuperación de información para responder preguntas. Primero se eligen los documentos más probables para la consulta y, después, se extraen los fragmentos más relevantes dentro de esos candidatos. Luego se combinan solo los índices de los documentos seleccionados, para acelerar la búsqueda y reducir el consumo de recursos.

Retroalimentación del Sprint: Se sugirió actualizar el registro cuando se agreguen, eliminen o modifiquen archivos, para evitar consultas con información antigua.

Sprint 6: Esquema de interfaz básico

En este Sprint desarrollamos una primera propuesta de interfaz para delimitar su estructura. Además, definimos la disposición base con barra lateral y área principal. En resumen, buscamos validar jerarquía visual y distribución de componentes. La figura 3 evidencia este primer acercamiento de baja fidelidad.

Retroalimentación del Sprint: Se propuso desarrollar un diseño más sobrio y profesional.

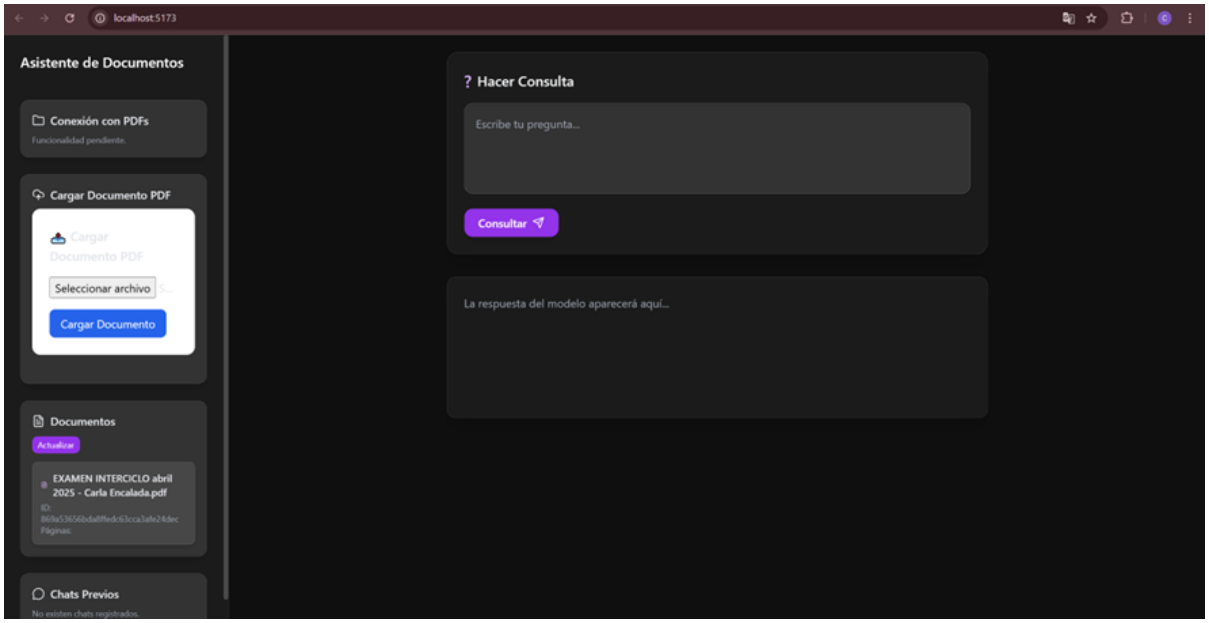

Figura 3: Interfaz inicial del asistente para validar distribución de componentes

Fuente: Elaboración propia (2026)

Sprint 7: Rediseño de interfaz e integración con el backend

Se implementó la interfaz final y se integró con la API mediante la URL definida en el entorno. Asimismo, se habilitó la gestión de sesiones, consultas y carga de PDFs asociados al chat, incluyendo ingesta por carpeta. La figura 4 presenta el resultado.

Retroalimentación del Sprint: Se recomendó simplificar componentes y mantener consistencia en tipografías y espaciados.

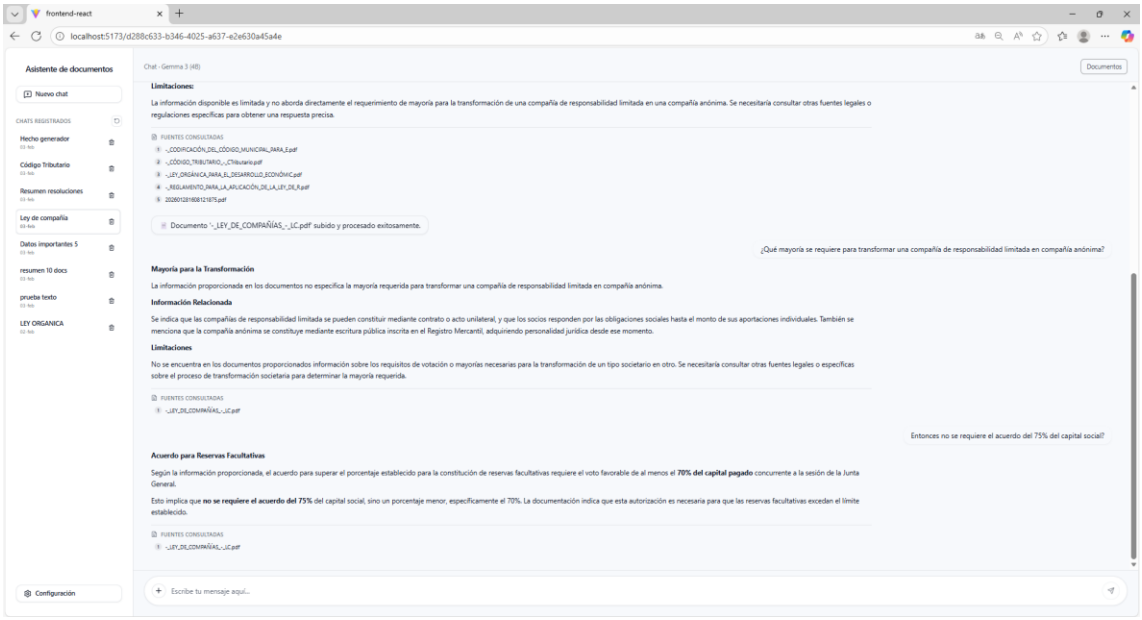

Figura 4: Interfaz final del asistente inteligente

Fuente: Elaboración propia (2026)

EVALUACIÓN DEL PROTOTIPO

Configuración del entorno de evaluación

Las pruebas se ejecutaron en un entorno local, específicamente en una computadora de escritorio (Desktop) de la Universidad Católica de Cuenca (UCACUE). El equipo utilizado fue un ASUS con procesador Intel Core i9-13900K (24C/32T), 64 GB de RAM y arquitectura x64. En la tabla 1 se resume la configuración del entorno y los cambios principales entre la versión inicial (V1) y la versión actual (V2), incluyendo el modelo LLM (Ollama) y el modelo de embeddings utilizados.

Tabla 1. Configuración del entorno (misma desktop)

Entorno	Modelo LLM (Ollama)	Modelo de embeddings	Observaciones
Desktop (V1 - versión inicial)	gemma3:4b	all-MiniLM-L6-v2	Configuración más conservadora para evitar saturación. Menor costo computacional.
Desktop (V2 - versión actual)	gemma3:27b	paraphrase-multilingual-MiniLM-L12-v2	Configuración más robusta pero más pesada. Mayor costo de inferencia y embeddings.

Fuente: elaboración propia

La comparación V1 vs V2 corresponde a cambios de configuración. En V2 se incrementa el tamaño del modelo LLM por defecto de gemma3:1b a gemma3:27b. Además, se utiliza un modelo de embeddings más pesado de all-MiniLM-L6-v2 a paraphrase-multilingual-MiniLM-L12-v2, lo que incrementa el costo computacional de la generación de embeddings y la respuesta del asistente.

Pruebas de rendimiento: ingesta e indexación de documentos

La validación end-to-end confirmó el funcionamiento integral del prototipo en tres escenarios. El primero (E2E-1) corresponde a la creación de chat con carga de un PDF y su consulta. El segundo y tercero escenario (E2E-2 y E2E-3) corresponden a la creación de chat con conexión de carpeta con 10 o 20 documentos, indexación y consulta posterior. En los tres casos se verificó persistencia del

historial y recuperación de evidencias para fundamentar la respuesta.

A partir de ello, se reporta el tiempo total de ingesta e indexación T(ing) como tiempo real (wall-clock), desde la extracción del PDF hasta que el contenido queda disponible para consulta. La comparación entre versiones se expresa mediante speedup (S) y variación porcentual (%Δ), calculados sobre T(ing) para los escenarios evaluados. Las ecuaciones se presentan a continuación:

1. Speedup (S)

Es "cuántas veces" es más rápido un entorno respecto al otro. Si comparas la configuración inicial vs la actual:

$$S = \frac{T_{V1}}{T_{V2}} \quad (1)$$

Interpretación rápida:

- S = 1 → igual de rápido
- S = 2 → Versión actual es 2× más rápido
- S = 4.5 → Versión actual es 4.5× más rápido

2. Variación porcentual (%Δ)

Es el porcentaje de reducción del tiempo al pasar de la configuración inicial a la actual.

$$\%\boldsymbol{\Delta} = \frac{\boldsymbol{T_{V1} - T_{V2}}}{\boldsymbol{T_{V1}}} \times \mathbf{100} \quad (2)$$

Interpretación rápida:

- %Δ = 0% → no hubo mejora
- %Δ = 50% → el tiempo bajó a la mitad
- %Δ = 80% → el tiempo bajó muchísimo (solo queda 20% del original)

Tabla 2. Pruebas de rendimiento: ingesta e indexación de documentos

Escenario	Carga	T(V1)	T(V2)	Speedup (S)	Variación porcentual (%Δ)
E2E-1	1 PDF extenso (100 págs)	14.80 s	19.84 s	0.75×	-34.1%
E2E-2	Conectar carpeta (10 PDFs)	149.81 s	260.13 s	0.58×	-73.6%
E2E-3	Conectar carpeta (20 PDFs)	156.52 s	277.70 s	0,56×	-77.4%

Fuente: elaboración propia

Los resultados evidencian que en la versión V2 se incrementa el tiempo de ingesta e indexación respecto a V1. Para un PDF, el tiempo pasa de 00:14.80 a 00:19.84 (S=0.75×); para 10 PDFs, de 2:29.81 a 4:20.13 (S=0.58×); y para 20 PDFs, de 2:36.52 a 4:37.70 (S=0.56×). En consecuencia, la configuración de V2 implica mayor costo computacional, especialmente en cargas masivas.

Sin embargo, este aumento se asocia a que V2 genera más fragmentos y embeddings, por mayor procesamiento. Lo cual incrementa el tiempo de ingesta, pero mejora la recuperación semántica al disponer de fragmentos más informativos, permitiendo producir respuestas mejor fundamentadas.

Pruebas de uso

Una evaluación de uso se aplicó con cuatro participantes (n = 4) mediante una encuesta tipo Likert organizada en cinco criterios (C1–C5). La Tabla 3 presenta los promedios (M) y la desviación estándar (DE) como referencia descriptiva, considerando el tamaño de la muestra. En términos generales, la percepción fue alta, con valores promedio entre 4.25 y 5.00. El criterio C4 (comprensión de evidencias/contexto) obtuvo la puntuación máxima (M = 5.00; DE = 0.00), lo que refleja consenso total entre los participantes. Los criterios C1–C3 también recibieron valoraciones favorables. Sin embargo, C3 mostró mayor variabilidad (DE = 1.00), lo que indica diferencias individuales en cómo se percibe la utilidad o continuidad del historial. El

promedio más bajo correspondió a C5 (experiencia general) (M = 4.25; DE = 0.96), lo que indica posibles mejoras en la interacción, como una comunicación más clara del estado del prototipo, como mensajes de progreso.

Tabla 3. Resultados de percepción por criterio (escala 1–5, n=4)

Criterio	Descripción	M	DE
C1	Facilidad de iniciar un chat y cargar documentos	4.75	0.50
C2	Claridad y utilidad de la respuesta del asistente	4.50	0.58
C3	Utilidad del historial y continuidad de conversación	4.50	1.00
C4	Comprensión de evidencias/contexto usado por el asistente	5.00	0.00
C5	Experiencia general de uso	4.25	0.96

Fuente: elaboración propia

Kwon et al. (2025) describe un escenario multi-documento que se relaciona con dynamic-selection-based retrieval-augmented generation (DS-RAG). Además, cuenta con la capacidad de mejorar la respuesta a preguntas que dependen de varias fuentes y también controla el tamaño del contexto (Kwon et al., 2025). En comparación con el caso del asistente inteligente, cuando se carga una carpeta con varios documentos PDFs realiza algo similar, pero de forma local. Ya que, primero prioriza los documentos candidatos y luego los fragmentos relevantes para construir el contexto de respuesta.

Aytar et al. (2025) en su trabajo incorporan un pipeline multietapa que integra GROBID, el cual es una gran herramienta para extraer y estructurar contenido antes de ser indexado. Además, RAGAS se utiliza como base para evaluar su sistema de manera automática (Aytar et al., 2025). En cambio, como se mencionó antes, el programa descrito en este trabajo prioriza la estabilidad del flujo y su eficiencia de procesamiento en ejecución local, sin aún integrar esas herramientas.

Conclusiones

Este trabajo demuestra que un asistente inteligente de consulta documental basado en RAG puede operar de forma consistente en un entorno local. El prototipo integra ingesta de PDFs, indexación vectorial, recuperación de fragmentos y generación de respuestas sustentadas en los documentos que se cargaron previamente. La validación E2E confirmó la persistencia del historial a lo largo del flujo del procesamiento de documentos. Su funcionamiento se verificó en dos escenarios, mediante la carga individual de un PDF o la conexión de una carpeta con múltiples documentos. Los resultados refuerzan la utilidad del prototipo cuando la información académica se encuentra en varios archivos.

En cuanto al rendimiento, la versión actual incrementó el tiempo de ingesta e indexación frente a la versión inicial. Este aumento se explica porque la segunda versión genera más embeddings, lo que eleva el costo de procesamiento. Sin embargo, esta decisión mejora la recuperación de evidencias y, en consecuencia, la calidad de las respuestas. En la evaluación de uso, la percepción general fue positiva, y la comprensión de respuesta generada por el programa destacó como el criterio mejor valorado. En conjunto, los resultados confirman que un asistente local mejora la consulta de PDFs mediante respuestas contextualizadas y referenciadas, sin dependencia obligatoria de servicios externos.

Como trabajos futuros, se recomienda ampliar la evaluación del prototipo en un contexto institucional, con más usuarios, perfiles diversos y escenarios reales. También se deben incorporar controles de seguridad alineados con las políticas universitarias. Para fortalecer la evidencia científica, conviene aplicar métricas formales de calidad RAG. Estas métricas deben cubrir precisión, cobertura de evidencia, fidelidad al contexto y consistencia de respuestas. En lo técnico, se sugiere optimizar segmentación, solapamiento y normalización del texto. Por último, es recomendable mejorar la interfaz de usuario con manejo explícito de estados y control de errores más robustos, con

el objetivo de estabilizar el historial y las consultas consecutivas.

Agradecimientos

Este trabajo ha sido apoyado por el Centro de Ingeniería de Software de la Universidad Católica de Cuenca.

Referencias Bibliográficas

Anchapaxi-Díaz, C. L., Pinenla-Palaguaray, Y. M., Caiza-Olapincha, S. P., Parra-Taboada, I. A., Abad-Guamán, M. A., & Viñamagua-Arias, B. V. (2024). Uso de Chatbots educativos y su impacto en el aprendizaje autónomo en bachillerato. *Revista Científica Retos de la Ciencia*, *1*(4), 200-214. https://doi.org/10.53877/rc.8.19e.202409.16

Aytar, A. Y., Kaya, K., & Kılıç, K. (2025). A synergistic multi-stage RAG architecture for boosting context relevance in data science literature. *Natural Language Processing Journal*, *13*, 100179. https://doi.org/10.1016/j.nlp.2025.100179

Bustamante Bula, R., & Camacho Bonilla, A. (2024). Inteligencia artificial (IA) en las escuelas: Una revisión sistemática (2019-2023). *Enunciación*, *29*(1), 62-82. https://doi.org/10.14483/22486798.22039

Campoverde-Molina, M., & Luján-Mora, S. (2026). Artificial intelligence in web accessibility: A systematic mapping study. *Computer Standards & Interfaces*, *96*, 104055. https://doi.org/10.1016/j.csi.2025.104055

Campuzano-Vásquez, J., Murillo-Guevara, N. N., & Sarango-Pintado, D. B. (2025). Uso de la inteligencia artificial en la educación superior: Estudio de caso Universidad Técnica de Machala. *INNOVA Research Journal*, *10*(2), 24-45. https://doi.org/10.33890/innova.v10.n2.2025.2754

Carvallo, J. P., & Erazo-Garzón, L. (2023). On the Use of ChatGPT to Support Requirements Engineering Teaching and Learning Process. En S. Berrezueta (Ed.), *Proceedings of the 18th Latin American Conference on Learning Technologies (LACLO 2023)* (pp. 328-342). Springer Nature Singapore. https://doi.org/10.1007/978-981-99-7353-8_25

Dragon, P. M., Mayo, J. L., Stocks, A. C., & Tatterson, R. (2025). Enhancing library discovery: An approach to understanding user access to electronic resources. *The Journal of Academic Librarianship*, *51*(4), 103064. https://doi.org/10.1016/j.acalib.2025.103064

Joshi, A., Kale, S., Chandel, S., & Pal, D. (2015). Likert Scale: Explored and Explained. *British Journal of Applied Science & Technology*, *7*(4), 396-403. https://doi.org/10.9734/BJAST/2015/14975

Khalifa, M., & Albadawy, M. (2024). Using artificial intelligence in academic writing and research: An essential productivity tool. *Computer Methods and Programs in Biomedicine Update*, *5*, 100145. https://doi.org/10.1016/j.cmpbup.2024.100145

Kwon, M., Bang, J., Hwang, S., Jang, J., & Lee, W. (2025). A Dynamic-Selection-Based, Retrieval-Augmented Generation Framework: Enhancing Multi-Document Question-Answering for Commercial Applications. *Electronics*, *14*(4), 659. https://doi.org/10.3390/electronics14040659

McGrath, C., Farazouli, A., & Cerratto-Pargman, T. (2025). Generative AI chatbots in higher education: A review of an emerging research area. *Higher Education*, *89*(6), 1533-1549. https://doi.org/10.1007/s10734-024-01288-w

Montoya Naguas, F. E., Lalangui Flores, C. E., Redrován Castillo, F. F., & Jumbo Castillo, F. A. (2025). Desarrollo de un Chatbot con Inteligencia Artificial para orientación vocacional según el perfil competencial de estudiantes de tercero bachillerato. *Informática y Sistemas*, *9*(2), 184. https://doi.org/10.33936/isrtic.v9i2.7907

Ortiz Vázquez, C. M., & Marín Guamán, M. A. (2025). Uso de la Inteligencia Artificial por Docentes de Odontología en la Universidad Católica de Cuenca: Use of Artificial Intelligence by Dentistry Faculty at the Catholic University of Cuenca. *Revista Scientific*, *10*(37), 201-219. https://doi.org/10.29394/Scientific.issn.2542-2987.2025.10.37.10.201-219

Schwaber, K., & Sutherland, J. (Eds.). (2012). The Scrum Guide. En *Software in 30 Days* (1.ª ed., pp. 133-152). Wiley. https://doi.org/10.1002/9781119203278.app2

Sigüenza Orellana, J., Andrade Cordero, C., & Chitacapa Espinoza, J. (2024). Validación del cuestionario para docentes: Percepción sobre el uso de ChatGPT en la educación superior. *Revista Andina de Educación*, *8*(1), 000816. https://doi.org/10.32719/26312816.2024.8.1.6

Terreros- Pesantez, D. F., Vásquez- Erazo, E. J., & Ramon- Poma, G. M. (2025). La inteligencia artificial en la gestión administrativa docente del Instituto Tecnológico del Azuay, Cuenca, Ecuador 2025. *Resistances. Journal of the Philosophy of History*, *6*(12), e250192. https://doi.org/10.46652/resistances.v6i12.192

Tramallino, C. P., & Marize Zeni, A. (2024). Avances y discusiones sobre el uso de inteligencia artificial (IA) en educación. *Educación*, *33*(64), 29-54. https://doi.org/10.18800/educacion.202401.M002

Universidad Católica de Cuenca. (2020, noviembre). *Modelo Educativo—Pedagógico "Kunanmanta"*. https://documentacion.ucacue.edu.ec/items/show/2700