PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

ESCUELA HÁBITAT, INFRAESTRUCTURA Y CREATIVIDAD

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

CHATBOT PERSONALIZADO MEDIANTE EL USO DE VOZ PARA LA ATENCIÓN GENERAL DE ESTUDIANTES DE PRIMER NIVEL EN LA CARRERA DE INGENIERÍA EN LA PUCE-I

NICK BRYAN LÓPEZ REINA

TUTOR/A: DULCE MILAGRO RIVERO ALBARRAN

IBARRA – ECUADOR

JULIO, 2026

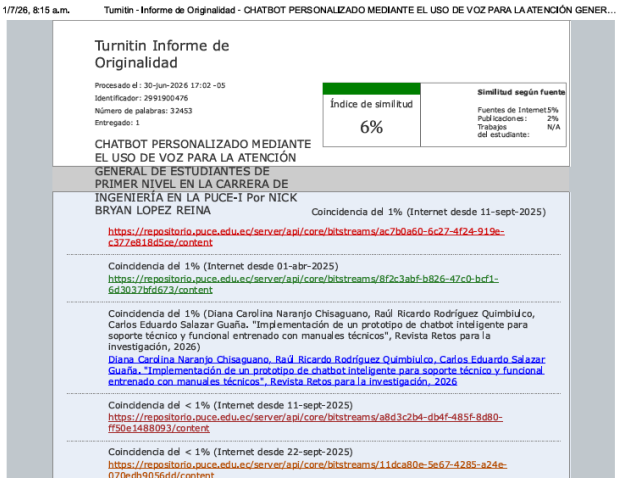
Ibarra, 10 de julio del 2026

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de **INTEGRACIÓN CURRICULAR** titulado: **CHATBOT PERSONALIZADO MEDIANTE EL USO DE VOZ PARA LA ATENCIÓN GENERAL DE ESTUDIANTES DE PRIMER NIVEL EN LA CARRERA DE INGENIERÍA EN LA PUCE-I**, presentado por el estudiante **LÓPEZ REINA NICK BRYAN** con cédula de ciudadanía N° **0850908161**, para obtener el Título de **INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.


1/7/26, 8:15 a.m. Turnitin - Informe de Originalidad - CHATBOT PERSONALIZADO MEDIANTE EL USO DE VOZ PARA LA ATENCIÓN GENER...

Turnitin Informe de Originalidad

Procesado el : 30-jun-2026 17:02 -05
Identificador: 2991900476
Número de palabras: 32453
Entregado: 1

Índice de similitud 6%

Similitud según fuente
Fuentes de Internet 5%
Publicaciones: 2%
Trabajos del estudiante: N/A

CHATBOT PERSONALIZADO MEDIANTE EL USO DE VOZ PARA LA ATENCIÓN GENERAL DE ESTUDIANTES DE PRIMER NIVEL EN LA CARRERA DE INGENIERÍA EN LA PUCE-I Por NICK BRYAN LOPEZ REINA

Coincidencia del 1% (Internet desde 11-sept-2025)
https://repositorio.puce.edu.ec/server/api/core/bitstreams/ac7b0a60-6c27-4f24-919e-c377e818d5ce/content

Coincidencia del 1% (Internet desde 01-abr-2025)
https://repositorio.puce.edu.ec/server/api/core/bitstreams/8f2c3abf-b826-47c0-bcf1-6d3037bfd673/content

Coincidencia del 1% (Diana Carolina Naranjo Chisaguano, Raúl Ricardo Rodríguez Quimbiulco, Carlos Eduardo Salazar Guaña. "Implementación de un prototipo de chatbot inteligente para soporte técnico y funcional entrenado con manuales técnicos", Revista Retos para la investigación, 2026)
Diana Carolina Naranjo Chisaguano, Raúl Ricardo Rodríguez Quimbiulco, Carlos Eduardo Salazar Guaña. "Implementación de un prototipo de chatbot inteligente para soporte técnico y funcional entrenado con manuales técnicos", Revista Retos para la investigación, 2026

Coincidencia del < 1% (Internet desde 11-sept-2025)
https://repositorio.puce.edu.ec/server/api/core/bitstreams/a8d3c2b4-db4f-485f-8d80-ff50e1488093/content

Coincidencia del < 1% (Internet desde 22-sept-2025)
https://repositorio.puce.edu.ec/server/api/core/bitstreams/11dca80e-5e67-4285-a24e-

(f): Dulce Milagro Rivero Albarrán — Firmado digitalmente por Dulce Milagro Rivero Albarrán Fecha: 2026.07.22 12:13:46 -05'00'

PhD. DULCE MILAGRO RIVERO ALBARRAN

TUTOR/A DE TRABAJO

C.C.: **1757608961**

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El tribunal examinador, aprueba el presente trabajo en nombre de la Pontificia Universidad Católica del Ecuador Ibarra:

Dulce Milagro Rivero Albarrán
Firmado digitalmente por Dulce Milagro Rivero Albarrán
Fecha: 2026.07.22 12:13:46 -05'00'

(f): .....

PhD. **DULCE MILAGRO RIVERO ALBARRAN**

C.C.: **1757608961**

Laura Guerra Torrealba
Firmado digitalmente por Laura Guerra Torrealba
Fecha: 2026.07.22 15:18:55 -05'00'

(f):.....

PhD. **LAURA ROSA GUERRA TORREALBA**

C.C.: **1757842784**

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
JOSE LUIS IBARRA ESTEVEZ

(f):.....

Mgts. **JOSÉ LUIS IBARRA ESTÉVEZ**

C.C.: **1002640728**

ACTA DE CESIÓN DE DERECHOS

Yo, **LÓPEZ REINA NICK BRYAN**, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: “Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilizaciones de sus obras o prestaciones a título gratuito y oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia”.

Ibarra, 10 de julio del 2026

(f): Validar únicamente en FirmaEC.
Firmado electrónicamente por:
NICK BRYAN LOPEZ REINA

LÓPEZ REINA NICK BRYAN

C.C.: **0850908161**

AUTORÍA

Yo, **LÓPEZ REINA NICK BRYAN**, portador de la cédula de ciudadanía N° **0850908161**, declaro que el presente trabajo de investigación es de total responsabilidad del autor, y eximo expresamente a la Pontificia Universidad Católica del Ecuador Ibarra de posibles reclamos o acciones legales.

(f): 

……..

LÓPEZ REINA NICK BRYAN

C.C.: **0850908161**

DEDICATORIA

Este logro representa mucho más que un objetivo alcanzado; es el resultado del amor, el apoyo y la guía de las personas más importantes en mi vida. A mis abuelos, mi abuela y mi abuelo, gracias por ser un ejemplo de esfuerzo, valores y sabiduría. Sus enseñanzas han sido la base que me ha permitido llegar hasta aquí. A mi familia, especialmente a mi mujer y a mi hijo, gracias por ser mi fortaleza, mi motivación y mi razón para seguir adelante. Su amor y apoyo incondicional han sido fundamentales en cada paso de este camino.

Este logro es tan mío como de ustedes.

Con todo mi amor y eterna gratitud.

AGRADECIMIENTOS

Agradezco a Dios y a mi familia por el apoyo brindado durante todo este proceso académico. De manera especial, expreso mi gratitud a mi tutora, PhD. **DULCE MILAGRO RIVERO ALBARRAN**, por su orientación, acompañamiento y observaciones durante el desarrollo del presente trabajo de integración curricular. Asimismo, agradezco a la Pontificia Universidad Católica del Ecuador Sede Ibarra por la formación recibida y por permitirme fortalecer mis conocimientos en el área de Tecnologías de la Información.

ÍNDICE DE CONTENIDOS

Contenido

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo tuvo como objetivo desarrollar un prototipo de chatbot académico personalizado con reconocimiento de voz para la atención de consultas generales y académicas de estudiantes de primer nivel de la carrera de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra. La investigación respondió a la necesidad de mejorar el acceso oportuno a información institucional y académica, especialmente en procesos relacionados con horarios, asignaturas, aulas, calificaciones, actividades y orientación general dentro del entorno universitario.

El sistema fue desarrollado como una investigación aplicada con enfoque mixto y metodología ágil *Scrum* adaptada a un contexto académico. La solución se implementó mediante una arquitectura web compuesta por un *frontend* en React, un *backend* en Django REST Framework y una base de datos PostgreSQL destinada a gestionar información académica simulada de estudiantes de primer nivel. Para las consultas institucionales se implementó una arquitectura *Retrieval-Augmented Generation* (RAG), apoyada en documentos curados, *embeddings*, FAISS y un modelo de lenguaje de gran escala. Además, se incorporó la interacción por voz mediante servicios de transcripción y síntesis de audio, junto con mecanismos de autenticación segura, verificación OTP, control de acceso y registros de eventos del sistema.

Los resultados obtenidos evidenciaron que el prototipo permitió diferenciar consultas públicas de consultas personalizadas, proteger la información académica del estudiante autenticado y ofrecer una experiencia conversacional integrada mediante texto y voz. La validación del sistema se realizó en un entorno controlado, utilizando pruebas funcionales, de usabilidad, seguridad, registro de eventos e interacción por voz. En conclusión, el chatbot desarrollado constituye una solución tecnológica viable para apoyar la orientación académica inicial de estudiantes de primer nivel, reduciendo tiempos de búsqueda de información y centralizando el acceso a servicios académicos relevantes.

Palabras clave: chatbot académico, inteligencia artificial, RAG, reconocimiento de voz, Django, React, PostgreSQL, registro de eventos.

ABSTRACT

This study aimed to develop a personalized academic chatbot prototype with voice recognition to support general and academic inquiries from first-level Engineering students at Pontificia Universidad Católica del Ecuador, Ibarra Campus. The project addressed the need to improve timely access to institutional and academic information, particularly regarding schedules, courses, classrooms, grades, academic activities, and general student orientation within the university environment.

The system was developed as applied research with a mixed-method approach and an agile *Scrum* methodology adapted to an academic context. The solution was implemented through a web architecture composed of a React *frontend*, a Django REST Framework *backend*, and a PostgreSQL database for managing simulated academic information from first-level students. Institutional queries were handled through a *Retrieval-Augmented Generation* (RAG) architecture, supported by curated documents, *embeddings*, FAISS, and a large language model. Additionally, voice interaction was integrated through speech-to-text and text-to-speech services, along with secure authentication, OTP verification, access control, and system event logging mechanisms.

The results showed that the prototype was able to distinguish between public institutional queries and personalized academic queries, protect the academic information of authenticated students, and provide an integrated conversational experience through both text and voice. The system was validated in a controlled environment through functional, usability, security, event logging, and voice interaction tests. In conclusion, the developed chatbot represents a viable technological solution to support the initial academic orientation of first-level students by reducing information search time and centralizing access to relevant academic services.

Keywords: academic chatbot, artificial intelligence, RAG, voice recognition, Django, React, PostgreSQL, event logging.

INTRODUCCIÓN

La transformación digital ha impulsado el uso de tecnologías inteligentes en la educación superior, especialmente en los procesos de atención académica, orientación estudiantil y acceso a información institucional. En este contexto, la inteligencia artificial ha permitido desarrollar asistentes conversacionales capaces de responder consultas frecuentes, reducir tiempos de búsqueda y apoyar al estudiante durante su proceso de adaptación universitaria. Por esta razón, los chatbots representan una alternativa tecnológica viable para fortalecer la comunicación entre la institución y los estudiantes.

El ingreso a la educación superior representa una etapa de adaptación que implica cambios académicos, administrativos y sociales, especialmente para los estudiantes de primer nivel en la carrera de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCE-I). Durante este periodo, los estudiantes requieren acceso constante a información básica como horarios, asignaturas, ubicación de aulas y datos relacionados con su planificación académica. La ausencia de mecanismos ágiles para acceder a esta información puede generar desorientación, afectar la organización personal y limitar el adecuado desarrollo de sus actividades académicas.

Los mecanismos tradicionales de atención, como la atención presencial o el uso de correos electrónicos, presentan limitaciones cuando los estudiantes requieren información fuera del horario laboral, debido a que dependen de la disponibilidad del personal administrativo y de procesos de seguimiento posterior. Estas restricciones dificultan la respuesta oportuna a preguntas frecuentes, incrementan la carga operativa del personal y reducen la eficiencia de los servicios institucionales. Ante este problema, el uso de soluciones tecnológicas basadas en IA permite una gestión más eficiente en términos de consultas, posibilitando un acceso rápido a datos estructurados (Oliveira & Matos, 2023).

Los avances recientes en inteligencia artificial han permitido el desarrollo de asistentes conversacionales capaces de comprender consultas formuladas en lenguaje natural y generar respuestas contextualizadas. No obstante, la utilización de modelos generativos en entornos académicos requiere mecanismos que permitan fundamentar las respuestas en información verificable. Por esta razón, la presente investigación incorporó una arquitectura *Retrieval-Augmented Generation* (RAG), orientada a recuperar información institucional relevante antes de generar cada respuesta.

La incorporación del reconocimiento de voz amplió las formas de interacción disponibles en el chatbot, permitiendo que los estudiantes formularan consultas habladas además de utilizar la entrada por texto. En este proyecto, la voz se implementó como un mecanismo complementario de accesibilidad y usabilidad, sin reemplazar la interacción escrita. De esta manera, el sistema ofreció una alternativa adicional para acceder a información académica e institucional dentro del entorno universitario.

A partir de lo anterior, se planteó el desarrollo de un prototipo de chatbot académico personalizado orientado a estudiantes de primer nivel de la carrera de Ingeniería en la PUCE-I. El sistema permitió gestionar consultas institucionales de carácter público y consultas académicas personalizadas, diferenciando la información general disponible para visitantes de los datos privados asociados al estudiante autenticado. Para ello, se integró una base de conocimiento institucional mediante arquitectura RAG y una base de datos académica estructurada en PostgreSQL, lo que permitió entregar respuestas contextualizadas sin exponer información personal en el modo público.

El chatbot desarrollado proporcionó respuestas relacionadas con actividades académicas cotidianas, tales como la consulta de horarios, asignaturas, calificaciones, actividades pendientes, notificaciones y ubicación de aulas. Estas respuestas se generaron según el perfil académico del estudiante autenticado, mientras que las consultas institucionales generales estuvieron disponibles desde la página pública del sistema. La personalización se sustentó en servicios *backend* desarrollados en Django REST Framework, una interfaz web implementada en React y mecanismos de comunicación mediante APIs REST, manteniendo como flujo principal de la tesis la experiencia del visitante público y del estudiante de primer nivel.

En este contexto, el presente trabajo tuvo como objetivo desarrollar un prototipo de chatbot académico personalizado basado en inteligencia artificial, orientado a atender consultas generales e información académica de estudiantes de primer nivel mediante procesamiento de lenguaje natural, recuperación contextual de información y reconocimiento de voz. Para ello, se establecieron los siguientes objetivos:

Objetivo general:

Desarrollar un prototipo de chatbot personalizado con reconocimiento de voz para la atención de consultas generales y académicas de los estudiantes de primer nivel en la carrera de Ingeniería de la PUCE-I.

Objetivos específicos:

- Diseñar una base de conocimiento institucional compuesta por documentos curados y fuentes relevantes para la atención de consultas generales mediante arquitectura RAG.
- Implementar una base de datos en PostgreSQL para gestionar información académica simulada de estudiantes de primer nivel.
- Integrar un modelo de lenguaje de gran escala que permita interpretar consultas y generar respuestas contextualizadas mediante recuperación de información.
- Construir una interfaz web en React integrada con un *backend* en Django REST Framework mediante servicios REST, diferenciando el acceso público al chatbot institucional y el acceso autenticado al *dashboard* estudiantil.
- Incorporar reconocimiento de voz como mecanismo complementario de interacción, permitiendo que el estudiante formule consultas habladas dentro del asistente conversacional.
- Evaluar el prototipo mediante pruebas funcionales, de usabilidad, seguridad e interacción por voz en un entorno controlado.

Finalmente, el documento se estructura en tres capítulos: el primero presenta el estado del arte y los fundamentos teóricos que sustentan la investigación; el segundo describe los materiales y métodos utilizados para el desarrollo del sistema; y el tercero expone los resultados obtenidos junto con su análisis correspondiente.

CAPÍTULO I

ESTADO DEL ARTE

El presente capítulo tiene como finalidad analizar los fundamentos teóricos y los antecedentes investigativos relacionados con el uso de chatbots basados en inteligencia artificial en el ámbito de la educación superior. Para ello, se examinan los principales enfoques tecnológicos que sustentan el desarrollo de asistentes conversacionales, así como su evolución, aplicaciones y limitaciones en contextos académicos. Dicho análisis permite establecer una base conceptual sólida que respalda el desarrollo del sistema propuesto, orientado a la atención de estudiantes de primer nivel mediante un chatbot personalizado.

1.1. Antecedentes

En los últimos años, la incorporación de tecnologías basadas en inteligencia artificial en la educación superior ha permitido automatizar procesos de atención y comunicación con los estudiantes. En este contexto, Labadze et al. (2023), mediante una revisión sistemática de la literatura, evidencian que los chatbots contribuyen a responder consultas frecuentes, mejorar la disponibilidad de información y reducir la carga operativa del personal administrativo. Sin embargo, aunque estos autores reconocen beneficios importantes en la atención estudiantil, también se identifica una limitación relacionada con la personalización de las respuestas, debido a que muchas soluciones analizadas no contemplan mecanismos de acceso a información académica individual. Esta limitación resulta relevante para la presente investigación, ya que el sistema propuesto integra una base de datos académica estructurada con el propósito de generar respuestas contextualizadas según el perfil del estudiante.

Por otra parte, Oliveira y Matos (2023) analizan los mecanismos tradicionales de atención estudiantil, tales como la atención presencial y el uso de correo electrónico, concluyendo que presentan limitaciones en disponibilidad, escalabilidad y tiempos de respuesta. No obstante, su análisis se enfoca principalmente en la necesidad de automatizar los servicios de atención, sin profundizar en mecanismos de personalización académica ni en el uso de modelos generativos vinculados a fuentes verificables. Frente a esta limitación, la presente investigación propone un chatbot académico inteligente capaz de procesar consultas en lenguaje natural y diferenciar entre información institucional general e información académica personalizada.

Asimismo, estudios recientes como los de McGrath et al. (2024) e Ilieva et al. (2023) analizan el impacto de los chatbots generativos en la educación superior, destacando su capacidad para producir respuestas más fluidas y naturales. Sin embargo, estos trabajos también advierten riesgos asociados con la generación de información incorrecta cuando los modelos no se encuentran conectados a fuentes de conocimiento verificables. Esta problemática resulta especialmente importante en contextos académicos, donde la precisión de la información entregada al estudiante constituye un requisito fundamental. Por esta razón, la presente investigación adopta una arquitectura híbrida basada en *Retrieval-Augmented Generation* (RAG), que permite recuperar información institucional previamente indexada antes de generar cada respuesta.

1.1.1. Chatbots en educación superior

Los chatbots han evolucionado desde sistemas basados en reglas hacia modelos impulsados por inteligencia artificial, capaces de interpretar consultas en lenguaje natural y ofrecer respuestas más flexibles. En educación superior, estos sistemas han sido utilizados para mejorar el acceso a información académica, apoyar servicios de orientación y reducir la carga operativa asociada a consultas frecuentes. En este sentido, Labadze et al. (2023) destacan su utilidad en procesos de atención estudiantil, mientras que Lambebo y Chen (2024) señalan que su adopción en educación superior se relaciona con factores como utilidad percibida, facilidad de uso, interactividad, privacidad y calidad del servicio.

No obstante, los sistemas tradicionales presentan limitaciones en la interpretación de consultas complejas y en la adaptación a diferentes contextos. En este sentido, McGrath et al. (2024) señalan que los modelos generativos ofrecen mayor flexibilidad en la interacción, aunque su desempeño depende en gran medida de la calidad de la información disponible.

Los estudios analizados evidencian la importancia de integrar mecanismos de comprensión del lenguaje natural con fuentes de información estructuradas y verificables. Esta combinación permite mejorar la precisión de las respuestas y ampliar las posibilidades de aplicación de los chatbots dentro de entornos educativos, donde la confiabilidad de la información constituye un requisito fundamental.

1.2. Marco Teórico

El desarrollo de sistemas conversacionales en entornos educativos requiere la integración de múltiples tecnologías, incluyendo inteligencia artificial, procesamiento de lenguaje natural y arquitecturas de software escalables. De acuerdo con McGrath et al. (2024) e Ilieva et al. (2023), esta combinación permite mejorar la interacción entre el usuario y el sistema, aunque su efectividad depende de la capacidad de integrar datos relevantes dentro del proceso de generación de respuestas.

La integración de estos componentes tecnológicos permite construir sistemas conversacionales capaces de ofrecer respuestas precisas, contextualizadas y adaptadas a las necesidades del estudiante dentro de un entorno académico específico.

1.2.1. Inteligencia Artificial

La inteligencia artificial engloba la capacidad de los sistemas computacionales para ejecutar tareas que tradicionalmente requerían intervención humana, tales como el aprendizaje a partir de datos, la toma de decisiones y la resolución de problemas. En el ámbito educativo, estas tecnologías han contribuido a la automatización de procesos de atención y al mejoramiento de los servicios dirigidos a los estudiantes. No obstante, la efectividad de estos sistemas depende de la calidad y disponibilidad de la información utilizada para generar respuestas, especialmente en contextos específicos donde se requiere conocimiento institucional especializado.

1.2.2. Chatbots o Asistentes Conversacionales

Los chatbots son sistemas que permiten a los usuarios comunicarse con una aplicación mediante lenguaje cotidiano, sin necesidad de aprender comandos específicos. En el ámbito estudiantil, Labadze et al. (2023) sostienen que estos sistemas contribuyen a reducir la carga del personal administrativo y a mejorar la disponibilidad del servicio. No obstante, los modelos basados únicamente en reglas presentan limitaciones cuando los usuarios formulan preguntas no previstas. Por su parte, McGrath et al. (2024) destacan que los chatbots impulsados por inteligencia artificial permiten superar varias de estas restricciones, especialmente en la comprensión de consultas expresadas de distintas maneras. Sin embargo, cuando estos modelos no disponen de suficiente contexto o información verificable, pueden generar respuestas imprecisas o alejadas del dominio institucional.

1.2.2.1. Chatbots Basados en Reglas

Los chatbots basados en reglas constituyen una de las primeras generaciones de sistemas conversacionales. Su funcionamiento se fundamenta en conjuntos predefinidos de reglas, patrones de texto y árboles de decisión que permiten asociar determinadas entradas del usuario con respuestas previamente configuradas.

En este tipo de sistemas, las respuestas se generan a partir de coincidencias exactas o parciales entre las palabras ingresadas por el usuario y las reglas almacenadas en el sistema. Por esta razón, su capacidad de comprensión es limitada y depende directamente de la cantidad y calidad de las reglas definidas durante el desarrollo.

Entre sus principales ventajas se encuentran la facilidad de implementación, el bajo consumo de recursos computacionales y el control total sobre las respuestas generadas. Sin embargo, presentan limitaciones importantes cuando los usuarios formulan preguntas utilizando expresiones no previstas, sinónimos o estructuras lingüísticas complejas.

Debido a estas restricciones, los chatbots basados exclusivamente en reglas suelen emplearse en escenarios donde las consultas son repetitivas y altamente estructuradas, como sistemas de preguntas frecuentes (FAQ), soporte técnico básico o atención automatizada de procesos específicos.

1.2.2.2. Chatbots Basados en IA

Los chatbots basados en inteligencia artificial representan una evolución significativa respecto a los sistemas tradicionales basados en reglas. Estos sistemas utilizan técnicas de procesamiento de lenguaje natural (*Natural Language Processing*, NLP), aprendizaje automático (*Machine Learning*) y modelos de lenguaje de gran escala (*Large Language Models*, LLMs) para comprender e interpretar las consultas realizadas por los usuarios.

A diferencia de los chatbots basados en reglas, estos sistemas no dependen exclusivamente de respuestas previamente definidas. En su lugar, analizan el contexto de la conversación, identifican la intención del usuario y generan respuestas dinámicamente utilizando modelos entrenados con grandes volúmenes de información.

El uso de inteligencia artificial permite mejorar la flexibilidad de la interacción, adaptarse a diferentes formas de expresión y ofrecer respuestas más naturales. Gracias a estas

capacidades, los chatbots inteligentes pueden aplicarse en ámbitos educativos, empresariales, financieros, sanitarios y de atención al cliente.

No obstante, estos sistemas también presentan desafíos relacionados con la precisión de la información generada, el consumo de recursos computacionales y la posibilidad de producir respuestas incorrectas cuando no disponen del contexto adecuado. Por esta razón, actualmente suelen complementarse con mecanismos de recuperación de información que permitan mejorar la confiabilidad de sus respuestas.

1.2.2.3. Chatbots Híbridos

Los chatbots híbridos combinan características de los sistemas basados en reglas y de los sistemas impulsados por inteligencia artificial, con el objetivo de aprovechar las ventajas de ambos enfoques. Esta integración permite mantener un mayor control sobre determinadas respuestas críticas mientras se incorporan capacidades avanzadas de comprensión y generación de lenguaje natural.

En una arquitectura híbrida, las consultas pueden procesarse mediante diferentes mecanismos dependiendo de su naturaleza. Las preguntas estructuradas y predecibles pueden resolverse mediante reglas específicas o consultas directas a bases de datos, mientras que las preguntas abiertas o contextuales pueden ser atendidas mediante modelos de inteligencia artificial.

La incorporación de arquitecturas *Retrieval-Augmented Generation* (RAG) ha fortalecido significativamente este enfoque, ya que permite complementar la generación de respuestas mediante la recuperación previa de información desde fuentes documentales o bases de conocimiento. De esta forma, el chatbot puede fundamentar sus respuestas en información verificable antes de generar una respuesta final.

Gracias a su capacidad para combinar múltiples fuentes de información y diferentes técnicas de procesamiento, los chatbots híbridos se han convertido en una de las alternativas más utilizadas en entornos educativos y empresariales, donde se requiere tanto precisión en los datos como flexibilidad conversacional.

1.2.2.4. Tipo de Chatbot Seleccionado Para la Investigación

Existen diferentes tipos de chatbots que pueden clasificarse según la forma en que procesan la información y generan respuestas. Entre los más comunes se encuentran los

chatbots basados en reglas, los chatbots impulsados por inteligencia artificial y los sistemas híbridos que combinan múltiples técnicas para mejorar su desempeño.

Para el desarrollo de la presente investigación se seleccionó un chatbot híbrido basado en inteligencia artificial generativa, debido a que esta arquitectura permite combinar la recuperación de información estructurada desde bases de datos relacionales con la generación de respuestas mediante modelos de lenguaje de gran escala (Large Language Models, LLMs).

La elección de este enfoque responde a las necesidades específicas del contexto académico, donde los estudiantes requieren tanto información personalizada, como horarios, asignaturas y aulas, así como información institucional de carácter general relacionada con procesos académicos, reglamentos y servicios universitarios.

A diferencia de los chatbots tradicionales basados únicamente en reglas predefinidas, el sistema propuesto incorpora capacidades de comprensión del lenguaje natural mediante el uso de modelos LLM, permitiendo interpretar consultas formuladas de diferentes maneras y generar respuestas más flexibles y contextualizadas. Asimismo, se implementa una arquitectura *Retrieval-Augmented Generation* (RAG), la cual complementa al modelo de lenguaje mediante la recuperación de información proveniente de una base de conocimiento actualizada antes de generar cada respuesta.

De esta manera, el chatbot desarrollado combina tres componentes principales: una base de datos académica relacional para la gestión de información personalizada del estudiante, una base de conocimiento documental para consultas institucionales y un modelo de lenguaje encargado de interpretar las preguntas y generar respuestas en lenguaje natural. Esta integración permite mejorar la precisión de las respuestas, reducir el riesgo de información incorrecta y ofrecer una experiencia conversacional más cercana a la interacción humana.

En consecuencia, el sistema desarrollado puede clasificarse como un chatbot académico inteligente basado en arquitectura híbrida RAG, capaz de integrar recuperación de información y generación de lenguaje natural para brindar asistencia académica personalizada a estudiantes de primer nivel de la Pontificia Universidad Católica del Ecuador Sede Ibarra.

1.2.3. Procesamiento de Lenguaje Natural (PLN)

El Procesamiento de Lenguaje Natural (PLN) constituye una rama de la inteligencia artificial orientada al análisis, comprensión y generación automática del lenguaje humano. Su finalidad es permitir que los sistemas computacionales interpreten consultas formuladas en lenguaje natural y produzcan respuestas comprensibles para los usuarios. En los últimos años, el desarrollo de modelos basados en arquitecturas Transformer ha transformado significativamente el campo del PLN, al mejorar la comprensión semántica, la generación de texto y la capacidad de mantener contexto dentro de una interacción conversacional (Zubiaga, 2024).

En los sistemas conversacionales modernos, el PLN ya no depende únicamente de reglas lingüísticas predefinidas o de modelos estadísticos tradicionales. Actualmente, los modelos de lenguaje de gran escala permiten interpretar consultas formuladas con diferentes estructuras, expresiones y niveles de complejidad. Esta evolución ha favorecido la implementación de asistentes conversacionales inteligentes en entornos educativos, donde los usuarios pueden formular preguntas de manera natural sin utilizar comandos específicos.

En la presente investigación, el PLN se considera la base conceptual que sustenta la interacción entre el estudiante y el chatbot académico. A partir de este enfoque, el sistema procesa consultas realizadas mediante texto o voz, interpreta su intención comunicativa y genera respuestas relacionadas con información académica personalizada o institucional.

1.2.4. Modelos de Lenguaje de Gran Escala (LLM)

Según Zhao et al. (2023), los Modelos de Lenguaje de Gran Escala (*Large Language Models*, LLM) representan una evolución significativa dentro de la inteligencia artificial, debido a que son capaces de comprender y generar lenguaje natural a partir del entrenamiento con grandes volúmenes de información textual. Estos modelos se fundamentan principalmente en arquitecturas Transformer, las cuales permiten analizar relaciones contextuales complejas entre palabras, frases y documentos.

De acuerdo con los autores, los LLM han demostrado un desempeño sobresaliente en tareas relacionadas con generación de texto, respuesta a preguntas, traducción automática, resumen documental y asistencia conversacional. Estas capacidades han favorecido su

adopción en diferentes áreas del conocimiento, incluyendo educación, salud, finanzas y atención al cliente.

No obstante, Zhao et al. (2023) también señalan que estos modelos pueden generar información incorrecta cuando no disponen de suficiente contexto o cuando las respuestas se construyen únicamente a partir del conocimiento aprendido durante su entrenamiento. Por esta razón, diversas investigaciones recientes han incorporado mecanismos de recuperación de información que complementan el proceso de generación de respuestas.

En la presente investigación, los LLM constituyen el componente encargado de interpretar las consultas realizadas por los usuarios y generar respuestas en lenguaje natural. Sin embargo, su funcionamiento se complementa mediante una arquitectura *Retrieval-Augmented Generation* (RAG), con el propósito de incorporar información institucional verificable antes de generar la respuesta final.

1.2.5. *Prompt Engineering*

Sahoo et al. (2024) definen el *Prompt Engineering* como el conjunto de técnicas orientadas al diseño y estructuración de instrucciones que permiten guiar el comportamiento de los modelos de lenguaje durante la generación de respuestas. Según estos autores, la calidad de los resultados producidos por un LLM depende en gran medida del contexto, las restricciones y las indicaciones proporcionadas al modelo antes del procesamiento de una consulta.

Los autores explican que un *prompt* puede incorporar información adicional relacionada con reglas de comportamiento, objetivos específicos, restricciones de contenido y contexto recuperado desde fuentes externas. Esta capacidad permite controlar parcialmente la generación de respuestas y mejorar su precisión dentro de dominios especializados.

En entornos académicos, el *Prompt Engineering* adquiere especial relevancia debido a que permite orientar al modelo hacia la utilización de información institucional válida, reduciendo la probabilidad de respuestas ambiguas o alejadas del contexto universitario.

En el sistema desarrollado durante esta investigación, el *Prompt Engineering* fue utilizado para establecer reglas de comportamiento del chatbot, diferenciar consultas académicas de consultas institucionales e incorporar contexto recuperado desde la base documental y la base de datos académica antes de solicitar la generación de una respuesta.

1.2.6. *Embeddings* y Representación Vectorial

Cao (2024) señala que los *embeddings* constituyen representaciones numéricas capaces de transformar palabras, frases o documentos completos en vectores matemáticos que preservan relaciones semánticas dentro de un espacio multidimensional. Gracias a este mecanismo, textos con significados similares pueden ubicarse próximos entre sí, incluso cuando utilizan vocabularios diferentes.

De acuerdo con el autor, los *embeddings* se han convertido en uno de los componentes fundamentales de los sistemas modernos de recuperación semántica de información, ya que permiten identificar relaciones conceptuales más allá de simples coincidencias léxicas. Esta característica resulta especialmente útil en motores de búsqueda inteligentes y sistemas conversacionales basados en inteligencia artificial.

Asimismo, Cao (2024) destaca que los *embeddings* permiten representar grandes volúmenes documentales en espacios vectoriales optimizados para realizar búsquedas por similitud. Como resultado, los sistemas pueden recuperar información relevante considerando el significado de una consulta y no únicamente las palabras exactas utilizadas por el usuario.

En arquitecturas *Retrieval-Augmented Generation*, los *embeddings* cumplen una función esencial al facilitar la indexación documental y la recuperación contextual de información. En la presente investigación, este mecanismo fue utilizado para transformar documentos institucionales en representaciones vectoriales que posteriormente fueron almacenadas e indexadas para realizar búsquedas semánticas eficientes.

1.2.7. Redes Neuronales y aprendizaje automático

Las redes neuronales y los modelos de aprendizaje automático forman parte de la base técnica de diversos sistemas conversacionales inteligentes, debido a su capacidad para reconocer patrones, apoyar la interpretación de entradas en lenguaje natural y mejorar la interacción entre usuario y sistema. En el ámbito educativo, Yildiz Durak y Onan (2024) analizan la aceptación de los sistemas de chatbot y evidencian que su uso depende de factores técnicos y perceptuales, como la utilidad percibida, la facilidad de uso, la confianza del usuario y la intención de adopción.

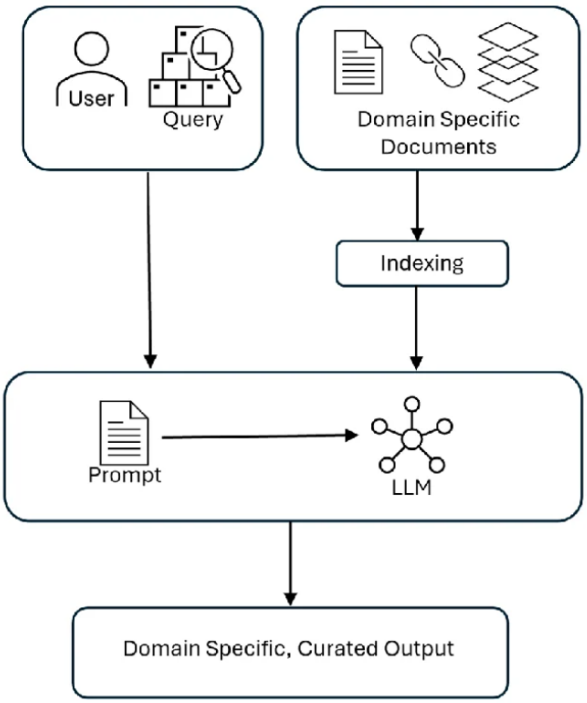
Sin embargo, estos modelos requieren datos adecuados para su entrenamiento y pueden presentar dificultades cuando se aplican en contextos institucionales específicos. Frente a esta limitación, los sistemas modernos basados en modelos de lenguaje incorporan

mecanismos de recuperación de información que permiten mejorar la calidad, pertinencia y confiabilidad de las respuestas generadas.

1.2.8. Arquitectura RAG

Tal como se observa en la Ilustración 1, la arquitectura Retrieval-Augmented Generation (RAG) integra dos procesos principales: la recuperación de información y la generación de respuestas. En primer lugar, el sistema identifica fragmentos relevantes dentro de una base documental previamente procesada e indexada. Posteriormente, dichos fragmentos se incorporan como contexto para que el modelo de lenguaje genere una respuesta más precisa y alineada con la información disponible. Esta combinación permite reducir la dependencia exclusiva del conocimiento interno del modelo y fundamentar las respuestas en fuentes documentales verificables.

Ilustración 1 Arquitectura basada en Retrieval-Augmented Generation (RAG).



Nota. Adaptado de DeBellis et al. (2025).

Investigaciones recientes han consolidado la arquitectura *Retrieval-Augmented Generation* (RAG) como un enfoque orientado a combinar modelos de lenguaje de gran escala con mecanismos de recuperación documental. Según Gao et al. (2023), RAG permite consultar fuentes de información externas antes de generar una respuesta, lo que contribuye a mejorar la precisión, reducir la generación de información incorrecta y proporcionar respuestas fundamentadas en conocimiento verificable.

De acuerdo con Gao et al. (2023), una arquitectura RAG se compone generalmente de tres etapas principales: indexación documental, recuperación de información relevante y generación de respuestas. Durante la etapa de indexación, los documentos son procesados y transformados en representaciones vectoriales mediante *embeddings*. Posteriormente, cuando el usuario realiza una consulta, el sistema recupera los fragmentos documentales con mayor similitud semántica. Finalmente, estos fragmentos son incorporados como contexto dentro del *prompt* enviado al modelo de lenguaje para generar una respuesta contextualizada.

Para implementar la recuperación semántica de información, las arquitecturas RAG suelen utilizar motores especializados de búsqueda vectorial. Entre las soluciones más utilizadas se encuentra FAISS (*Facebook AI Similarity Search*), una biblioteca orientada a la búsqueda eficiente por similitud y agrupamiento de vectores densos. De acuerdo con Douze et al. (2024), FAISS proporciona métodos de indexación, búsqueda, compresión, transformación y agrupamiento de vectores, lo que permite recuperar información relevante a partir de representaciones vectoriales. Gracias a estas características, FAISS resulta adecuado para sistemas modernos de recuperación documental, especialmente en arquitecturas *Retrieval-Augmented Generation* (RAG), donde se requiere localizar información contextual antes de generar una respuesta.

En una arquitectura RAG, los documentos previamente transformados en *embeddings* son almacenados dentro de índices vectoriales administrados por motores de búsqueda como FAISS. Posteriormente, cuando el usuario realiza una consulta, el sistema genera la representación vectorial correspondiente y la compara con los vectores almacenados, permitiendo recuperar los fragmentos documentales que presentan mayor similitud semántica. Este mecanismo facilita la recuperación contextual de información más allá de coincidencias exactas entre palabras.

Asimismo, Gao et al. (2023) destacan que RAG incorpora el concepto de grounding, mediante el cual las respuestas generadas por el modelo de lenguaje se fundamentan en información recuperada desde fuentes verificables. Esta característica resulta especialmente importante en entornos académicos, donde la precisión y confiabilidad de la información constituyen requisitos fundamentales para apoyar procesos de consulta y orientación estudiantil.

En la presente investigación, la arquitectura RAG fue utilizada para atender consultas institucionales relacionadas con reglamentos académicos, procesos universitarios, servicios institucionales y preguntas frecuentes de los estudiantes. Para ello, los documentos institucionales fueron segmentados, transformados en *embeddings* e indexados mediante mecanismos de búsqueda vectorial. Cuando un usuario realizaba una consulta, el sistema recuperaba los fragmentos documentales más relevantes y los incorporaba al contexto enviado al modelo de lenguaje, permitiendo generar respuestas fundamentadas en información verificable y alineada con el entorno universitario.

1.2.9. Reconocimiento por voz

El reconocimiento por voz permite convertir el lenguaje hablado en texto, facilitando la interacción con sistemas digitales mediante comandos orales. En sistemas conversacionales, esta funcionalidad amplía las modalidades de entrada disponibles para el usuario y puede mejorar la experiencia de interacción cuando se implementa como complemento de la escritura. Dentro del presente proyecto, el reconocimiento de voz fue utilizado para permitir que el estudiante formule consultas habladas, las cuales posteriormente fueron procesadas por el backend del chatbot.

No obstante, su precisión puede verse afectada por factores como el ruido o la variabilidad en la pronunciación. En este contexto, su implementación en el sistema propuesto se planteó como un complemento orientado a ampliar las formas de interacción, sin reemplazar los mecanismos tradicionales de entrada de texto.

Los sistemas modernos de reconocimiento de voz no se limitan únicamente a la transcripción del lenguaje hablado. Actualmente es posible integrar procesos de conversión de voz a texto (*Speech-to-Text*) y generación de voz sintética a partir de texto (*Text-to-Speech*), permitiendo desarrollar experiencias conversacionales multimodales. Esta integración resulta especialmente útil en entornos educativos, donde los estudiantes pueden interactuar con sistemas inteligentes mediante comandos de voz y recibir respuestas habladas de forma natural.

1.2.10. Personalización en Sistemas Educativos

La personalización constituye un elemento clave en los sistemas educativos inteligentes, debido a que permite adaptar la interacción según las características, necesidades y contexto del usuario. Yildiz Durak y Onan (2024) señalan que esta

adaptación mejora significativamente la experiencia de interacción, especialmente cuando el sistema puede responder de acuerdo con información específica del estudiante.

Sin embargo, muchos sistemas conversacionales carecen de acceso a datos individuales, lo que limita su capacidad para ofrecer respuestas realmente contextualizadas. Frente a esta problemática, la integración de bases de datos estructuradas permite consultar información académica asociada al estudiante, como horarios, asignaturas, calificaciones o actividades pendientes. En la presente investigación, este enfoque sustenta la personalización del chatbot, ya que las respuestas académicas se generan a partir del perfil del estudiante autenticado y no únicamente desde información general.

1.2.11. Arquitectura Cliente-Servidor y APIs REST

Las arquitecturas basadas en el modelo cliente-servidor permiten organizar una aplicación en componentes separados, donde el *frontend* gestiona la interacción con el usuario y el *backend* procesa la lógica de negocio, la autenticación y el acceso a datos. En este tipo de arquitectura, las APIs REST facilitan la comunicación entre ambas capas mediante servicios estructurados, lo que favorece la mantenibilidad, la integración de módulos y la evolución progresiva del sistema. En este sentido, Belov (2023) destaca la importancia de los modelos de interacción cliente-servidor en el desarrollo de aplicaciones web modernas.

En el presente proyecto, esta arquitectura se implementó mediante un *backend* desarrollado en Django y una interfaz de usuario en React, lo que permitió gestionar de manera eficiente la interacción del usuario con el sistema, así como la integración de los diferentes componentes tecnológicos.

1.2.12. Seguridad y Registro de eventos en Sistemas Inteligentes

La incorporación de mecanismos de seguridad constituye un aspecto fundamental en el desarrollo de aplicaciones basadas en inteligencia artificial que gestionan información sensible de los usuarios. En entornos académicos, la protección de datos personales, registros académicos y actividades realizadas dentro de la plataforma requiere la implementación de controles de acceso adecuados que garanticen la confidencialidad e integridad de la información.

Entre las estrategias más utilizadas se encuentran los sistemas de autenticación basados en *tokens*, la verificación de identidad mediante múltiples factores de autenticación y la

aplicación de políticas de autorización basadas en roles. Estas técnicas permiten restringir el acceso a recursos específicos según el perfil de cada usuario.

De forma complementaria, los mecanismos de registros de eventos permiten registrar eventos relevantes ocurridos dentro del sistema, facilitando la trazabilidad de acciones, el monitoreo de incidentes y el análisis posterior de actividades realizadas por los usuarios. La combinación de seguridad y registros de eventos contribuye a fortalecer la confianza, la transparencia y la gobernanza de los sistemas inteligentes modernos.

CAPÍTULO II

MATERIALES Y MÉTODOS

En el presente capítulo se describió el enfoque metodológico y los procedimientos utilizados para el desarrollo del chatbot académico inteligente, orientado a la atención de estudiantes de primer nivel en la carrera de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCE-I). Asimismo, se expusieron los fundamentos de la investigación, la metodología de desarrollo adoptada y la estructura técnica del sistema, con el fin de sustentar la coherencia, funcionalidad y pertinencia de la solución propuesta.

2.1. Aspectos generales de la investigación

En esta sección se presentaron los aspectos metodológicos que guiaron el desarrollo de la investigación, incluyendo el tipo de estudio, el enfoque aplicado, el alcance y las técnicas utilizadas para la recolección de información. Estos elementos permitieron comprender las necesidades de los estudiantes de primer nivel y definir criterios técnicos para el diseño, construcción y evaluación del prototipo.

2.1.1. Tipo de investigación

El desarrollo de este proyecto se enmarcó en una investigación de tipo aplicada, debido a que tuvo como finalidad desarrollar una solución tecnológica orientada a resolver una problemática específica en el contexto académico. En este caso, se abordó la necesidad de mejorar el acceso a la información por parte de los estudiantes de primer nivel de la carrera de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra, mediante la implementación de un chatbot académico personalizado.

2.1.2. Enfoque de la investigación

El presente estudio adoptó un enfoque mixto, debido a que combinó técnicas cualitativas y cuantitativas durante el desarrollo del chatbot académico inteligente.

El enfoque cualitativo se utilizó en la etapa de levantamiento de información, mediante observación directa y entrevistas no estructuradas aplicadas a estudiantes de primer nivel. Estas técnicas permitieron identificar necesidades académicas, dificultades de acceso a la información y requerimientos funcionales considerados durante el diseño del sistema.

Por otra parte, el enfoque cuantitativo se aplicó durante la fase de validación del prototipo, a través de métricas relacionadas con tiempos de respuesta, cumplimiento de criterios de

aceptación, comportamiento funcional del chatbot, interacción por voz, seguridad y registro de eventos. De esta manera, se pudo contrastar el funcionamiento del sistema con los resultados esperados definidos en el plan de pruebas.

La integración de ambos enfoques permitió comprender las necesidades de los usuarios y evaluar de manera objetiva el desempeño técnico de la solución desarrollada.

2.1.3. Alcance de la investigación

La investigación tuvo un alcance descriptivo y experimental. El componente descriptivo permitió caracterizar las necesidades de información de los estudiantes de primer nivel y analizar las tecnologías empleadas en el desarrollo del chatbot académico. El componente experimental se evidenció en la construcción y validación de un prototipo funcional, evaluado mediante pruebas controladas sobre consultas académicas, institucionales, interacción por voz, seguridad y registro de eventos.

El prototipo fue desarrollado en un entorno controlado utilizando datos académicos simulados que representaron información asociada a estudiantes de primer nivel, tales como horarios, asignaturas, aulas, calificaciones, actividades, entregas y notificaciones. Esta decisión permitió evaluar técnicamente el comportamiento del sistema sin utilizar información académica real.

El alcance incluyó el desarrollo de una plataforma académica inteligente compuesta por un chatbot institucional público, un *dashboard* académico para el estudiante autenticado, consultas personalizadas, recuperación contextual mediante arquitectura RAG, interacción por texto y voz, autenticación segura y registros de eventos. La validación se enfocó en verificar el funcionamiento de estos componentes desde la experiencia del visitante público y del estudiante de primer nivel.

Adicionalmente, el alcance del sistema se delimitó específicamente a estudiantes de primer nivel de la carrera de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra. Por esta razón, el prototipo no contempla funcionalidades dirigidas a otros niveles académicos ni a procesos administrativos complejos. Esta delimitación permitió mantener coherencia con los objetivos planteados y delimitar adecuadamente el alcance de la investigación.

Finalmente, se aclaró que el prototipo fue desarrollado únicamente con fines académicos y de investigación. Por esta razón, no se contempló su alojamiento, implementación productiva o despliegue en los servidores institucionales de la universidad.

2.1.4. Técnicas de recolección de información

Para el levantamiento de información se emplearon técnicas de carácter cualitativo orientadas a comprender las necesidades reales de los estudiantes y las limitaciones de los mecanismos tradicionales de atención académica. Entre estas técnicas se utilizaron la observación directa y la entrevista no estructurada, las cuales permitieron obtener información relevante para definir requerimientos funcionales, requerimientos no funcionales e historias de usuario del sistema.

La aplicación de estas técnicas facilitó la identificación de patrones de comportamiento, tipos de consultas frecuentes y dificultades en el acceso a la información académica. Los hallazgos obtenidos sirvieron como insumo para definir los requerimientos funcionales y no funcionales de la solución conversacional propuesta.

2.1.5. Observación directa

La observación directa se utilizó con el propósito de analizar el comportamiento de los estudiantes de primer nivel en su interacción con los canales de información institucional. A través de esta técnica se identificaron dificultades relacionadas con la localización de aulas, consulta de horarios y acceso a información académica.

Asimismo, se evidenciaron limitaciones relacionadas con la dependencia de terceros, la desorganización de la información y los tiempos prolongados de respuesta, lo cual justificó la necesidad de una solución automatizada.

Durante el desarrollo del proyecto, se consideraron las condiciones reales del entorno académico de la PUCE-I, lo que permitió ajustar la solución propuesta a necesidades concretas de los estudiantes, especialmente en relación con el acceso oportuno a la información académica.

2.1.6. Entrevistas no estructuradas

Las entrevistas realizadas fueron de tipo no estructurado y tuvieron como finalidad obtener información flexible y contextualizada sobre la experiencia de los estudiantes. Esta técnica permitió ajustar las preguntas durante la interacción, lo que facilitó la exploración de necesidades específicas y percepciones sobre los servicios institucionales.

A diferencia de los instrumentos estructurados, este tipo de entrevistas favoreció la obtención de información más detallada y espontánea, lo que resulta especialmente útil en la identificación de requerimientos para el desarrollo de sistemas interactivos como el chatbot propuesto.

Para la ejecución de las entrevistas se elaboró un guion base de preguntas, el cual sirvió como orientación durante el desarrollo de cada sesión. Este guion fue construido a partir de los objetivos de la investigación y del análisis previo del problema identificado, considerando las principales dificultades que enfrentan los estudiantes de primer nivel en el acceso a información académica.

Las preguntas se elaboraron tomando como referencia estudios relacionados con experiencia de usuario y sistemas conversacionales en entornos educativos, donde se prioriza la comprensión de necesidades reales del usuario. En este sentido, se incluyeron preguntas dirigidas a identificar aspectos como la frecuencia de consultas académicas, los canales utilizados para obtener información, las dificultades encontradas y las expectativas frente a una herramienta automatizada de apoyo.

Las entrevistas fueron aplicadas directamente a estudiantes de primer nivel de la carrera de Ingeniería de la PUCE-I, lo que permitió obtener información contextualizada y alineada con el alcance del proyecto. Esta información sirvió como base para la definición de requerimientos y el diseño funcional del sistema propuesto.

2.1.7. Resultados de las entrevistas

A partir de las entrevistas no estructuradas realizadas a 55 estudiantes de primer nivel de la carrera de Ingeniería de la PUCE-I, seleccionados mediante muestreo por conveniencia, se identificaron las principales necesidades informativas y limitaciones en los canales de atención institucional. Para organizar los hallazgos, las respuestas fueron agrupadas en categorías recurrentes, lo que permitió presentar una síntesis cualitativa en la Tabla 1 y una distribución porcentual de las categorías más frecuentes en la Tabla 2.

Tabla 1 Síntesis de resultados obtenidos en entrevistas a estudiantes de primer nivel

Pregunta Realizada	Resultado Obtenido
¿Qué tipo de información consultas con mayor frecuencia en la universidad?	Predomina la consulta de horarios, asignaturas del día y ubicación de aulas.

¿Cómo obtienes actualmente esa información?	Principalmente mediante compañeros, redes sociales o consulta presencial.
¿Has tenido dificultades para acceder a información académica?	Se identificaron problemas de acceso, desorganización y falta de claridad.
¿Qué tan rápido recibes respuesta a tus consultas?	Los tiempos de respuesta suelen ser variables y, en muchos casos, tardíos.
¿Te gustaría contar con un sistema automatizado?	Existe una alta aceptación hacia el uso de un chatbot académico.
¿Qué funcionalidades consideras necesarias?	Acceso a horarios, materias, ubicación de aulas y respuestas personalizadas.

Tabla 2 Distribución de respuestas obtenidas en las entrevistas realizadas a estudiantes de Ingeniería (n = 55)

Pregunta	Respuesta	Cantidad	Porcentaje
Información más consultada	Horarios	35	63.64%
Información más consultada	Aulas	12	21.82%
Información más consultada	Asignaturas	8	14.54%
Desea utilizar un chatbot académico	Sí	52	94.55%
Desea utilizar un chatbot académico	No	3	5.45%

Como se observa en la Tabla 2, el 63,64 % de los estudiantes consultó con mayor frecuencia información relacionada con horarios académicos, seguido por la ubicación de aulas con un 21,82 % y las asignaturas programadas con un 14,54 %. Estos resultados evidencian que las principales necesidades de información se concentran en elementos básicos de organización académica, lo que confirma la pertinencia de implementar un mecanismo de acceso rápido y centralizado para los estudiantes de primer nivel.

Asimismo, se identificó que una proporción significativa de estudiantes utiliza canales informales para resolver sus consultas, tales como compañeros, grupos de mensajería

instantánea o redes sociales, situación que puede generar retrasos y diferencias en la calidad de la información recibida. De igual manera, el 94,55 % de los participantes manifestó interés en disponer de una herramienta automatizada capaz de proporcionar respuestas inmediatas y personalizadas, mientras que únicamente el 5,45 % indicó no considerar necesaria esta funcionalidad.

Los resultados obtenidos permitieron sustentar la definición de los requerimientos funcionales del sistema y orientar el desarrollo del chatbot hacia la resolución de las necesidades informativas identificadas durante el proceso de levantamiento de información, especialmente aquellas relacionadas con el acceso a horarios, asignaturas y ubicación de aulas.

2.2. Metodología de Desarrollo del Sistema

El desarrollo del sistema se llevó a cabo mediante una adaptación de la **metodología ágil *Scrum***, debido a su enfoque iterativo e incremental. Esta metodología permitió organizar el trabajo en *Sprints*, gestionar los requerimientos mediante un *Product Backlog* priorizado y validar progresivamente cada incremento funcional desarrollado durante la construcción del chatbot académico inteligente.

En este proyecto, *Scrum* fue adaptado a un contexto académico con un equipo reducido. La tutora cumplió el rol de *Product Owner*, orientando la priorización y revisión de los requerimientos, mientras que el autor asumió las responsabilidades de *Scrum Master* y desarrollador principal del sistema. Esta adaptación permitió mantener una planificación ordenada, sin perder la naturaleza iterativa del proceso de desarrollo.

2.2.1. Casos de Uso y Alcance Funcional

Para definir el comportamiento funcional del chatbot académico, se identificaron los principales escenarios de interacción entre el visitante público, el estudiante de primer nivel autenticado y los controles técnicos de soporte del sistema. Esta delimitación permitió separar las consultas institucionales generales de las consultas académicas personalizadas, evitando que la información privada del estudiante fuera accesible desde el modo público.

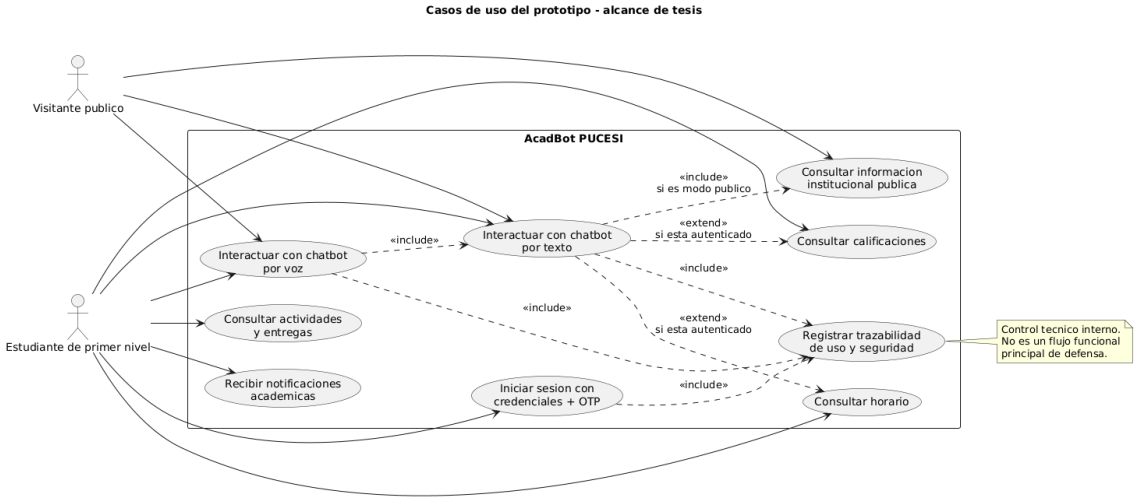
Entre los casos de uso principales se identificaron la consulta de información institucional mediante el chatbot público, el acceso autenticado al *dashboard* estudiantil, la consulta de horarios, asignaturas, calificaciones, actividades pendientes, notificaciones y ubicación de aulas, así como la interacción con el asistente mediante texto o voz. Estos

casos de uso permitieron establecer las funcionalidades esenciales del prototipo y organizar el alcance de las consultas que el sistema fue capaz de procesar.

El chatbot fue orientado específicamente a estudiantes de primer nivel de la carrera de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra. Por esta razón, la validación funcional se centró en dos escenarios principales: el visitante público que realiza consultas institucionales generales y el estudiante autenticado que accede a información académica personalizada. Los módulos relacionados con docentes, administración técnica y registros de eventos fueron considerados componentes de soporte para alimentar actividades, administrar datos, mantener trazabilidad técnica y observar el comportamiento del sistema, sin constituir el flujo funcional principal evaluado en la tesis.

Los casos de uso fueron definidos tomando como referencia los incrementos funcionales desarrollados durante los *Sprints*: consultas académicas personalizadas, consultas institucionales mediante recuperación documental, interacción por voz, experiencia conversacional web, autenticación segura y registros de eventos. Esta organización permitió mantener coherencia entre los objetivos de la investigación, el desarrollo del prototipo y la validación aplicada.

Ilustración 2 Casos de Uso.



2.2.2. *Product Backlog*

A partir del análisis de los requerimientos del sistema, se definió el Product Backlog, el cual agrupó las funcionalidades necesarias para el desarrollo del chatbot académico. Este artefacto permitió organizar el trabajo de manera priorizada, tomando

como referencia las necesidades identificadas en los estudiantes de primer nivel y el alcance funcional definido para el prototipo.

Los requerimientos se clasificaron en funcionales y no funcionales, con el propósito de establecer una visión clara del comportamiento esperado del sistema y de las características de calidad que debía cumplir. Los requerimientos funcionales se relacionaron con la atención de consultas académicas personalizadas, consultas institucionales mediante arquitectura RAG, interacción por voz, interfaz conversacional, autenticación segura y registros de eventos. Por su parte, los requerimientos no funcionales establecieron criterios de usabilidad, eficiencia, seguridad, disponibilidad y escalabilidad.

A partir de estos requerimientos, se definieron historias de usuario orientadas principalmente al estudiante de primer nivel y al visitante público. Cada historia de usuario permitió representar una funcionalidad desde la perspectiva del usuario final, manteniendo la separación entre información institucional pública e información académica personalizada.

Cada historia de usuario incluyó los siguientes elementos: identificador, nombre, estimación de tiempo, importancia, descripción, criterios de aceptación y dependencias. Estos elementos facilitaron la trazabilidad de los requerimientos, la priorización de funcionalidades y la planificación de los *Sprints* de desarrollo. En la Tabla 3 se presenta el *Product Backlog* definido para el sistema.

Tabla 3 Product Backlog del chatbot académico personalizado para estudiantes de primer nivel.

HISTORIAS DE USUARIO						
ID	**NOMBRE**	**ESTIMACIÓN (DÍAS)**	**IMPORTANCIA**	**DESCRIPCIÓN DE HISTORIA DE USUARIO**	**CRITERIOS DE ACEPTACIÓN**	**DEPENDENCIAS**
S-1	Consulta académica personalizada	10	Alta	Como estudiante de primer nivel, deseo consultar información académica personalizada relacionada con horarios, asignaturas y aulas para organizar mis actividades universitarias.	• El sistema debe recuperar horarios desde PostgreSQL. • Debe mostrar asignaturas asociadas. • Debe indicar el aula correspondiente. • La respuesta debe mostrarse en menos de 3 segundos.	Ninguna

S-2	Consultas institucionales mediante RAG	10	Alta	Como estudiante, deseo realizar consultas institucionales para resolver dudas académicas y administrativas mediante lenguaje natural.	• El sistema debe consultar documentos indexados. • Debe generar respuestas verificadas. • Debe utilizar arquitectura RAG. • La respuesta debe mantener coherencia contextual. • La respuesta debe mostrarse en menos de 5 segundos.	S-1
S-3	Interacción mediante reconocimiento de voz	10	Media	Como estudiante, deseo interactuar mediante comandos de voz para acceder al sistema de forma más rápida y accesible.	• El sistema debe convertir voz a texto. • Debe interpretar correctamente la consulta. • Debe integrarse con la interfaz web.	S-1, S-2
S-4	Interfaz conversacional inteligente	10	Alta	Como estudiante, deseo interactuar con una interfaz conversacional intuitiva para obtener respuestas académicas de manera eficiente.	• El *frontend* debe mostrar respuestas conversacionales. • Debe mantener flujo de mensajes. • Debe gestionar estados de carga y respuesta. • Debe reiniciarse cada 30 segundos para no tener una sesión de chatbot abierta de forma continua.	S-1, S-2, S-3
S-5	Autenticación segura, control de acceso y registros de eventos	10	Alta	Como estudiante de primer nivel, deseo acceder al sistema mediante autenticación segura para proteger mi información académica personalizada y garantizar que mis acciones principales queden	• El estudiante debe autenticarse mediante credenciales válidas. • El sistema debe solicitar verificación OTP antes de habilitar el acceso al *dashboard* académico. • El sistema debe emitir *tokens*	S-4

				registradas para fines de trazabilidad.	JWT para gestionar la sesión autenticada. • El sistema debe permitir el acceso únicamente a usuarios autorizados. • El sistema debe impedir el acceso a información académica personalizada sin una sesión válida. • Las acciones relevantes realizadas dentro de la plataforma deben quedar registradas en los registros de eventos. • El sistema debe proteger la información académica personalizada del estudiante.	

Planificación: *Sprint* 0

Previo al inicio de los *Sprints* de desarrollo, se ejecutó un *Sprint* 0 orientado a la planificación general del proyecto. Durante esta fase se definieron los objetivos del sistema, el alcance funcional, las historias de usuario, los requerimientos funcionales y no funcionales, así como la estimación inicial del esfuerzo requerido para cada funcionalidad.

Como principal artefacto de *Scrum* se elaboró el *Product Backlog*, en el cual se registraron las historias de usuario priorizadas de acuerdo con su valor para el estudiante y su complejidad técnica. Este backlog constituyó la base para organizar el trabajo de manera incremental y planificar los *Sprints* posteriores.

Posteriormente, se realizó el análisis y refinamiento de las historias de usuario, con el propósito de descomponer cada requerimiento en tareas específicas que facilitaran su implementación y seguimiento durante el desarrollo del sistema. Este proceso permitió

identificar dependencias entre funcionalidades, estimar con mayor precisión el esfuerzo requerido y establecer una secuencia lógica de construcción del chatbot académico.

La planificación del desarrollo se estructuró de manera progresiva. En primer lugar, se consideraron las actividades relacionadas con la base de datos académica y la gestión de información personalizada del estudiante. Luego, se planificaron las funcionalidades asociadas a la recuperación de información institucional mediante arquitectura *Retrieval-Augmented Generation* (RAG), la interacción por voz y la interfaz conversacional. Finalmente, se integraron las actividades relacionadas con autenticación segura, control de acceso, registros de eventos y validación del sistema.

La planificación realizada durante el *Sprint* 0 permitió mantener la trazabilidad entre los requerimientos identificados, las historias de usuario definidas y las funcionalidades implementadas en cada *Sprint*. Como resultado de este proceso, se obtuvo un conjunto de tareas organizadas según su prioridad, dependencia y esfuerzo estimado, cuyo análisis y refinamiento se presenta en la Tabla 4.

Tabla 4 Análisis y refinamiento de las historias de usuario.

Prioridad	**Número de historias de Usuario**	**ID de Tarea**	**Tareas**	**Días – (lunes - viernes) 4horas**
Alta	1	S-1-1	Diseño de base de datos académica	2
		S-1-2	Configuración de PostgreSQL	2
		S-1-3	Desarrollo API REST académica	2
		S-1-4	Integración de horarios y asignaturas	1
		S-1-5	Implementación consulta de aulas	1
		S-1-6	Pruebas funcionales académicas	2
Alta	2	S-2-1	Organización de documentos institucionales	2
		S-2-2	Configuración arquitectura RAG	2
		S-2-3	Implementación recuperación documental	2
		S-2-4	Integración con LLM	2
		S-2-5	Validación de respuestas institucionales	2
Media	3	S-3-1	Implementación reconocimiento de voz	2
		S-3-2	Conversión Speech-to-Text	2
		S-3-3	Integración *frontend* voz	2
		S-3-4	Interpretación NLP de voz	2
		S-3-5	Pruebas multimodales	2
Alta	4	S-4-1	Diseño interfaz conversacional React	2
		S-4-2	Gestión flujo de mensajes	2
		S-4-3	Manejo de estados y respuestas	2
		S-4-4	Integración chatbot *frontend*	2
		S-4-5	Pruebas integrales UI/UX	2
Alta	5	S-5-1	Implementación de registros de eventos	2
		S-5-2	Implementación de autenticación JWT	2
		S-5-3	Gestión de sesiones seguras	2
		S-5-4	Control de acceso basado en roles	2

		S-5-5	Pruebas de seguridad y autenticación	2

Una vez definido el conjunto de tareas, se procedió a su organización en *Sprints*, considerando la capacidad de desarrollo, las dependencias entre actividades y la necesidad de construir el sistema de manera progresiva. Esta planificación permitió distribuir el trabajo en iteraciones, asegurando que cada *Sprint* aportara un incremento funcional al prototipo.

La organización de los *Sprints* respondió a una secuencia lógica de construcción del sistema. Inicialmente se abordaron las tareas relacionadas con la configuración base y la estructura de datos académicos. Posteriormente, se desarrollaron las funcionalidades vinculadas a la gestión de información personalizada, la recuperación institucional mediante RAG, la interacción por voz y la interfaz conversacional. Finalmente, se integraron los mecanismos de seguridad, autenticación, registros de eventos y validación general del sistema.

De manera general, se estableció una duración de dos semanas por *Sprint*, considerando jornadas de trabajo de cuatro horas diarias de lunes a viernes. Esta planificación permitió mantener un desarrollo iterativo y progresivo acorde con el contexto académico del proyecto.

Duración del *Sprint* = 2 semanas = 10 días hábiles

Velocidad de desarrollo por *Sprint* = 40 horas

La distribución de las tareas en cada *Sprint* se presenta en la Tabla 5, donde se detalla la planificación del desarrollo en función de la prioridad, las dependencias y el incremento funcional correspondiente.

Tabla 5 Distribución de Sprints.

SPRINT	**TAREA**
1	S-1
2	S-2
3	S-3
4	S-4
5	S-5

Requerimientos Funcionales

A partir del análisis del *Product Backlog* y de las necesidades identificadas mediante observación directa y entrevistas no estructuradas, se definieron los

requerimientos funcionales del sistema. Estos describen las funcionalidades que el chatbot académico debía cumplir para satisfacer los objetivos planteados en el proyecto y responder a las necesidades de los estudiantes de primer nivel.

Los requerimientos funcionales permitieron establecer el comportamiento esperado del sistema, su interacción con el usuario y la comunicación entre los componentes tecnológicos que lo conforman, como la interfaz web, el *backend*, la base de datos académica, la arquitectura RAG, los servicios de voz y los mecanismos de seguridad.

A continuación, se detallan los principales requerimientos funcionales:

- El sistema debe consultar información académica desde la base de datos y presentar respuestas asociadas únicamente al estudiante autenticado.
- El chatbot debe responder consultas institucionales generales mediante información recuperada desde la base de conocimiento implementada con arquitectura RAG.
- El sistema debe diferenciar entre consultas institucionales públicas y consultas académicas personalizadas, aplicando autenticación únicamente cuando se solicite información protegida.
- El sistema debe impedir el acceso a información académica personalizada cuando el usuario no cuente con una sesión válida.
- El sistema debe permitir el inicio de sesión del estudiante mediante credenciales válidas y verificación OTP como mecanismo de autenticación de dos factores.
- El chatbot debe permitir la interacción mediante texto y voz desde la interfaz conversacional.
- El sistema debe convertir las consultas habladas en texto procesable antes de enviarlas al *backend* para su análisis.
- El sistema debe generar respuestas en lenguaje natural, claras, coherentes y relacionadas con la intención de la consulta realizada.
- El sistema debe registrar eventos relevantes asociados con autenticación, consultas al chatbot, acceso a información académica, interacción por voz, errores y cierre de sesión.

Requerimientos No Funcionales

Los requerimientos no funcionales establecen las características de calidad que debía cumplir el sistema para garantizar su correcto desempeño, seguridad, usabilidad, disponibilidad y capacidad de crecimiento durante la validación del prototipo.

Usabilidad

- La interfaz conversacional debe ser clara e intuitiva para visitantes públicos y estudiantes autenticados, permitiendo que el usuario realice una consulta sin requerir asistencia externa.
- El widget conversacional debe permitir la apertura, cierre, envío de consultas, visualización del historial reciente y reinicio por inactividad de forma sencilla.
- Las respuestas generadas por el chatbot deben ser comprensibles, coherentes y fáciles de interpretar durante las pruebas de validación del prototipo.

Eficiencia

- El sistema debe responder consultas académicas personalizadas directas en un tiempo máximo de 3 segundos bajo condiciones normales de prueba.
- Las consultas institucionales procesadas mediante arquitectura RAG deben mantener un tiempo de respuesta máximo recomendado de 5 segundos, considerando la recuperación documental y la generación de la respuesta.
- La interfaz debe mantener una interacción fluida durante el envío de consultas, la carga de respuestas y la reproducción de audio, sin bloqueos visibles durante la ejecución de las pruebas funcionales.

Seguridad

- El sistema debe proteger la información académica personalizada mediante autenticación, control de acceso y validación de sesión activa.
- El acceso al *dashboard* académico debe requerir credenciales válidas y verificación OTP antes de permitir la consulta de información personal del estudiante.
- El sistema debe utilizar *tokens* JWT para gestionar sesiones autenticadas y rechazar solicitudes cuando el *token* esté ausente, sea inválido o haya expirado.
- El sistema debe impedir el acceso a recursos protegidos en el 100 % de los casos en los que no exista una sesión válida.

Registros de eventos

- El sistema debe registrar eventos relevantes para fines de monitoreo, trazabilidad y revisión del comportamiento del prototipo.
- Los registros de eventos deben incluir información mínima como usuario o identificador de sesión, fecha, hora, tipo de evento, resultado de la acción y dirección IP cuando esté disponible.
- El sistema debe almacenar el 100 % de los eventos críticos definidos durante la validación, incluyendo inicio de sesión, cierre de sesión, consultas académicas, consultas institucionales, errores e interacción por voz.

Disponibilidad

- La interfaz web y el chatbot deben mantenerse disponibles durante la ejecución de los casos de prueba aplicados en el entorno controlado.
- El chatbot público debe permanecer disponible para la atención de consultas institucionales generales durante la validación del prototipo.
- Los servicios de autenticación, consulta académica, consulta institucional y voz deben responder correctamente durante la ejecución de los casos de prueba aplicados en el entorno controlado.

Escalabilidad

- La arquitectura debe permitir la incorporación de nuevas fuentes documentales a la base de conocimiento sin modificar la lógica conversacional principal.
- La base de conocimiento debe permitir la actualización o reconstrucción del índice documental cuando se agreguen nuevos reglamentos, procedimientos o documentos institucionales.
- La estructura modular del *backend* debe facilitar la integración futura de nuevos servicios académicos sin afectar los módulos existentes.
- La interfaz *frontend* debe permitir la ampliación de funcionalidades del chatbot sin alterar el funcionamiento principal del widget conversacional.

Roles del Equipo *Scrum*

El desarrollo del sistema se ejecutó en un entorno académico con un único desarrollador, por lo que los roles de *Scrum* fueron adaptados al contexto del proyecto. Esta adaptación permitió organizar el trabajo de manera iterativa, mantener una

planificación progresiva y validar cada incremento funcional sin requerir un equipo de desarrollo amplio.

La aplicación fue desarrollada para responder a las necesidades identificadas en estudiantes de primer nivel de la carrera de Ingeniería de la Pontificia Universidad Católica del Ecuador Sede Ibarra, quienes requerían acceso rápido y centralizado a información académica e institucional. En este sentido, los roles fueron asumidos de la siguiente manera.

***Product Owner* (representado por la tutora de tesis):**

La tutora de tesis asumió el rol de *Product Owner*, participando en la definición, validación y priorización del *Product Backlog* conforme a las necesidades detectadas durante el levantamiento de información. Asimismo, supervisó el refinamiento de las historias de usuario, la coherencia funcional de los incrementos desarrollados y la alineación del sistema con los objetivos de la investigación.

***Scrum* Master (asumido por el desarrollador):**

El autor asumió el rol de *Scrum Master*, encargado de organizar el proceso de desarrollo mediante *Sprints*, planificar las actividades de cada iteración, controlar el avance del proyecto, resolver impedimentos técnicos y mantener la mejora continua durante la construcción del prototipo.

Equipo de desarrollo (asumido por el desarrollador):

El autor también asumió el rol de equipo de desarrollo, ejecutando las tareas técnicas relacionadas con el diseño de la base de datos, desarrollo del *backend* en Django REST *Framework*, implementación del *frontend* en React, integración de la arquitectura RAG, configuración del modelo de lenguaje, incorporación de interacción por voz, autenticación segura, registros de eventos y ejecución del plan de pruebas.

Esta adaptación de *Scrum* resultó adecuada para un proyecto académico de investigación aplicada, debido a que permitió mantener una organización incremental del trabajo sin perder la trazabilidad entre requerimientos, desarrollo y validación del sistema. De esta manera, el proceso metodológico se mantuvo orientado a satisfacer las necesidades del usuario principal definido para la tesis: el estudiante de primer nivel.

2.3. Diseño de la arquitectura de software del Sistema

El diseño del chatbot académico inteligente se estructuró, con base en la metodología, de manera incremental mediante los *Sprints* establecidos en la planificación, alineados con las historias de usuario definidas en el *Product Backlog*. Cada *Sprint* permitió incorporar nuevas funcionalidades y componentes arquitectónicos, garantizando una evolución progresiva del sistema conversacional. A continuación, se describe cada uno de los productos obtenidos durante el proceso de desarrollo.

2.3.1. Diagramas de secuencias desarrollados en cada *Sprint*

Se presentan los diagramas de secuencia elaborados en cada *Sprint* como parte del desarrollo de funcionalidades.

2.3.1.1. *Sprint* 1: Consultas Académicas Personalizadas

Durante el *Sprint* 1 se implementaron las funcionalidades relacionadas con la consulta académica personalizada para estudiantes de primer nivel. El objetivo principal de esta iteración fue desarrollar los mecanismos necesarios para recuperar y presentar información académica específica del estudiante, incluyendo horarios, asignaturas y ubicación de aulas.

Para ello, inicialmente se realizó la construcción de la estructura de almacenamiento de datos académicos, definiendo las entidades necesarias para representar estudiantes, asignaturas, docentes, horarios y aulas. Posteriormente se configuró el sistema gestor de base de datos PostgreSQL y se desarrollaron los servicios encargados de acceder a la información almacenada.

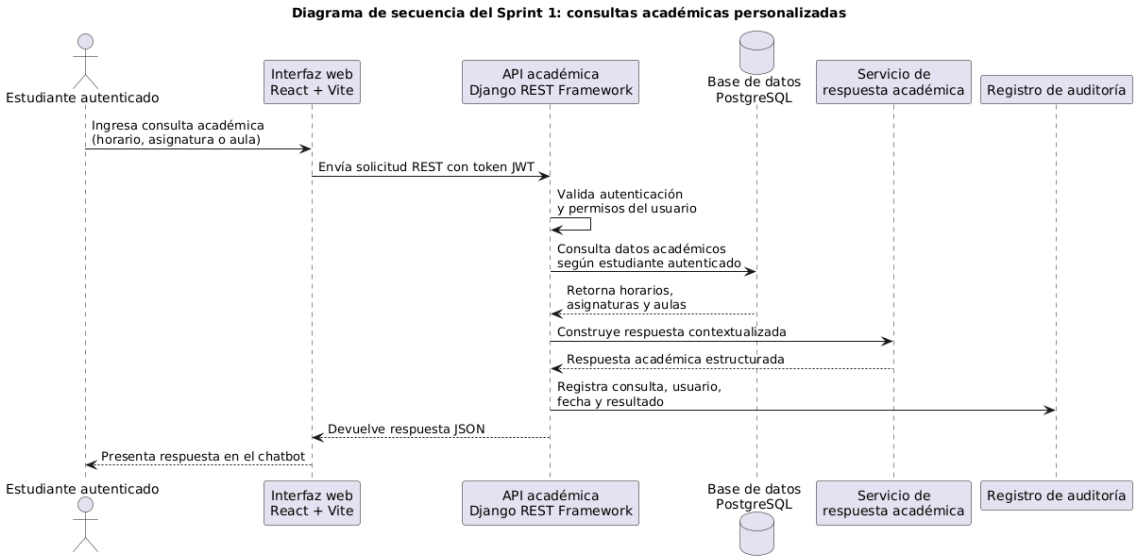
Una vez disponible la información académica, se implementaron los servicios REST necesarios para procesar las solicitudes realizadas por los usuarios. Estos servicios permitieron recuperar los horarios asignados al estudiante, las materias registradas en su carga académica y la ubicación de las aulas correspondientes.

La personalización implementada en la investigación se fundamentó en el acceso controlado a información académica asociada al estudiante autenticado. Mediante la integración de la base de datos PostgreSQL y los servicios desarrollados en Django REST Framework, el sistema permitió proporcionar respuestas relacionadas con horarios, calificaciones, asignaturas, actividades pendientes y notificaciones académicas. Esta funcionalidad permitió adaptar las respuestas al contexto académico de cada usuario y mejorar la utilidad del asistente conversacional.

Finalmente, se ejecutaron pruebas funcionales orientadas a verificar la correcta recuperación de los datos académicos y la generación de respuestas coherentes para las consultas realizadas por los estudiantes.

Como resultado de este *Sprint* se obtuvo un módulo funcional capaz de proporcionar información académica personalizada mediante consultas realizadas a la base de datos académica implementada para el prototipo.

Ilustración 3 Diagrama de Secuencia Sprint 1.



2.3.1.1.1. Diseño de la base de datos académica

La base de datos académica fue incorporada con el propósito de proporcionar al sistema una fuente estructurada de información que permitiera responder consultas académicas personalizadas. Su diseño contempló las entidades necesarias para representar estudiantes, asignaturas, docentes, horarios, aulas, actividades académicas y notificaciones, estableciendo las relaciones requeridas para garantizar la integridad de los datos.

Esta estructura permitió asociar la información académica a cada estudiante autenticado, facilitando la recuperación de datos específicos durante el procesamiento de consultas. De esta manera, las respuestas generadas por el chatbot pudieron fundamentarse en información verificable almacenada dentro del sistema.

Asimismo, la organización de la base académica permitió diferenciar claramente la información institucional de la información académica personalizada. Mientras las consultas generales fueron gestionadas mediante mecanismos independientes de

recuperación documental, las consultas personalizadas requirieron acceso controlado a los datos asociados al estudiante.

Dentro de la arquitectura implementada, esta funcionalidad fue gestionada principalmente mediante la aplicación *students*, responsable de administrar perfiles estudiantiles, matrículas, asignaturas, horarios, calificaciones, actividades y notificaciones académicas.

2.3.1.1.2. Implementación de PostgreSQL

Una vez definido el modelo de datos, se procedió a la implementación de la base de datos utilizando PostgreSQL como sistema gestor relacional. Esta tecnología fue seleccionada debido a su capacidad para gestionar relaciones complejas entre entidades académicas y garantizar la consistencia de la información almacenada.

La configuración de PostgreSQL se realizó desde el *backend* desarrollado en Django mediante variables de entorno, permitiendo centralizar la administración de la conexión y evitar la exposición de credenciales sensibles dentro del código fuente.

El uso de PostgreSQL proporcionó mecanismos de integridad referencial, restricciones de unicidad e índices orientados a optimizar el rendimiento de consultas frecuentes. Estas características facilitaron la recuperación eficiente de información académica relacionada con asignaturas, horarios, calificaciones y actividades registradas en el sistema.

Adicionalmente, PostgreSQL permitió almacenar información asociada a la interacción de usuarios autenticados, contribuyendo a la trazabilidad de las consultas realizadas y al soporte de funcionalidades posteriores relacionadas con registro de eventos y monitoreo.

2.3.1.1.3. Desarrollo de Servicios REST

Posteriormente se desarrollaron los servicios REST encargados de exponer la información académica almacenada en PostgreSQL hacia la interfaz *frontend* y los componentes conversacionales del sistema. Esta capa fue implementada mediante Django REST Framework, manteniendo una separación entre modelos, serializadores, vistas y lógica de negocio.

Los servicios desarrollados permitieron acceder a información relacionada con perfiles estudiantiles, asignaturas matriculadas, horarios, calificaciones, actividades académicas y notificaciones. Asimismo, constituyeron el mecanismo de comunicación entre la base de datos académica y los diferentes módulos de la plataforma.

Con el fin de proteger la información académica de los usuarios, los *endpoints* incorporaron mecanismos de autenticación basados en JSON *Web Tokens* (JWT), restringiendo el acceso únicamente a estudiantes previamente autenticados.

La implementación de esta capa permitió integrar de manera eficiente la base académica con la interfaz web y el servicio conversacional, proporcionando la información necesaria para la generación de respuestas personalizadas.

2.3.1.1.4. Personalización de respuestas académicas

La personalización de respuestas académicas constituyó el principal resultado funcional obtenido durante el *Sprint* 1. Esta funcionalidad permitió que el chatbot utilizara información académica asociada al estudiante autenticado para generar respuestas contextualizadas según su situación académica particular.

Para ello, el sistema recuperó información proveniente de la base de datos académica y construyó dinámicamente el contexto utilizado durante la generación de respuestas. Entre los datos considerados se incluyeron asignaturas matriculadas, horarios, calificaciones, actividades pendientes y notificaciones académicas.

La integración de esta información dentro del flujo conversacional permitió responder consultas específicas relacionadas con el desempeño académico del estudiante, evitando respuestas genéricas cuando la consulta requería información personalizada.

Asimismo, se estableció una separación entre las consultas institucionales generales y las consultas académicas personalizadas. Mientras las primeras utilizaron mecanismos de recuperación documental, las segundas dependieron exclusivamente de información académica asociada al usuario autenticado.

Como resultado, el chatbot evolucionó desde un asistente informativo general hacia una herramienta capaz de proporcionar orientación académica personalizada, mejorando la utilidad práctica del sistema dentro del entorno universitario.

2.3.1.2. *Sprint* 2: Consultas institucionales

Durante el *Sprint* 2 se desarrollaron las funcionalidades orientadas a la atención de consultas institucionales mediante la implementación de una arquitectura *Retrieval-Augmented Generation* (RAG). El objetivo principal de esta iteración fue permitir que el chatbot pudiera responder preguntas relacionadas con reglamentos, procesos académicos,

normativas universitarias y otra información institucional utilizando fuentes verificadas de conocimiento.

Como primera actividad se realizó la recopilación y organización de documentación institucional proveniente de fuentes oficiales de la universidad. Esta información fue clasificada y preparada para ser utilizada como base de conocimiento del chatbot, garantizando que las respuestas generadas estuvieran respaldadas por información válida y actualizada.

Posteriormente, se implementó el procesamiento documental, etapa en la cual los documentos institucionales fueron segmentados en fragmentos de información de tamaño controlado. Este procedimiento facilitó la organización del contenido para su posterior recuperación mediante técnicas de búsqueda semántica. La segmentación fue diseñada para preservar el contexto de cada fragmento, mejorando la relevancia y precisión de la información recuperada durante las consultas realizadas por los usuarios.

De manera complementaria, se incorporó un mecanismo de recuperación léxica basado en coincidencias textuales, el cual actúa como respaldo cuando la búsqueda semántica no identifica suficiente contexto relevante. Esta estrategia híbrida fortaleció la capacidad de recuperación de información del sistema, permitiendo responder consultas institucionales específicas mediante la combinación de similitud semántica y coincidencias directas sobre el contenido documental.

Una vez procesados los documentos, se incorporó un mecanismo de generación de *embeddings* encargado de transformar cada fragmento textual en representaciones vectoriales. Estas representaciones permitieron modelar semánticamente el contenido institucional y almacenarlo dentro de un índice vectorial, facilitando la recuperación de información basada en significado contextual en lugar de coincidencias exactas de palabras.

Para gestionar el almacenamiento y la búsqueda de los vectores generados, se utilizó FAISS (*Facebook AI Similarity Search*), una biblioteca especializada en indexación y búsqueda por similitud. Esta tecnología permitió localizar de manera eficiente los fragmentos documentales más relevantes para cada consulta realizada por el usuario, optimizando el rendimiento del proceso de recuperación contextual dentro de la arquitectura RAG.

Como parte central del *Sprint*, se desarrolló la arquitectura *Retrieval-Augmented Generation* (RAG), la cual permitió integrar procesos de recuperación contextual con capacidades avanzadas de generación de lenguaje natural. Gracias a esta arquitectura, el sistema pudo localizar información relevante dentro de la base documental antes de generar una respuesta para el usuario.

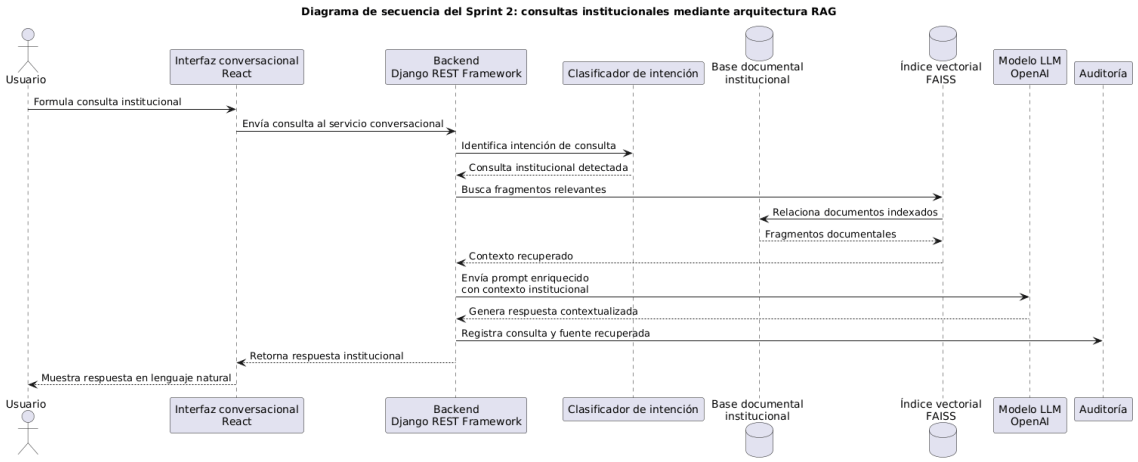
De forma complementaria, se realizó la configuración e integración del modelo de lenguaje de gran escala (LLM) utilizado por el chatbot. Esta integración permitió interpretar consultas formuladas en lenguaje natural y generar respuestas conversacionales contextualizadas utilizando la información recuperada desde la base de conocimiento institucional.

Asimismo, se implementó un mecanismo de construcción dinámica de *prompts*, mediante el cual la consulta realizada por el usuario se combina con los fragmentos documentales recuperados durante el proceso de búsqueda semántica. Este enfoque permitió proporcionar al modelo de lenguaje el contexto necesario para generar respuestas más precisas y alineadas con la información institucional disponible.

Finalmente, se ejecutaron pruebas funcionales orientadas a validar la precisión de la recuperación documental, la coherencia de las respuestas generadas y el correcto funcionamiento de la arquitectura RAG integrada con el modelo LLM.

Como resultado de este *Sprint* se obtuvo un sistema capaz de responder consultas institucionales utilizando información verificable proveniente de documentos oficiales, mejorando significativamente la confiabilidad y precisión de las respuestas generadas por el chatbot académico.

Ilustración 4 Diagrama de Secuencia Sprint 2.



2.3.1.2.1. Base de Conocimiento del Sistema

La base de conocimiento constituye el repositorio documental utilizado por la arquitectura RAG para proporcionar respuestas fundamentadas en información institucional. Su función principal consistió en organizar documentos previamente curados, segmentados e indexados, de manera que el sistema pudiera recuperar fragmentos relevantes antes de generar una respuesta mediante el modelo de lenguaje.

En el sistema desarrollado, la base de conocimiento estuvo conformada por documentos institucionales en formato `.txt` ubicados en el directorio documental del *backend*. Estos archivos fueron seleccionados considerando su utilidad para responder consultas frecuentes sobre información institucional, servicios del campus, mallas de Ingeniería, horarios, prerrequisitos y procesos académicos generales. De esta manera, el chatbot pudo responder consultas públicas sin acceder a información privada de estudiantes.

Para evitar ruido documental durante la recuperación, no todos los archivos generados durante el proceso de recopilación fueron utilizados como fuentes activas del RAG. Archivos como `scraped_web_content.txt` y `pucesi_crawl_index.txt` se conservaron como evidencia del rastreo y apoyo documental, pero fueron excluidos del índice por defecto para evitar contenido repetido, histórico o no depurado. De igual forma, archivos de prueba no fueron considerados como parte de la base institucional activa.

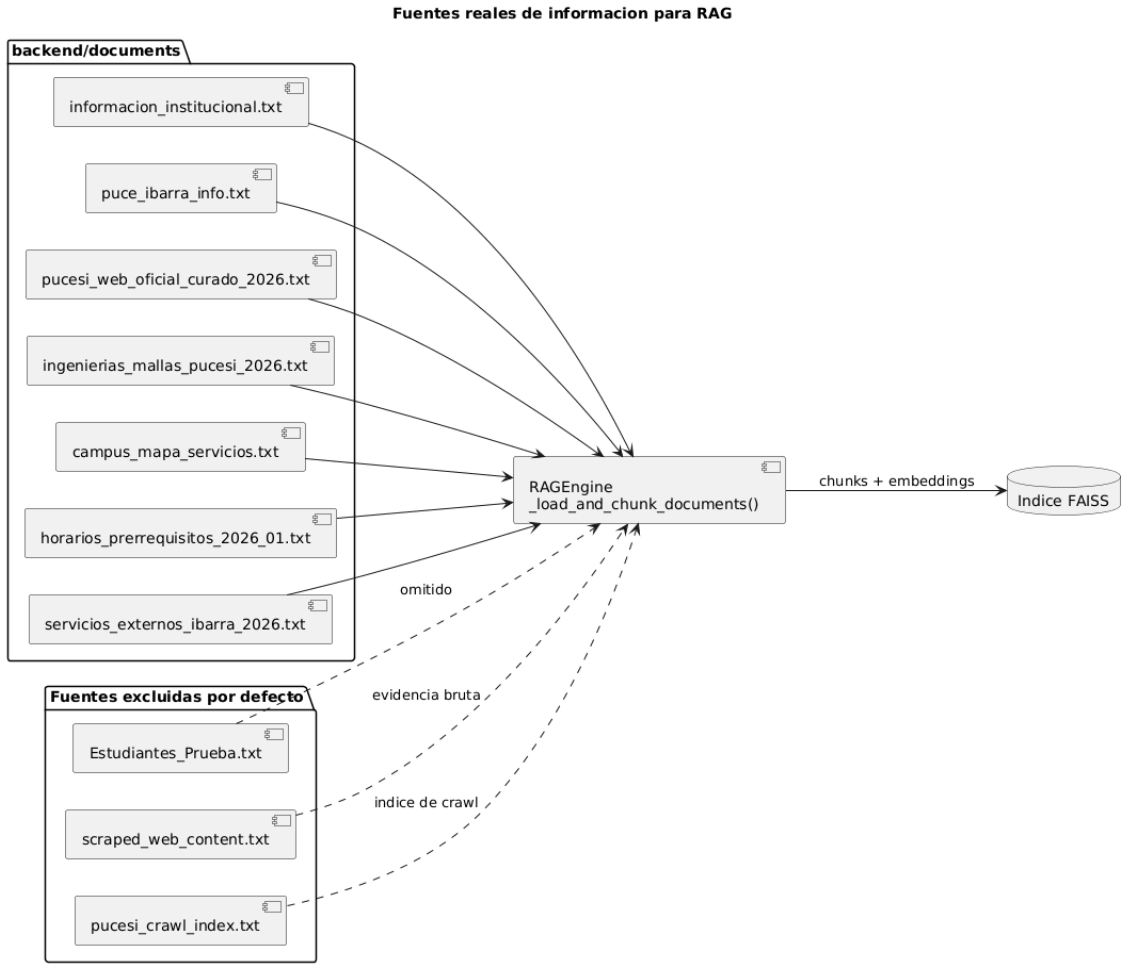
2.3.1.2.1.1. Fuentes de Información

Las fuentes activas utilizadas por el RAG incluyeron documentos relacionados con información institucional de la PUCE-I, datos complementarios de la sede, contenido curado de la web oficial, mallas de las carreras de Ingeniería, servicios externos cercanos,

mapa temporal del campus, servicios confirmados, horarios y prerrequisitos del periodo académico definido para el prototipo. Esta selección permitió que el chatbot respondiera consultas institucionales generales con base en información previamente organizada, evitando depender únicamente del conocimiento interno del modelo generativo.

La separación entre fuentes activas y archivos de evidencia permitió mejorar la calidad de la recuperación documental. Por esta razón, el proceso de indexación priorizó documentos depurados y omitió archivos brutos, históricos o de prueba que podían afectar la precisión de las respuestas generadas por el sistema.

Ilustración 5 Fuentes de información.



Con el propósito de garantizar la calidad y confiabilidad de la información utilizada por el chatbot, los documentos incorporados a la base de conocimiento fueron obtenidos exclusivamente de fuentes institucionales oficiales de la Pontificia Universidad Católica del Ecuador Sede Ibarra. Entre estos documentos se incluyeron reglamentos académicos, procedimientos institucionales, información de servicios universitarios y material de

orientación estudiantil. Esta selección permitió asegurar que las respuestas generadas por el sistema se fundamentaran en contenidos válidos, verificables y alineados con la información oficial disponible para los estudiantes.

2.3.1.2.1.2. Organización de Documentos

Los documentos incorporados a la base de conocimiento fueron organizados de acuerdo con categorías temáticas relacionadas con procesos académicos, servicios estudiantiles, normativa institucional y procedimientos administrativos.

Esta organización facilitó la administración del conocimiento y mejoró la eficiencia de los mecanismos de recuperación utilizados por la arquitectura RAG.

2.3.1.2.1.3. Actualización de Contenido

La actualización de contenido es un proceso continuo, orientado a mantener vigente la información disponible para el chatbot. La incorporación de nuevos documentos o la modificación de contenidos existentes requiere únicamente la actualización de la base de conocimiento y la regeneración de los índices correspondientes.

Este enfoque permite que el sistema evolucione de manera flexible conforme cambian los procedimientos institucionales, garantizando que las respuestas proporcionadas a los usuarios reflejen la información más reciente disponible.

2.3.1.2.2. Configuración e integración del modelo LLM

El chatbot académico inteligente fue diseñado utilizando una arquitectura cliente-servidor basada en un *frontend* desarrollado en React y un *backend* implementado en Django. La comunicación inteligente del sistema se estableció mediante la integración de modelos de lenguaje de gran escala (LLMs) proporcionados por OpenAI a través de su API.

Para el procesamiento de consultas en lenguaje natural, el *backend* realiza solicitudes al modelo LLM enviando el contexto recuperado desde la arquitectura *Retrieval-Augmented Generation* (RAG), junto con la consulta realizada por el usuario. Esto permitió generar respuestas contextualizadas utilizando información institucional previamente almacenada.

La integración del modelo de lenguaje fue complementada mediante mecanismos de construcción dinámica de contexto. Dependiendo del estado de autenticación del usuario,

el sistema incorporó información académica personalizada obtenida desde la base de datos académica implementada para el prototipo o contexto documental recuperado mediante la arquitectura RAG. Esta estrategia permitió generar respuestas diferenciadas para consultas privadas relacionadas con horarios, actividades, calificaciones y perfil académico, así como para consultas institucionales basadas en documentación oficial de la universidad.

La parametrización del modelo incluyó elementos como:

- modelo LLM utilizado;
- temperatura de generación;
- límite de *tokens*;
- mensajes del sistema;
- contexto recuperado mediante *embeddings*;
- procesamiento de *prompts* académicos.

Asimismo, el sistema implementó una capa de recuperación documental basada en *embeddings* vectoriales, permitiendo localizar fragmentos relevantes de información antes de enviar la consulta al modelo generativo.

El procesamiento de lenguaje natural implementado en el sistema no se basa en el entrenamiento de modelos propios desde cero, sino en técnicas de ingeniería de *prompts* contextuales. Este enfoque permitió controlar el comportamiento del modelo LLM mediante instrucciones específicas relacionadas con privacidad, temporalidad académica, personalización de respuestas y restricciones de generación de información.

El diseño del *prompt* maestro permitió establecer reglas de comportamiento para el chatbot, diferenciando consultas académicas actuales, historiales académicos y consultas institucionales generales. Además, se implementaron restricciones orientadas a evitar la generación de información inexistente, priorizando únicamente datos recuperados desde fuentes verificadas dentro de la arquitectura RAG y la base de datos académica.

Adicionalmente, se definió una jerarquía de utilización de fuentes de información. El sistema prioriza la información académica estructurada cuando la consulta corresponde a datos personalizados del estudiante. Para las consultas institucionales, se utiliza preferentemente el contexto recuperado mediante la arquitectura RAG. Cuando no existe información suficiente para generar una respuesta confiable, el modelo proporciona

respuestas controladas indicando la ausencia de información verificable, reduciendo el riesgo de producir contenido incorrecto o no respaldado por las fuentes disponibles.

La ilustración 6 muestra el fragmento de código encargado de inicializar el cliente OpenAI utilizando la clave API configurada dentro del entorno del sistema.

Ilustración 6 Configuración de conexión entre Django y OpenAI mediante API.

```
34      def get_client(self):
35          """Inicializa el cliente OpenAI de forma lazy."""
36          if self._client is None:
37              from openai import OpenAI
38              self._client = OpenAI(api_key=settings.OPENAI_API_KEY)
39          return self._client
```

```
31      def get_client(self):
32          """Obtiene el cliente OpenAI configurado."""
33          from openai import OpenAI
34          return OpenAI(api_key=settings.OPENAI_API_KEY)
```

Para la generación de respuestas se seleccionó el modelo GPT-4o-mini proporcionado por OpenAI. La elección se fundamentó en su equilibrio entre capacidad de razonamiento, velocidad de respuesta y costos operativos, características que resultan adecuadas para entornos académicos donde se requiere procesar múltiples consultas de manera eficiente. Adicionalmente, se analizaron otras alternativas disponibles en el mercado, como GPT-4o, Claude, Gemini y Llama. Sin embargo, GPT-4o-mini presentó una relación favorable entre rendimiento y consumo de recursos para las necesidades del prototipo desarrollado, permitiendo generar respuestas contextualizadas con una latencia adecuada sin incrementar significativamente los costos de operación.

La arquitectura RAG fue implementada mediante un motor de recuperación contextual encargado de buscar información relevante dentro del repositorio documental institucional antes de generar la respuesta final del chatbot. Este proceso combina recuperación semántica basada en *embeddings* vectoriales con generación de respuestas mediante el modelo LLM.

La ilustración 7 presenta el flujo de integración entre el servicio principal del chatbot y el motor RAG, donde primero se recupera el contexto más relevante y posteriormente este es enviado al modelo OpenAI para generar respuestas contextualizadas. Adicionalmente, el motor RAG realiza procesos de vectorización y búsqueda por similitud utilizando *embeddings* para localizar documentos relacionados con la consulta del usuario.

Para la indexación y recuperación semántica de documentos se seleccionó FAISS (*Facebook AI Similarity Search*), una biblioteca especializada en búsquedas vectoriales de alta eficiencia. La elección de esta tecnología se debió a su capacidad para gestionar representaciones vectoriales y realizar búsquedas de similitud sobre colecciones documentales en tiempos reducidos. Considerando que el prototipo trabajó con una base documental institucional de tamaño medio, FAISS proporcionó un mecanismo adecuado para recuperar fragmentos relevantes que posteriormente fueron utilizados como contexto dentro de la arquitectura *Retrieval-Augmented Generation* (RAG).

Ilustración 7 Implementación de recuperación contextual mediante arquitectura RAG.

```
168         # 3. Búsqueda RAG (siempre activa)
169         rag_results = rag_engine.search(user_message)
170         rag_context = rag_engine.format_context(rag_results)
171
```

```
186         messages.append({
187             "role": "system",
188             "content": f"INFORMACIÓN DE DOCUMENTOS INSTITUCIONALES PUCESI
189         })
```

```
210         # Generar embedding de la query
211         query_embedding = np.array([self._get_embedding(query)], dtype=np
212         faiss.normalize_L2(query_embedding)
213
214         # Buscar en el índice
215         scores, indices = self._index.search(query_embedding, top_k)
216
```

2.3.1.2.3. Procesamiento documental y recuperación contextual

Con el propósito de construir la base de conocimiento institucional, los documentos seleccionados para el RAG fueron sometidos a un proceso de preparación, depuración y segmentación. Esta etapa permitió transformar la información institucional en unidades de texto manejables, evitando que el modelo de lenguaje recibiera documentos completos o contenido no relacionado con la consulta del usuario.

Durante el procesamiento documental, el sistema trabajó únicamente con fuentes institucionales activas y documentos previamente curados. Los archivos brutos generados durante el rastreo web se conservaron como evidencia técnica, pero no fueron incorporados por defecto al índice vectorial cuando podían contener ruido, información repetida o contenido no depurado.

Posteriormente, cada fragmento documental fue convertido en una representación vectorial mediante *embeddings* y almacenado en un índice FAISS. Cuando el usuario realiza una consulta institucional, el sistema transforma la pregunta en una representación compatible, ejecuta la búsqueda semántica sobre el índice y recupera los fragmentos más relevantes. Esta información es incorporada al contexto enviado al modelo de lenguaje, permitiendo generar respuestas fundamentadas en documentos institucionales y reduciendo el riesgo de producir información no verificada.

2.3.1.2.4. Arquitectura RAG

La arquitectura *Retrieval-Augmented Generation* (RAG) implementada en el chatbot académico fue diseñada con el propósito de complementar las capacidades del modelo de lenguaje mediante la recuperación de información institucional previamente almacenada y validada. Este enfoque permite que las respuestas generadas se fundamenten en contenido específico de la institución, reduciendo la probabilidad de generar información incorrecta o desactualizada.

El funcionamiento de la arquitectura se basa en la recuperación de fragmentos documentales relevantes para cada consulta realizada por el usuario. Dichos fragmentos son incorporados como contexto dentro de la solicitud enviada al modelo de lenguaje, permitiendo generar respuestas contextualizadas y alineadas con la información institucional disponible.

Para optimizar la recuperación de información, los documentos institucionales fueron transformados en representaciones vectoriales almacenadas dentro de un índice FAISS. Asimismo, se incorporó un mecanismo complementario de recuperación léxica que permite localizar información relevante mediante coincidencia textual cuando no es posible utilizar recuperación semántica.

La configuración implementada considera parámetros relacionados con el procesamiento documental, generación de *embeddings*, recuperación semántica y generación de respuestas. Los principales componentes y criterios técnicos utilizados en la arquitectura RAG se presentan en la Tabla 6.

Tabla 6 Parámetros de configuración de la arquitectura RAG

Parámetro	Valor

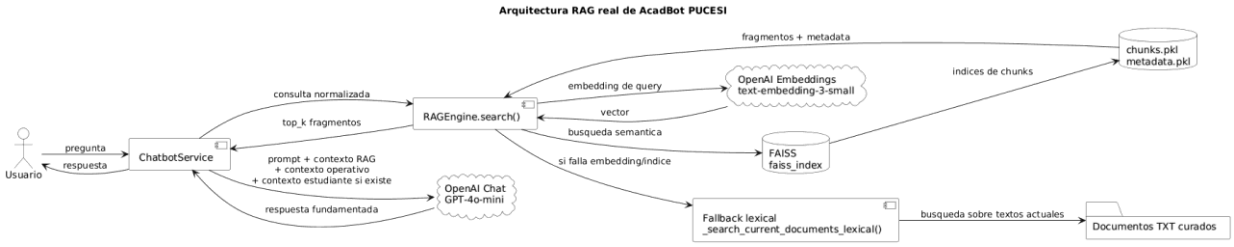
Fuente documental activa	Documentos institucionales curados en formato .txt
Archivos excluidos por defecto	Archivos de prueba, evidencia bruta de *scraping* y registros de rastreo no depurados
Técnica de segmentación	Fragmentación textual de documentos institucionales
Tamaño de fragmento	500 caracteres
Solapamiento entre fragmentos	50 caracteres
Modelo de *embeddings*	*text-embedding-3-small*
Motor vectorial	FAISS
Estrategia principal de recuperación	Búsqueda semántica por similitud vectorial
Estrategia complementaria	Búsqueda léxica de respaldo
Cantidad de fragmentos recuperados	Top-K = 5
Contexto enviado al LLM	Fragmentos documentales relevantes recuperados
Modelo LLM	GPT-4o-mini
Idioma de respuesta	Español
Criterio de respuesta	No inventar información cuando no exista contexto suficiente
Endpoints relacionados	*/api/rag/build/ y /api/rag/search/*

La configuración presentada describe los componentes reales utilizados en la arquitectura RAG del sistema. Como se muestra en la Ilustración 8, el proceso inicia con la consulta realizada por el usuario, seguida por la recuperación de información desde la base documental indexada mediante mecanismos de búsqueda semántica y léxica. Posteriormente, los fragmentos recuperados son incorporados al contexto enviado al modelo de lenguaje para la generación de respuestas.

Esta estructura permitió recuperar información institucional desde documentos validados, incorporar fragmentos relevantes dentro del contexto del modelo de lenguaje y generar respuestas en español con menor riesgo de producir información no verificada. Además,

la búsqueda léxica de respaldo permitió mantener la recuperación de información en consultas específicas cuando la búsqueda semántica no resultaba suficiente.

Ilustración 8 Arquitectura general del mecanismo Retrieval-Augmented Generation (RAG).



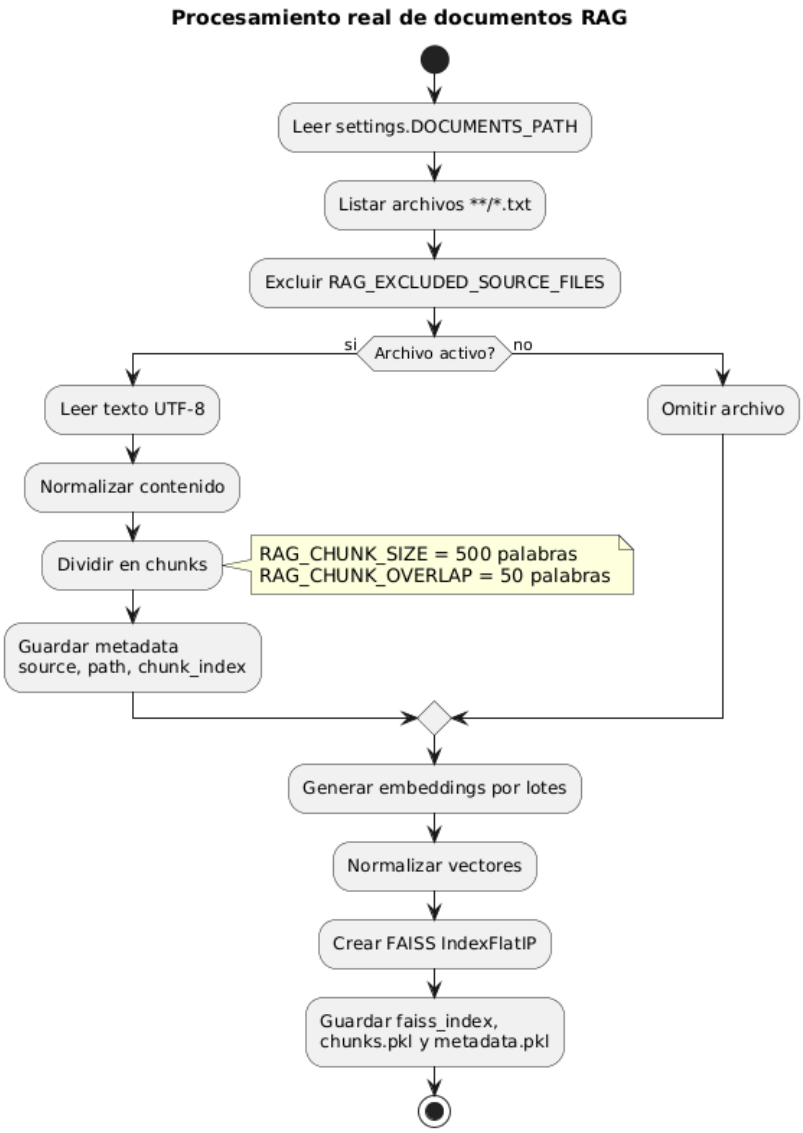
2.3.1.2.4.1. Procesamiento de Documentos

La primera etapa de la arquitectura RAG correspondió al procesamiento de documentos institucionales. Como se muestra en la Ilustración 9, el proceso inició con la recopilación y selección de fuentes relacionadas con información institucional, servicios del campus, mallas académicas, horarios, prerrequisitos y procesos generales de orientación estudiantil.

Una vez seleccionadas las fuentes activas, el contenido textual fue depurado y segmentado mediante una estrategia de fragmentación documental. Para el prototipo se emplearon fragmentos de 500 caracteres con un solapamiento de 50 caracteres entre segmentos consecutivos, lo que permitió conservar continuidad semántica entre fragmentos cercanos y mejorar la recuperación posterior de información.

Este procesamiento permitió transformar documentos extensos en unidades de información más precisas, facilitando que el sistema recuperara únicamente los fragmentos relacionados con la consulta del usuario. De esta manera, el chatbot no dependió únicamente del conocimiento interno del modelo generativo, sino de información previamente organizada dentro de la base de conocimiento institucional.

Ilustración 9 Procesamiento y segmentación de documentos institucionales.



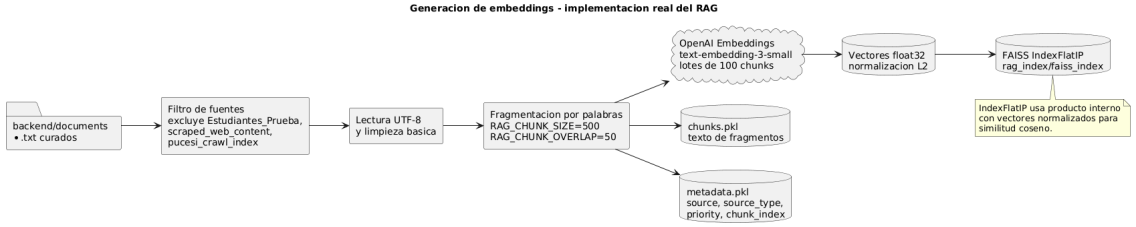
2.3.1.2.4.2. Generación de *Embeddings*

Después del procesamiento documental, cada fragmento generado fue transformado en una representación vectorial utilizando el modelo *text-embedding-3-small* de OpenAI. Como se observa en la Ilustración 10, cada segmento de texto fue convertido en un vector numérico capaz de representar su significado semántico dentro del espacio vectorial.

Los *embeddings* permitieron comparar matemáticamente la consulta realizada por el usuario con los fragmentos almacenados en la base de conocimiento. Esta representación resultó fundamental para identificar coincidencias semánticas, incluso cuando la pregunta no utilizó las mismas palabras presentes en el documento institucional.

Las representaciones vectoriales fueron almacenadas en FAISS, utilizado como motor vectorial para ejecutar búsquedas por similitud de manera eficiente. Esta elección permitió recuperar fragmentos relevantes en tiempos adecuados para el funcionamiento del chatbot dentro de una plataforma web interactiva.

Ilustración 10 Conversión de fragmentos documentales en representaciones vectoriales.



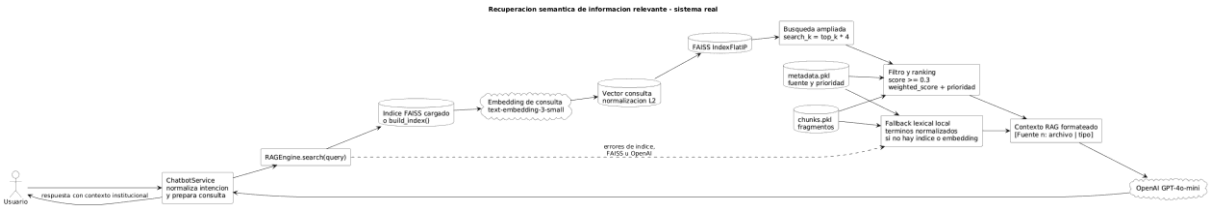
2.3.1.2.4.3. Recuperación semántica de información relevante

Una vez construido el índice vectorial, el sistema pudo recuperar información institucional a partir de las consultas realizadas por los usuarios. Como se muestra en la Ilustración 11, la pregunta ingresada fue procesada y transformada en una representación vectorial, la cual fue comparada con los fragmentos almacenados en FAISS.

El mecanismo principal de recuperación se basó en búsqueda semántica por similitud vectorial. Para cada consulta, el sistema recuperó los fragmentos más relevantes de acuerdo con la configuración Top-K definida para el prototipo. Estos fragmentos fueron utilizados como contexto documental antes de generar la respuesta final.

De manera complementaria, se incorporó una búsqueda léxica de respaldo. Este mecanismo permitió localizar información mediante coincidencias textuales cuando la búsqueda semántica no resultaba suficiente o cuando la consulta contenía términos específicos, nombres de servicios, ubicaciones o expresiones institucionales concretas. La combinación de recuperación semántica y respaldo léxico fortaleció la precisión del sistema frente a diferentes formas de preguntar.

Ilustración 11 Recuperación semántica de información relevante.



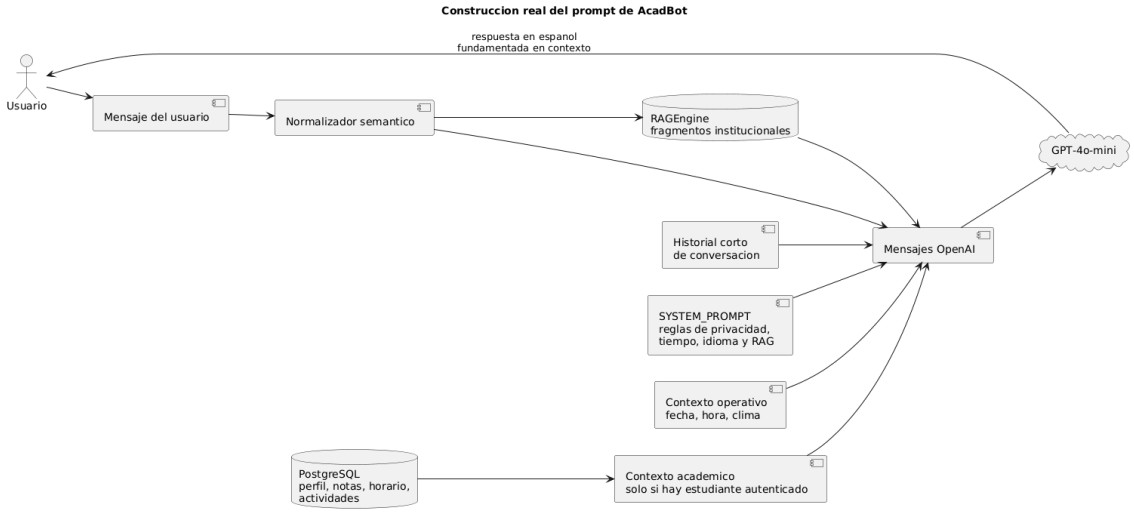
2.3.1.2.4.4. Construcción del *Prompt* enriquecido

Después de recuperar los fragmentos relevantes, el sistema construyó un *prompt* enriquecido para el modelo de lenguaje. Como se observa en la Ilustración 12, este *prompt* integró la consulta del usuario, el contexto documental recuperado mediante RAG y las instrucciones de comportamiento definidas para el chatbot académico.

El *prompt* fue diseñado para que el modelo respondiera en español, de forma clara y directa, utilizando únicamente la información disponible en el contexto proporcionado. Cuando la base documental no contenía información suficiente para responder una consulta, el sistema debía evitar inventar datos y orientar al usuario indicando que no existía contexto suficiente o que debía confirmar la información con la instancia institucional correspondiente.

En el modo público, el *prompt* incorporó únicamente información institucional general recuperada desde el RAG. En el modo autenticado, el sistema pudo complementar la respuesta con contexto académico del estudiante, como horarios, asignaturas, calificaciones, actividades y notificaciones, siempre que la consulta correspondiera a información personalizada y existiera una sesión válida. Esta separación permitió proteger los datos privados del estudiante y mantener respuestas diferenciadas según el tipo de usuario.

Ilustración 12 Construcción del prompt enriquecido para el modelo de lenguaje.



2.3.1.3. *Sprint* 3: Reconocimiento de Voz

Durante el *Sprint* 3 se incorporó la funcionalidad de reconocimiento de voz con el objetivo de ampliar las formas de interacción disponibles para los estudiantes. Esta

funcionalidad fue desarrollada como un mecanismo complementario a la entrada tradicional mediante texto, permitiendo que las consultas académicas e institucionales puedan realizarse utilizando comandos de voz.

El desarrollo inició con la implementación de los mecanismos de captura de audio desde la interfaz web, aprovechando las capacidades del navegador para acceder al micrófono del dispositivo. Una vez obtenida la entrada de voz, se integró un servicio de conversión encargado de transformar las expresiones habladas en texto procesable por el sistema.

Adicionalmente, se incorporaron servicios de inteligencia artificial especializados en reconocimiento y síntesis de voz, permitiendo optimizar la precisión de las transcripciones y mejorar la interacción conversacional. Estas tecnologías facilitaron el procesamiento de consultas formuladas en español y la generación de respuestas habladas con una pronunciación adecuada para términos académicos e institucionales utilizados dentro del entorno universitario.

Posteriormente, se desarrolló la lógica necesaria para interpretar las consultas convertidas a texto y enviarlas al motor conversacional del chatbot. Esta integración permitió reutilizar las funcionalidades implementadas en los *Sprint* anteriores, de modo que las consultas por voz siguieran el mismo flujo de procesamiento utilizado para las consultas escritas.

De forma complementaria, se incorporó un servicio de síntesis de voz encargado de convertir las respuestas generadas por el chatbot en audio reproducible para el usuario. Este mecanismo permitió que la interacción no se limitara únicamente al reconocimiento de voz, sino que incluyera también la devolución de respuestas habladas, proporcionando una experiencia conversacional más natural y cercana a la comunicación humana.

Adicionalmente, se implementaron mecanismos de validación orientados a gestionar posibles errores durante la captura de audio, mejorar la experiencia de interacción y garantizar que las consultas reconocidas fueran procesadas correctamente antes de generar una respuesta.

Finalmente, se realizaron pruebas multimodales que permitieron verificar el funcionamiento conjunto de las modalidades de entrada por texto y voz, evaluando la correcta conversión de audio, la interpretación de consultas y la generación de respuestas coherentes por parte del sistema.

Como resultado de este *Sprint* se obtuvo un chatbot capaz de recibir consultas mediante comandos de voz, transformarlas en texto e integrarlas al flujo conversacional existente, proporcionando una experiencia de interacción más accesible y flexible para los estudiantes.

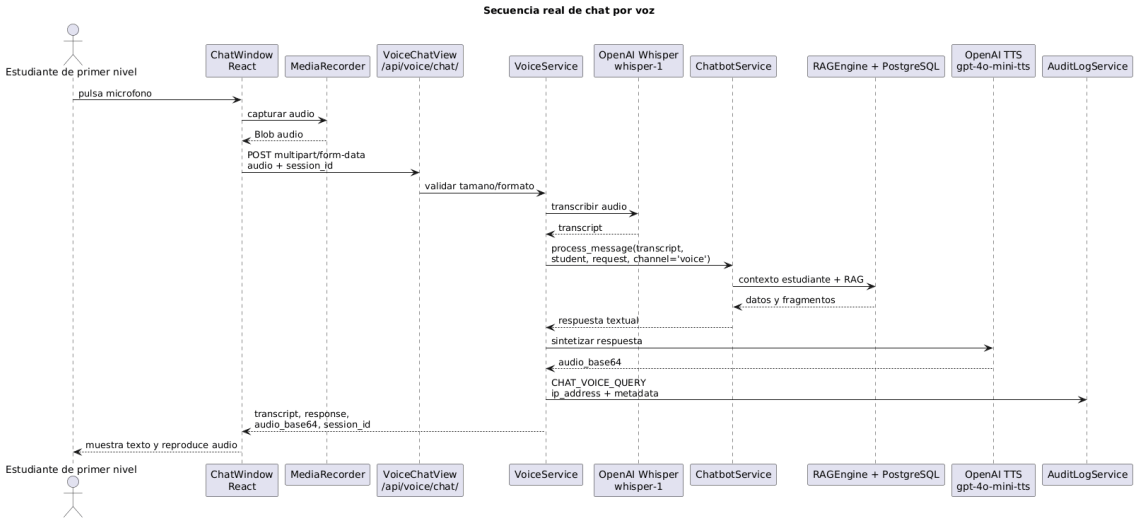
La funcionalidad de reconocimiento de voz fue implementada mediante una arquitectura de procesamiento de audio integrada con los servicios de inteligencia artificial de OpenAI. El flujo inicia cuando el usuario interactúa con el componente de micrófono disponible en la interfaz desarrollada en React. A través de las APIs nativas del navegador (*getUserMedia* y *MediaRecorder*), la aplicación captura el audio emitido por el usuario y lo almacena temporalmente en formato *WebM*.

Una vez finalizada la grabación, el archivo de audio es enviado al *backend* desarrollado en Django REST Framework mediante una solicitud HTTP al servicio de voz. En el servidor, el audio es validado y procesado utilizando tecnología *Speech-to-Text* (STT) basada en el modelo Whisper de OpenAI, encargado de convertir el contenido hablado en texto interpretable por el sistema.

Posteriormente, el texto obtenido es enviado al motor conversacional del chatbot, el cual determina si la consulta corresponde a información académica personalizada o a información institucional gestionada mediante la arquitectura *Retrieval-Augmented Generation* (RAG). Después de generar la respuesta textual, el sistema utiliza un servicio *Text-to-Speech* (TTS) para sintetizar la respuesta en formato de audio, permitiendo que el usuario reciba una retroalimentación hablada además de la respuesta escrita.

Este proceso convierte la interacción por voz en una extensión natural del flujo conversacional implementado, manteniendo la integración con los mecanismos de autenticación, recuperación de información académica y generación de respuestas basadas en inteligencia artificial.

Ilustración 13 Diagrama de Secuencia Sprint 3.



2.3.1.3.1. Interacción mediante Texto

La interacción mediante texto constituye el mecanismo principal de comunicación entre el usuario y el chatbot académico. A través del campo de entrada, los estudiantes pueden formular consultas relacionadas con horarios, asignaturas, calificaciones y procesos institucionales utilizando lenguaje natural.

Cuando el usuario envía una consulta, el sistema procesa la información y la envía al *backend* mediante servicios REST. Posteriormente, el chatbot analiza la solicitud, recupera el contexto correspondiente y genera una respuesta que es presentada dentro del historial de conversación.

Este mecanismo permite mantener una comunicación fluida y favorece la consulta rápida de información académica sin necesidad de navegar por múltiples secciones del sistema.

2.3.1.3.2. Interacción mediante Voz

Con el propósito de mejorar la accesibilidad y la experiencia de usuario, el chatbot incorpora funcionalidades de interacción mediante voz. Esta característica permite capturar consultas habladas utilizando el micrófono del dispositivo y convertirlas automáticamente en texto para su posterior procesamiento.

Durante la grabación, la interfaz proporciona indicadores visuales que permiten al usuario identificar que el sistema se encuentra escuchando una consulta. Una vez finalizada la captura, el audio es enviado al *backend* para realizar el reconocimiento de voz y generar la respuesta correspondiente.

Esta funcionalidad facilita la interacción con el sistema en situaciones donde el ingreso manual de texto resulta menos conveniente para el usuario.

2.3.1.3.3. Generación de Respuestas Habladas

Una vez generada la respuesta textual por el chatbot, el sistema ejecuta un proceso de síntesis de voz encargado de transformar el contenido textual en audio. Este procedimiento permite que el usuario escuche la respuesta producida por el asistente sin necesidad de leerla manualmente.

La respuesta generada es convertida a formato de audio y enviada nuevamente a la interfaz web, donde puede ser reproducida automáticamente o bajo demanda del usuario. Esta funcionalidad complementa el reconocimiento de voz implementado previamente y permite establecer un ciclo completo de interacción basado en entrada y salida de voz.

La incorporación de respuestas habladas contribuyó a mejorar la accesibilidad del sistema, facilitando la interacción con estudiantes que prefieren utilizar comandos de voz o que requieren alternativas al uso tradicional del teclado.

2.3.1.4. *Sprint* 4: Interfaz Conversacional

Durante el *Sprint* 4 se desarrolló la interfaz conversacional inteligente encargada de facilitar la interacción entre el estudiante y el chatbot académico. El objetivo principal de esta iteración fue proporcionar una experiencia de usuario intuitiva, dinámica y accesible que permitiera utilizar todas las funcionalidades implementadas en los *Sprint* anteriores.

Como primera actividad se diseñó la estructura visual de la interfaz utilizando React, definiendo los componentes necesarios para la gestión de conversaciones, visualización de mensajes y control de interacciones. Se incorporó un chatbot flotante accesible desde cualquier sección de la aplicación, permitiendo que los usuarios pudieran realizar consultas sin abandonar la página en la que se encontraban.

Posteriormente, se desarrolló el flujo conversacional encargado de gestionar el intercambio de mensajes entre el usuario y el sistema. Este componente fue diseñado para mantener el historial de conversaciones, mostrar mensajes enviados y recibidos, así como organizar la información generada durante cada sesión de uso.

Además, se implementaron mecanismos de gestión de estados que permiten informar al usuario sobre el procesamiento de sus consultas mediante indicadores visuales de carga

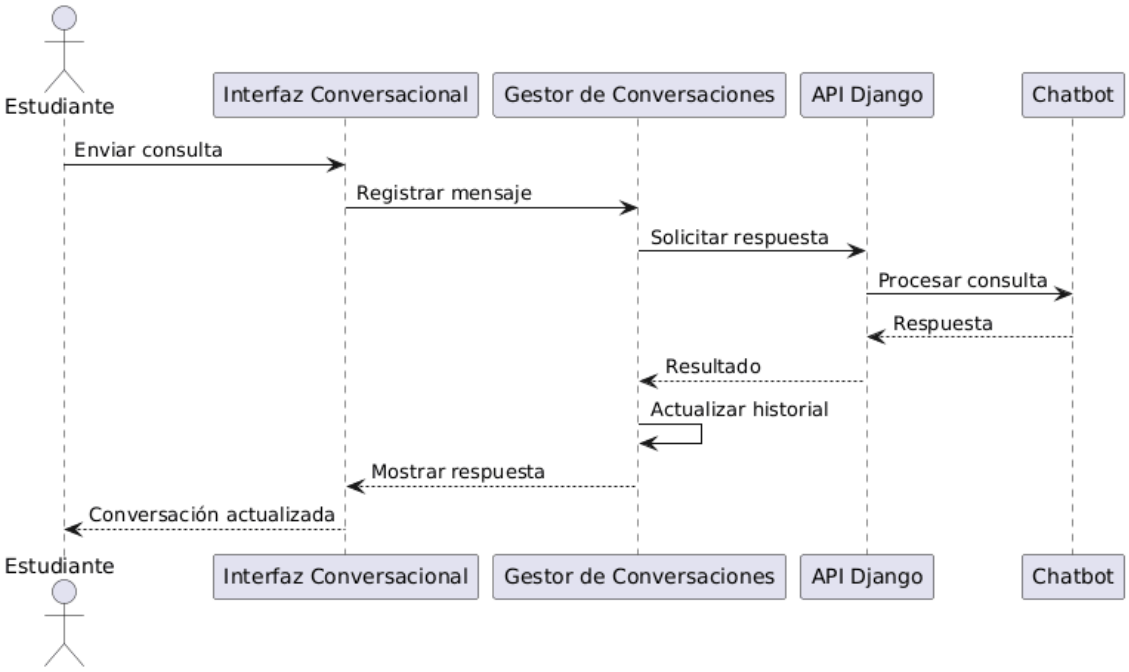
y espera. Esto contribuyó a mejorar la percepción de respuesta del sistema y a proporcionar una experiencia de uso más fluida.

Como complemento, se integraron las funcionalidades desarrolladas en los *Sprint* anteriores, incluyendo las consultas académicas personalizadas, las consultas institucionales mediante arquitectura RAG y el reconocimiento de voz. De esta manera, la interfaz se convirtió en el punto central de acceso a todos los servicios ofrecidos por el chatbot.

Finalmente, se ejecutaron pruebas de usabilidad y experiencia de usuario orientadas a validar la navegación, la presentación de respuestas, la accesibilidad de los controles y la estabilidad general de la interfaz.

Como resultado de este *Sprint* se obtuvo una interfaz conversacional completamente funcional, capaz de integrar texto, voz y servicios inteligentes dentro de una experiencia de usuario unificada y orientada a las necesidades de los estudiantes de primer nivel.

Ilustración 14 Diagrama de Secuencia Sprint 4.



2.3.1.4.1. Implementación de páginas públicas y privadas

Se desarrolló un esquema de navegación basado en rutas públicas y protegidas. Las páginas públicas permiten acceder a información institucional y funcionalidades generales del chatbot, mientras que las páginas privadas requieren autenticación para

acceder a información académica personalizada. Esta separación permitió proteger los datos del estudiante y mejorar la organización funcional de la plataforma.

2.3.1.4.2. Desarrollo del *dashboard* académico del estudiante

Se implementó un panel académico personalizado que permite visualizar información relacionada con materias, actividades, horarios, notificaciones y consultas académicas. Este *dashboard* actúa como el principal entorno de interacción entre el estudiante y el sistema, integrando la información obtenida desde la base de datos académica y los servicios conversacionales.

2.3.1.4.3. Desarrollo del panel docente como componente de apoyo

Se construyó una interfaz de soporte para docentes, orientada a la gestión de información académica necesaria para alimentar el flujo del estudiante, como materias, horarios y actividades asociadas a sus cursos. Esta funcionalidad no amplió el alcance principal de la tesis, sino que actuó como un componente operativo de apoyo para mantener disponible la información consultada posteriormente por el estudiante autenticado.

2.3.1.4.4. Integración del chatbot conversacional

Se integró el componente principal del chatbot dentro de la interfaz web mediante una ventana conversacional interactiva. Este componente permite realizar consultas académicas personalizadas, consultas institucionales basadas en arquitectura RAG y consultas mediante voz, centralizando todas las capacidades inteligentes del sistema en una única interfaz de usuario.

2.3.1.4.5. Gestión de sesiones y experiencia de usuario

Se implementó un mecanismo de gestión de sesiones basado en autenticación segura, permitiendo mantener el estado del usuario durante su interacción con la plataforma. Asimismo, se incorporaron elementos de retroalimentación visual, indicadores de carga, mensajes informativos y notificaciones que contribuyen a mejorar la experiencia de uso y la accesibilidad del sistema.

2.3.1.5. *Sprint* 5: Seguridad y registros de eventos

Durante el *Sprint* 5 se implementaron los mecanismos de seguridad, autenticación y los registros de eventos necesarios para proteger el acceso a la información académica personalizada del estudiante. Esta iteración tuvo como finalidad garantizar que las consultas privadas, como horarios, calificaciones, actividades y notificaciones, solo

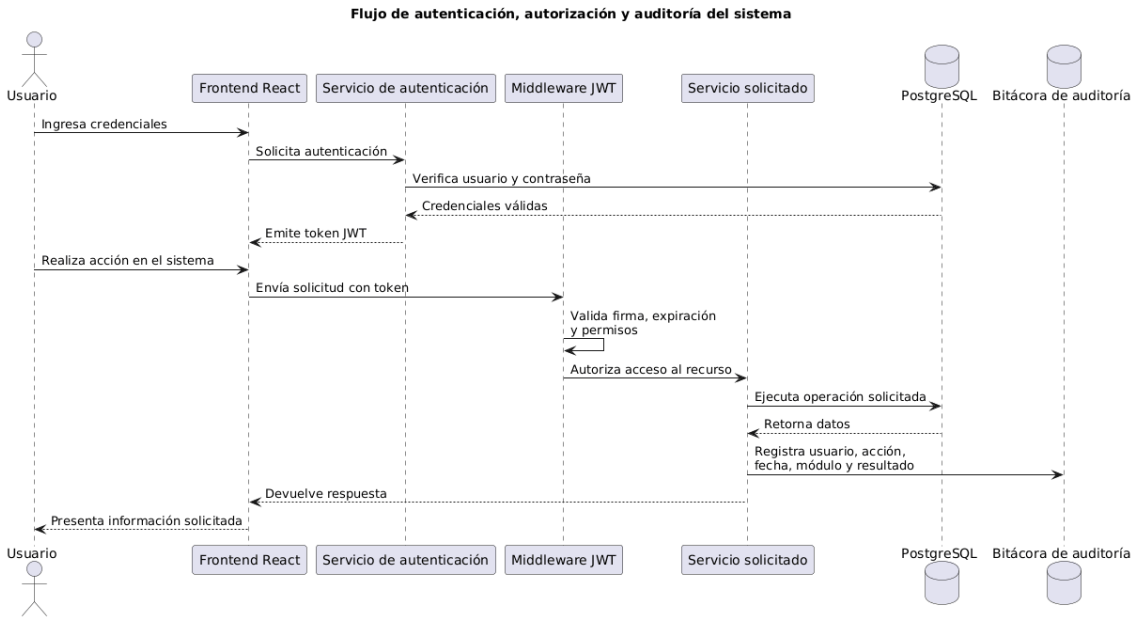
estuvieran disponibles para usuarios autenticados y con permisos válidos dentro del sistema.

El desarrollo incluyó la configuración de autenticación mediante *tokens* JWT, la incorporación de autenticación de dos factores mediante códigos OTP temporales, el uso de *hashing* seguro de contraseñas, el control de intentos fallidos de inicio de sesión y el bloqueo temporal de cuentas ante accesos incorrectos reiterados. Estos controles permitieron fortalecer el proceso de validación de identidad antes de habilitar el acceso al *dashboard* académico del estudiante.

Además, se incorporaron mecanismos de protección complementarios, como límites de solicitudes por tipo de usuario o servicio, configuración de *CORS*, uso de variables de entorno para datos sensibles, identificadores UUIDv4 en los modelos principales y validaciones de entrada para consultas, audio y archivos de actividades. Estas medidas contribuyeron a reducir riesgos asociados con accesos no autorizados, abuso de *endpoints* y exposición de información sensible.

Finalmente, se implementó un servicio centralizado de registros de eventos, encargado de almacenar y organizar las acciones relevantes generadas durante el funcionamiento del sistema. Este servicio permitió evidenciar procesos relacionados con la autenticación, cierre de sesión, consultas al chatbot, consultas por voz, acceso a datos académicos, operaciones RAG, errores del sistema y acciones de soporte operativo. La Ilustración 15 presenta el flujo general de seguridad y registros de eventos implementado durante este *Sprint*.

Ilustración 15 Diagrama de Secuencia Sprint 5.



2.3.1.5.1. Seguridad del sistema

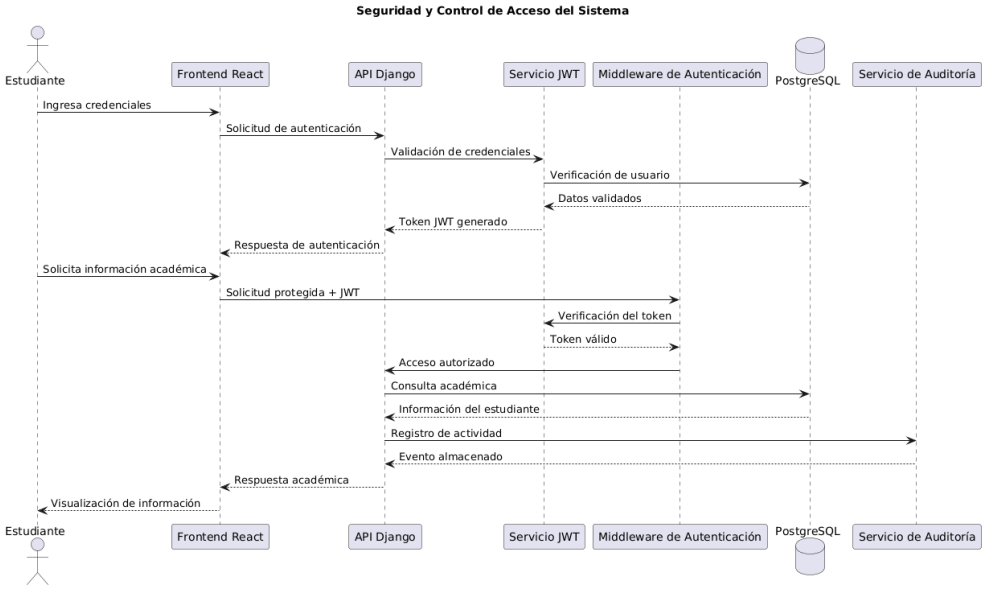
Como se observa en la Ilustración 16, la seguridad del sistema fue organizada por capas con el propósito de proteger tanto la autenticación del usuario como el acceso a los recursos académicos personalizados. La primera capa corresponde a la protección de las solicitudes que llegan al *backend*, donde se aplican configuraciones de *CORS*, cabeceras de seguridad y límites de solicitudes para reducir riesgos asociados con orígenes no autorizados o uso excesivo de los servicios.

La segunda capa se relaciona con la autenticación y gestión de sesiones. En esta etapa se validan las credenciales del usuario, se aplica *hashing* seguro de contraseñas, se genera un código OTP temporal cuando corresponde y se emiten *tokens* JWT para mantener la sesión autenticada. El uso de *access token* y *refresh token* permitió controlar la duración de las sesiones y renovar el acceso sin exponer nuevamente las credenciales del estudiante.

La tercera capa corresponde a la autorización y control de acceso. En esta fase, el *backend* verifica que el usuario autenticado tenga permisos para acceder a los recursos solicitados, evitando que un usuario consulte información académica que no le corresponde. Las rutas protegidas del *frontend* complementan este proceso a nivel de navegación, pero la validación definitiva se realiza en el *backend*.

Finalmente, la capa de registros de eventos permitió almacenar las acciones relevantes generadas durante el funcionamiento del sistema, incluyendo inicio de sesión, intentos fallidos, envío de OTP, cierre de sesión, consultas al chatbot, acceso a datos académicos, consultas RAG, consultas por voz y errores. Estos registros contribuyeron a mejorar la trazabilidad técnica de las acciones realizadas dentro de la plataforma.

Ilustración 16 Seguridad del Sistema.



Como se aprecia en la ilustración, el acceso a la información académica personalizada requiere un proceso de validación compuesto por múltiples etapas de seguridad. Inicialmente, el usuario debe autenticarse mediante sus credenciales, tras lo cual el sistema verifica la identidad utilizando mecanismos complementarios de validación antes de conceder acceso a los recursos protegidos.

La plataforma incorporó medidas de protección como autenticación basada en *tokens* JWT, validación mediante códigos de verificación, control de acceso basado en roles, gestión segura de sesiones y registros de eventos del sistema. La integración de estos mecanismos permitió fortalecer la confidencialidad, integridad y disponibilidad de la información académica gestionada por el sistema, además de mejorar la trazabilidad técnica de las acciones realizadas por los usuarios.

2.3.1.5.1.1. Control de Acceso

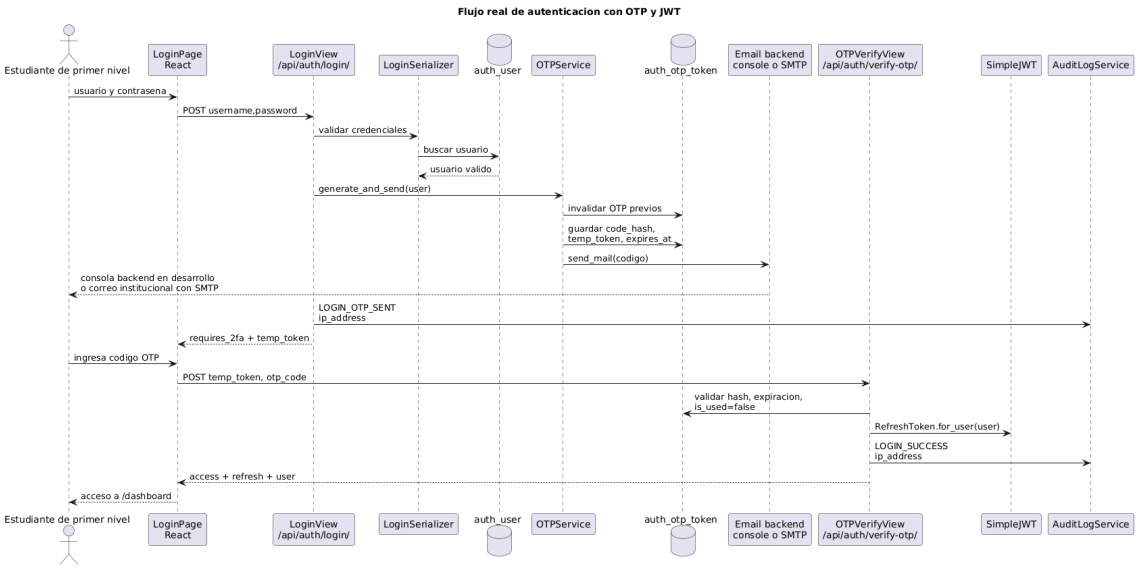
El control de acceso fue implementado con el objetivo de garantizar que únicamente los usuarios autorizados puedan acceder a la información y funcionalidades protegidas del sistema. Como se observa en la Ilustración 17, el proceso inicia cuando el

estudiante ingresa sus credenciales en la interfaz de autenticación. Estas credenciales son enviadas al *backend*, donde se valida la identidad del usuario y se verifica si la cuenta se encuentra habilitada o bloqueada temporalmente por intentos fallidos.

Cuando las credenciales son correctas y el segundo factor está activo, el sistema genera un código OTP temporal de seis dígitos. Este código es enviado al usuario mediante el mecanismo de correo configurado para el entorno de ejecución. En ambiente de desarrollo, el código puede visualizarse en la consola del *backend* por el uso del *backend* de correo de Django; en un entorno con SMTP configurado, el código se envía al correo institucional del usuario.

El código OTP no se almacena como texto plano para su reutilización directa, sino que se gestiona como un valor temporal de un solo uso. Una vez validado el código, el sistema emite los *tokens* JWT correspondientes y permite que el estudiante acceda al *dashboard* académico. En caso de credenciales incorrectas, OTP inválido o intentos reiterados, el sistema registra el evento y aplica las restricciones de seguridad correspondientes.

Ilustración 17 Flujo de autenticación y control de acceso implementado en el sistema.



Este flujo permitió proteger el acceso a la información académica personalizada, ya que las consultas privadas del estudiante solo se habilitan después de superar el proceso de autenticación y verificación. De esta manera, el modo público del chatbot se mantiene separado del modo autenticado, evitando la exposición de datos académicos individuales.

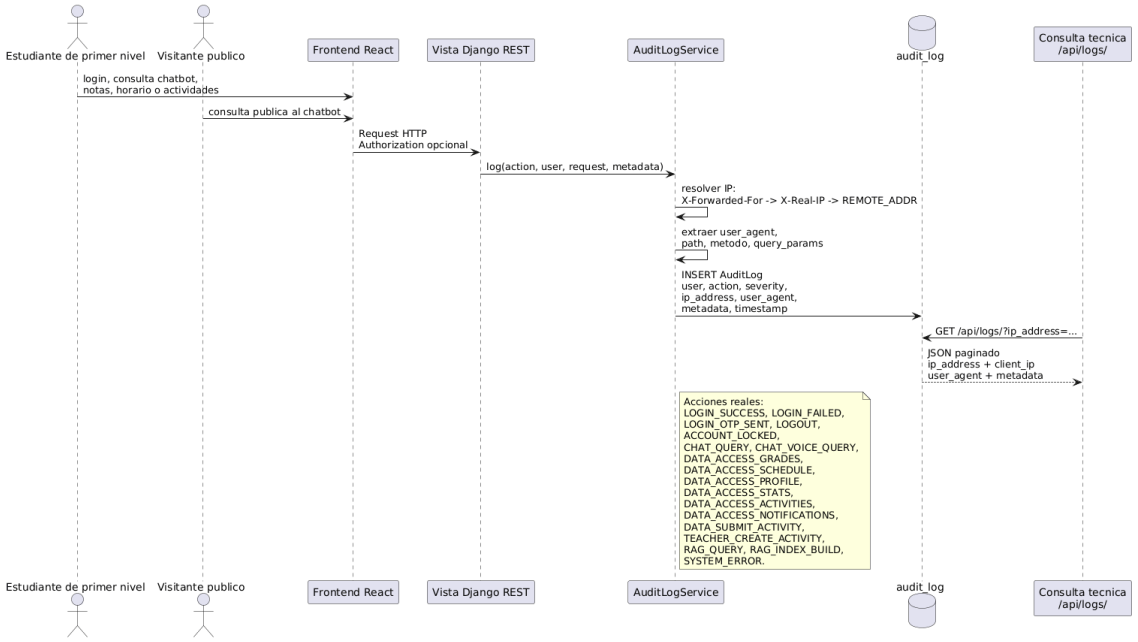
2.3.1.5.1.2. Registros de eventos

El sistema incorporó un servicio centralizado de registros de eventos, encargado de almacenar las acciones relevantes generadas durante el uso del prototipo. Como se observa en la Ilustración 18, cada evento registrado conserva información asociada al usuario, acción realizada, fecha y hora, nivel de severidad, dirección IP, agente de usuario y metadatos complementarios en formato estructurado. Esta información permitió mantener trazabilidad técnica sobre las operaciones ejecutadas dentro de la plataforma.

El modelo de registros de eventos contempló acciones relacionadas con autenticación, consultas al chatbot, interacción por voz, acceso a datos académicos, operaciones RAG, errores del sistema y acciones de soporte operativo. En total, el sistema implementó 18 acciones registrables, entre ellas: inicio de sesión exitoso, intento fallido de autenticación, envío de OTP, cierre de sesión, bloqueo de cuenta, consulta textual, consulta por voz, acceso a calificaciones, consulta de horario, consulta de perfil, visualización de estadísticas, revisión de actividades, consulta de notificaciones, entrega de actividades, consulta RAG, reconstrucción del índice RAG, registro de errores del sistema y acciones de soporte operativo.

La acción relacionada con la creación de actividades por parte del docente se mantuvo como un evento de soporte operativo, debido a que permite alimentar el flujo académico del estudiante, pero no forma parte del alcance funcional principal evaluado en la tesis. De esta manera, los registros de eventos fueron utilizados como un mecanismo técnico de trazabilidad, control y monitoreo, sin ampliar el objetivo central del proyecto hacia la administración docente.

Ilustración 18 Servicio centralizado de registros de eventos del sistema



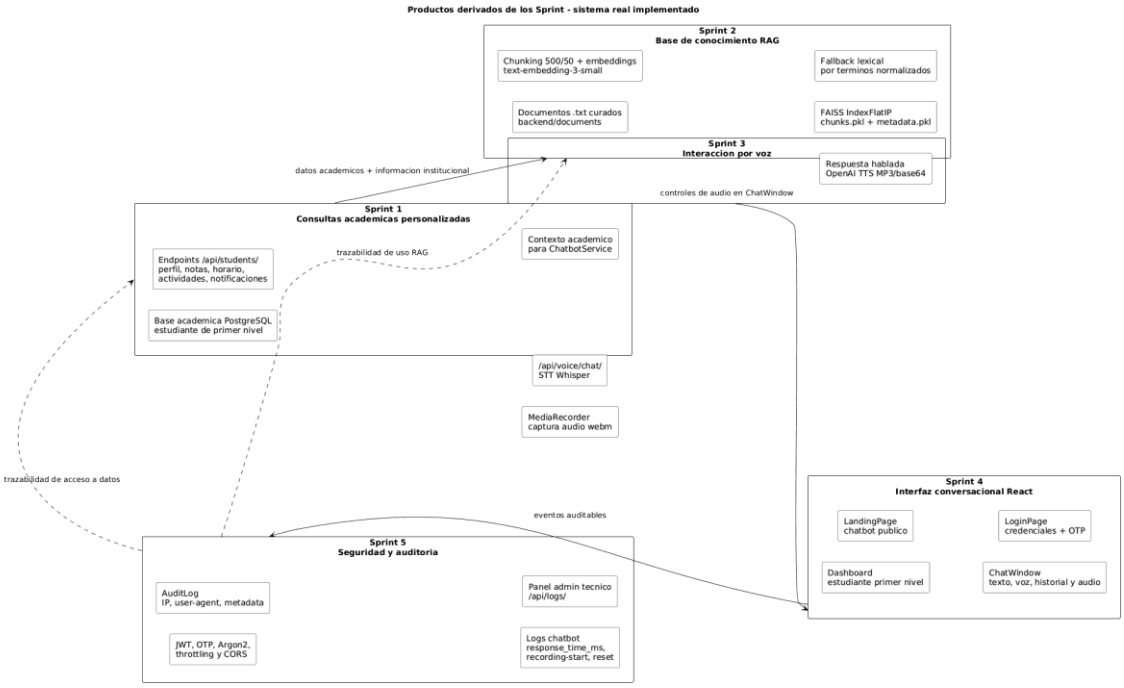
El mecanismo de registros de eventos implementado permitió almacenar de forma centralizada las acciones generadas dentro de la plataforma, incluyendo procesos de autenticación, consultas académicas, accesos a recursos protegidos y eventos de seguridad. La información registrada proporcionó trazabilidad técnica sobre las actividades realizadas por los usuarios y facilitó las tareas de monitoreo, control y revisión ante posibles incidencias dentro del sistema.

2.3.2. Productos derivados de los *Sprint* de desarrollo

Como resultado de la ejecución de los *Sprints* planificados mediante la metodología *Scrum*, se obtuvieron diversos productos tecnológicos que conforman la solución propuesta. Como se observa en la Ilustración 19, los productos generados abarcan los principales componentes funcionales y tecnológicos desarrollados durante las diferentes iteraciones del proyecto.

Entre los productos obtenidos se encuentran la arquitectura de software del chatbot académico, la base de datos académica en PostgreSQL, la base de conocimiento institucional implementada mediante arquitectura RAG, los servicios *backend* desarrollados en Django REST *Framework*, la interfaz web construida en React, el *dashboard* académico del estudiante, el widget conversacional público y autenticado, los servicios de interacción por voz, y los mecanismos de seguridad y registros de eventos incorporados para el control, monitoreo y seguimiento de las operaciones realizadas dentro del sistema.

Ilustración 19 Productos obtenidos durante la ejecución de los Sprint de desarrollo.



2.3.2.1. Diseño de Arquitectura General

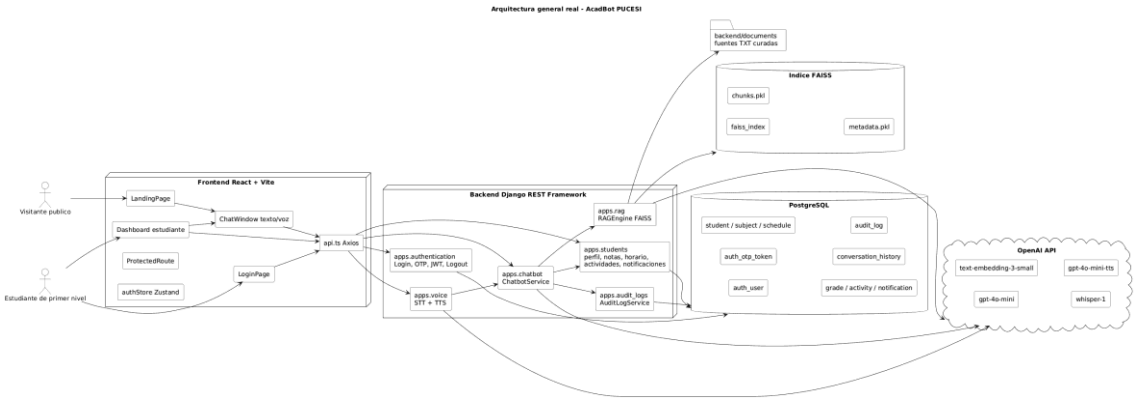
La solución propuesta fue desarrollada bajo una arquitectura cliente-servidor multicapa, orientada a separar la interfaz de usuario, la lógica de negocio, la gestión de datos académicos, la recuperación documental y los servicios de inteligencia artificial. Como se muestra en la Ilustración 20, el *frontend* desarrollado en React actúa como punto de acceso para el visitante público y para el estudiante autenticado, mientras que el *backend* implementado en Django REST Framework centraliza la autenticación, el procesamiento de consultas, la recuperación de información, el acceso a datos académicos y el servicio de registros de eventos del sistema.

La arquitectura integra una base de datos PostgreSQL para almacenar información académica simulada del estudiante de primer nivel, una base de conocimiento institucional procesada mediante arquitectura *Retrieval-Augmented Generation* (RAG), un índice vectorial FAISS y servicios de OpenAI para generación de respuestas, *embeddings*, transcripción de voz y síntesis de audio. Todas las operaciones se realizan a través de servicios REST, por lo que el *frontend* no accede directamente a la base de datos ni al índice vectorial.

Esta organización permitió diferenciar dos modos de uso dentro del sistema: el modo público, destinado a consultas institucionales generales sin autenticación, y el modo personalizado, disponible para estudiantes autenticados con acceso a horarios,

asignaturas, calificaciones, actividades, entregas y notificaciones. Los componentes docentes y administrativos se mantuvieron como soporte técnico para alimentar datos, gestionar actividades y revisar trazabilidad, sin formar parte del flujo funcional principal evaluado en la tesis.

Ilustración 20 Arquitectura General del Sistema.



La arquitectura presentada integra los principales componentes que intervienen durante el procesamiento de las consultas realizadas por los usuarios. El flujo inicia cuando una solicitud es enviada desde la interfaz desarrollada en React hacia los servicios *backend* implementados en Django REST Framework, los cuales gestionan la lógica de negocio y determinan el origen de la información requerida.

Cuando la consulta corresponde a información académica personalizada, el sistema recupera los datos desde la base de datos PostgreSQL, donde se almacena información relacionada con estudiantes, asignaturas, horarios, actividades, calificaciones y notificaciones académicas. Por otra parte, las consultas de carácter institucional son procesadas mediante la arquitectura *Retrieval-Augmented Generation* (RAG), que recupera información relevante desde los documentos previamente indexados en FAISS utilizando técnicas de búsqueda semántica basadas en *embeddings*.

Posteriormente, el contexto recuperado es incorporado en un *prompt* enriquecido que es enviado al modelo GPT-4o-mini para la generación de la respuesta. Finalmente, la información generada es devuelta a la interfaz del usuario, permitiendo presentar respuestas contextualizadas mediante texto o voz según el mecanismo de interacción utilizado.

La integración de estos componentes permitió combinar información académica estructurada, conocimiento institucional y capacidades avanzadas de inteligencia

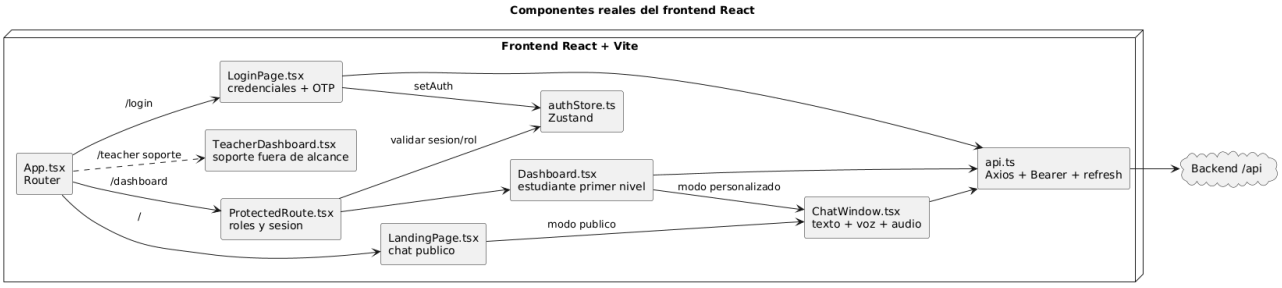
artificial dentro de una arquitectura unificada, favoreciendo la generación de respuestas precisas y fundamentadas en información verificable.

2.3.2.1.1. Diseño de la arquitectura del *Frontend*

La interfaz *frontend* constituye la capa de presentación del sistema y fue desarrollada utilizando React. Este componente es responsable de la interacción directa con los usuarios, permitiendo acceder a los diferentes servicios académicos disponibles dentro de la plataforma.

Como se observa en la Ilustración 21, la arquitectura *frontend* integra los principales componentes visuales utilizados por los usuarios no autenticados, incluyendo la interfaz principal de navegación, los módulos informativos institucionales y el chatbot público. Estos elementos permiten acceder a información institucional y realizar consultas mediante lenguaje natural desde cualquier sección de la plataforma.

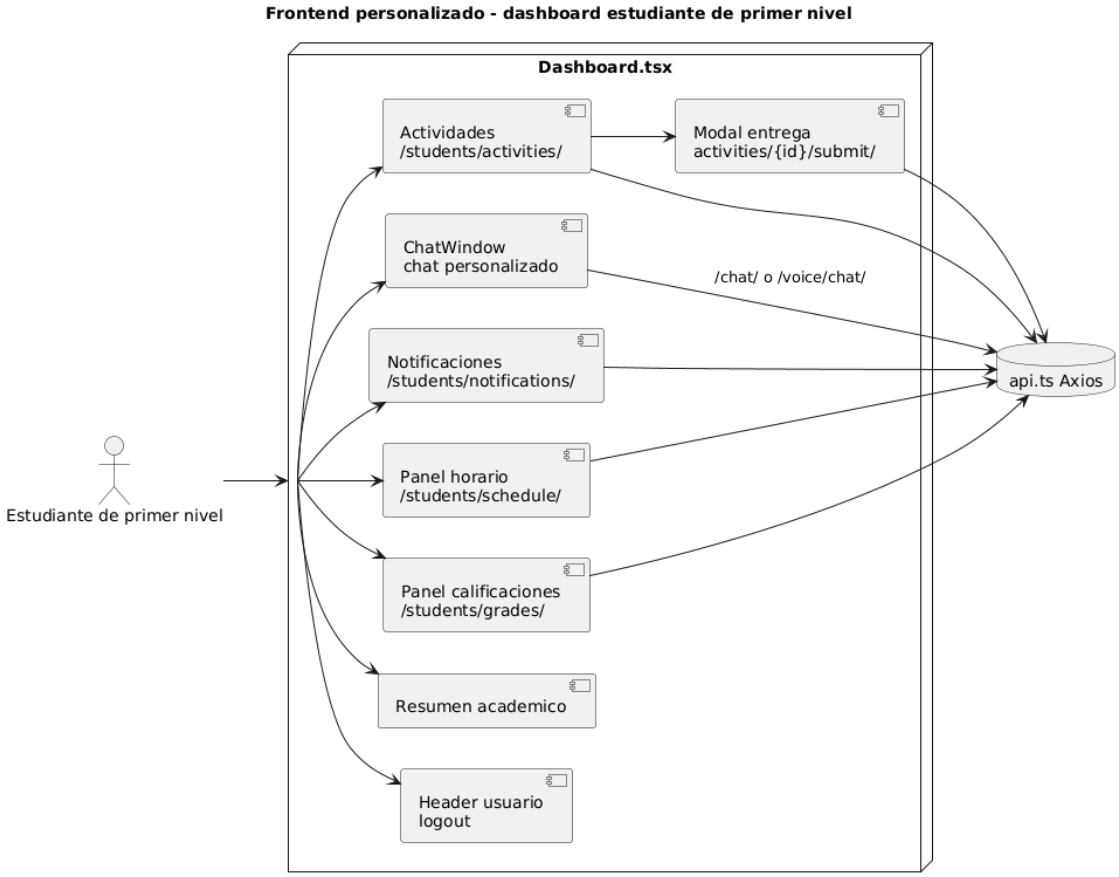
Ilustración 21 Componentes de la capa frontend desarrollada en React.



Uno de los componentes principales implementados es el chatbot flotante inteligente, diseñado para mantenerse accesible durante toda la navegación. Este componente permite realizar consultas institucionales mediante texto o voz sin necesidad de abandonar la página en la que se encuentra el usuario.

Por otra parte, la Ilustración 22 muestra la arquitectura *frontend* correspondiente al entorno autenticado del estudiante de primer nivel. En esta vista se incorporan componentes destinados a la gestión de información académica personalizada, incluyendo calificaciones, horario académico, notificaciones, actividades disponibles, entregas de tareas y acceso al chatbot autenticado con contexto del estudiante.

Ilustración 22 Componentes de la capa frontend personalizada desarrollada en React.



El componente conversacional funciona como un widget flotante reutilizable en dos escenarios: la página pública y el *dashboard* estudiantil. En la página pública, el chatbot opera sin sesión autenticada y responde únicamente consultas institucionales generales; en el *dashboard*, el mismo componente se ejecuta con sesión válida y puede utilizar información académica personalizada del estudiante. Esta separación permitió mantener diferenciados los datos públicos y privados dentro de la interfaz.

El chatbot funciona mediante dos estados principales: cerrado y abierto. En estado cerrado se visualiza un botón flotante ubicado en la esquina inferior derecha de la pantalla; al activarlo, se despliega una ventana conversacional desde la cual el usuario puede interactuar mediante texto o voz. La interfaz incorpora historial local de conversación, captura de audio mediante micrófono, envío de audio al *backend*, reproducción de respuestas habladas y cierre de sesión desde el entorno autenticado.

Las rutas relacionadas con docente y administración técnica se mantuvieron únicamente como soporte operativo del prototipo, principalmente para la carga de actividades, administración de datos y revisión de trazabilidad. Por esta razón, la descripción funcional

del *frontend* se centró en las rutas principales del alcance de tesis: página pública, autenticación y *dashboard* del estudiante.

2.3.2.1.2. Diseño de la arquitectura del *Backend*

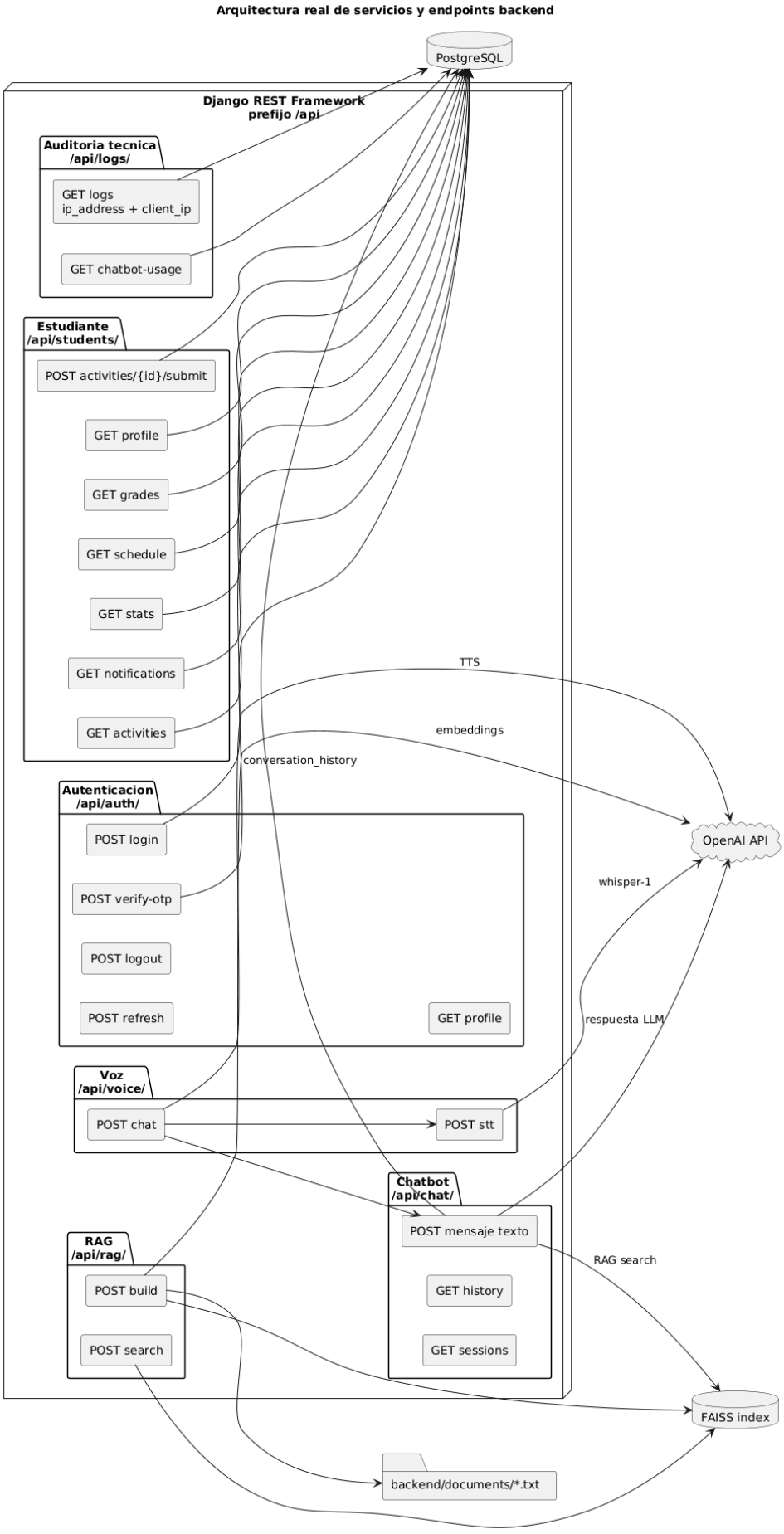
La capa *backend* fue desarrollada utilizando Django y Django REST Framework, proporcionando una arquitectura basada en servicios REST para la comunicación entre la interfaz *frontend* y los componentes internos del sistema. Esta capa centralizó la autenticación, el acceso a información académica, el procesamiento conversacional, la recuperación documental mediante RAG, la interacción por voz y el servicio de registros de eventos del sistema.

La solución se organizó mediante aplicaciones internas especializadas. El módulo de autenticación gestionó el inicio de sesión, la verificación OTP, la emisión y renovación de *tokens* JWT, el cierre de sesión y el perfil autenticado. El módulo de estudiantes administró la información académica del estudiante de primer nivel, incluyendo perfil, calificaciones, horarios, actividades, entregas y notificaciones. El módulo de chatbot procesó las consultas textuales, incorporó contexto académico cuando existió sesión autenticada y coordinó la recuperación institucional mediante RAG.

De manera complementaria, el módulo RAG se encargó de la construcción y consulta del índice FAISS, la fragmentación documental, la generación de *embeddings* y el respaldo léxico sobre documentos institucionales. El módulo de voz gestionó la transcripción de audio, el envío de consultas habladas al motor conversacional y la síntesis de respuestas en audio. Finalmente, el módulo de registros de eventos almacenó acciones relevantes relacionadas con autenticación, consultas al chatbot, acceso a datos académicos, interacción por voz, operaciones RAG y errores del sistema.

Los *endpoints* de docente, administración técnica, reconstrucción del índice RAG y consulta de logs se consideraron servicios de soporte del prototipo. Estos permitieron alimentar datos, crear actividades y revisar trazabilidad, pero no constituyeron el flujo funcional principal evaluado en la tesis, el cual se centró en el visitante público y en el estudiante de primer nivel autenticado.

Ilustración 23 Arquitectura de servicios y endpoints del backend.


Arquitectura real de servicios y endpoints backend
PostgreSQL
Django REST Framework
prefijo /api
Auditoria tecnica
/api/logs/
GET logs
ip_address + client_ip
GET chatbot-usage
Estudiante
/api/students/
POST activities/{id}/submit
GET profile
GET grades
GET schedule
GET stats
GET notifications
GET activities
TTS
embeddings
conversation_history
Autenticacion
/api/auth/
POST login
POST verify-otp
POST logout
POST refresh
GET profile
OpenAI API
whisper-1
respuesta LLM
Voz
/api/voice/
POST chat
POST stt
Chatbot
/api/chat/
POST mensaje texto
GET history
GET sessions
RAG search
RAG
/api/rag/
POST build
POST search
FAISS index
backend/documents/*.txt

2.3.2.1.3. Entorno Tecnológico del Sistema

El chatbot académico inteligente fue desarrollado sobre una arquitectura cliente-servidor orientada a la construcción de aplicaciones web modernas basadas en inteligencia artificial, recuperación contextual de información y procesamiento de lenguaje natural. La selección de tecnologías se realizó considerando criterios de escalabilidad, compatibilidad, facilidad de integración y soporte para arquitecturas conversacionales basadas en modelos de lenguaje de gran escala (LLMs).

En la capa de presentación se utilizó React 19 como framework principal para el desarrollo de la interfaz de usuario. Esta tecnología permitió construir una aplicación web basada en componentes reutilizables, facilitando la implementación de una experiencia conversacional dinámica y altamente interactiva. Complementariamente, se empleó TailwindCSS para el diseño visual de la plataforma, permitiendo desarrollar una interfaz responsiva adaptable a diferentes resoluciones de pantalla y dispositivos de acceso.

La capa de negocio fue desarrollada mediante Django 4.2 y Django REST Framework, tecnologías que permitieron implementar la lógica de negocio del sistema, la gestión de autenticación de usuarios, la administración de servicios REST y la comunicación entre la interfaz conversacional y los diferentes componentes de inteligencia artificial. La utilización de una arquitectura basada en APIs REST favoreció el desacoplamiento entre *frontend* y *backend*, facilitando la mantenibilidad y escalabilidad de la solución propuesta.

Para la gestión de información académica personalizada se utilizó PostgreSQL 16 como sistema gestor de bases de datos relacional. Esta tecnología permitió almacenar y consultar información estructurada relacionada con estudiantes, asignaturas, horarios, aulas, actividades académicas y registros institucionales. La elección de PostgreSQL se fundamentó en su robustez, rendimiento y amplio soporte para aplicaciones empresariales y académicas.

La funcionalidad de recuperación contextual de información institucional fue implementada mediante una arquitectura *Retrieval-Augmented Generation* (RAG). Para ello, se empleó FAISS (*Facebook AI Similarity Search*) como motor de búsqueda vectorial, encargado de almacenar y recuperar representaciones semánticas de los documentos institucionales previamente indexados. Este mecanismo permitió realizar

búsquedas basadas en similitud semántica, incrementando la precisión de las respuestas generadas por el chatbot.

En el componente de inteligencia artificial se utilizaron los servicios proporcionados por OpenAI mediante acceso programático a través de su API oficial. Para la generación de respuestas se empleó el modelo GPT-4o-mini, seleccionado por su capacidad para comprender lenguaje natural, interpretar contexto conversacional y producir respuestas coherentes en tiempo real. Adicionalmente, para la construcción del índice vectorial se utilizó el modelo *text-embedding-3-small*, responsable de transformar los documentos institucionales en representaciones numéricas que posteriormente fueron utilizadas por FAISS durante los procesos de recuperación semántica.

Con la finalidad de ampliar las modalidades de interacción disponibles para el usuario, se incorporaron mecanismos de reconocimiento de voz mediante servicios de conversión Speech-to-Text, permitiendo transformar consultas habladas en texto procesable por el sistema conversacional. Esta funcionalidad complementó la interacción tradicional basada en texto, mejorando la accesibilidad y experiencia de uso del chatbot.

Durante el proceso de desarrollo se utilizó GitHub como plataforma de control de versiones y gestión colaborativa del código fuente. Esta herramienta permitió mantener trazabilidad sobre los cambios realizados en cada *Sprint*, gestionar versiones estables del proyecto y facilitar el seguimiento de incidencias durante el desarrollo incremental de la solución.

El sistema fue desarrollado y probado en un entorno local utilizando Windows 11 como sistema operativo principal. Para la ejecución del *backend* se implementó un entorno virtual de Python que permitió aislar dependencias y garantizar la reproducibilidad del proyecto. Por su parte, el *frontend* fue ejecutado mediante Node.js y Vite, proporcionando tiempos reducidos de compilación y actualización durante las fases de desarrollo y prueba.

Finalmente, las configuraciones sensibles relacionadas con credenciales de acceso, claves API, parámetros de conexión y variables de entorno fueron gestionadas mediante archivos .env. Esta estrategia permitió separar la configuración del código fuente, fortaleciendo la seguridad del sistema y reduciendo la exposición de información crítica durante el ciclo de desarrollo.

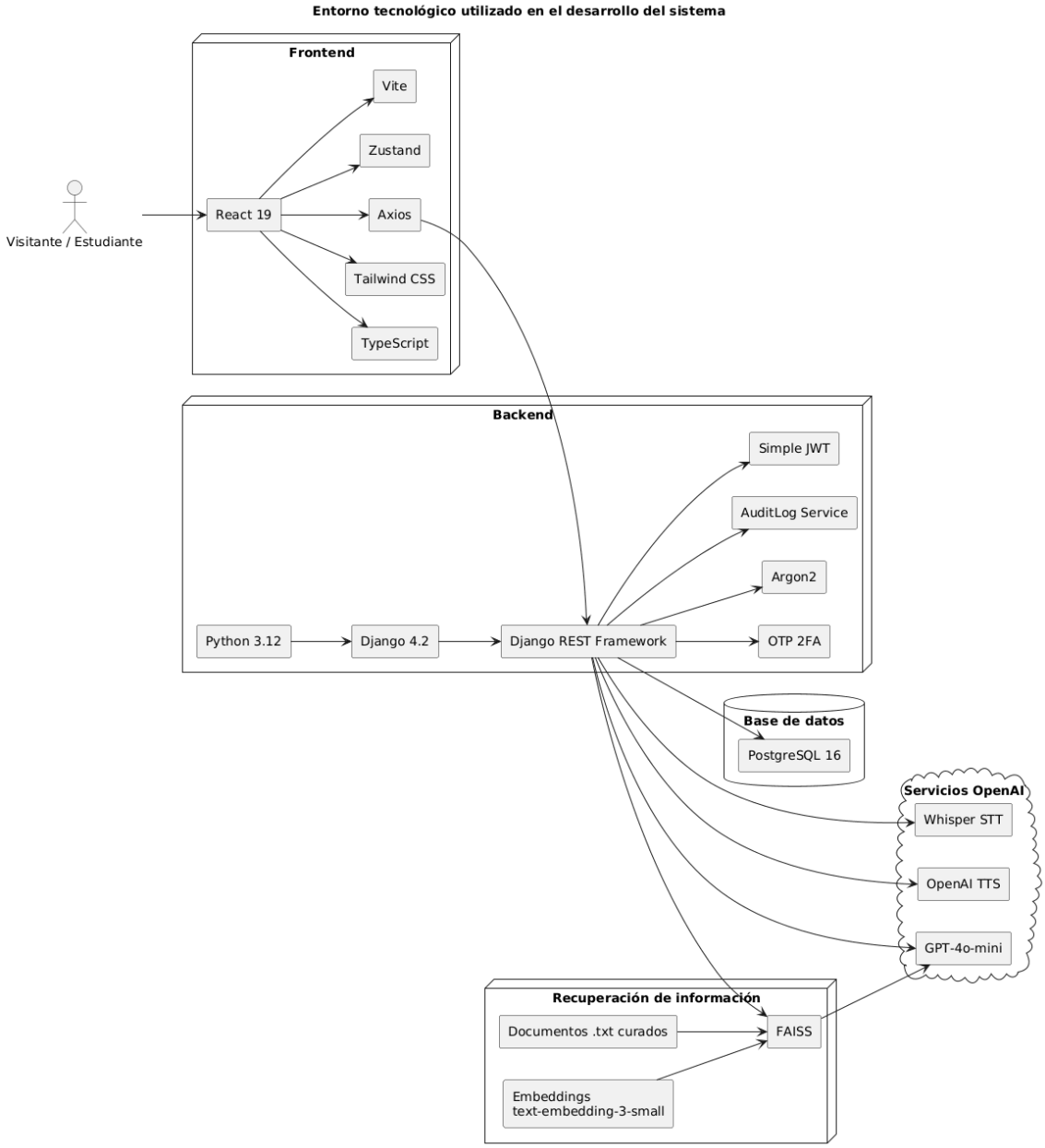
La Tabla 7 presenta las principales tecnologías utilizadas durante el desarrollo del chatbot académico inteligente.

Tabla 7 Tecnologías utilizadas en el desarrollo del sistema.

Componente	**Tecnología Utilizada**
Frontend	React 19 + Vite + TypeScript + TailwindCSS
Cliente HTTP	Axios
Estado de Autenticación	Zustand
Backend	Django 4.2 + Django REST Framework
Base de Datos	PostgreSQL 16
Lenguaje *Backend*	Python 3.12
Autenticación	Simple JWT + OTP
IA Generativa	OpenAI GPT-4o-mini
Embeddings	*text-embedding-3-small*
Motor Vectorial	FAISS
Voz	Whisper STT + OpenAI TTS
API REST	Django REST Framework
Control de Versiones	GitHub
Configuración	Variables de entorno mediante archivo .env
Arquitectura IA	*Retrieval-Augmented Generation* (RAG)

Con la finalidad de visualizar la integración de las principales tecnologías empleadas durante el desarrollo del chatbot académico inteligente, la ilustración 24 presenta el entorno tecnológico implementado en la solución propuesta.

Ilustración 24 Entorno tecnológico utilizado en el desarrollo del sistema.



Como se observa en la ilustración 24, la arquitectura tecnológica integra los componentes responsables de la interfaz de usuario, la lógica de negocio, el almacenamiento de información académica y los servicios de inteligencia artificial. Esta organización permitió desacoplar las responsabilidades del sistema y facilitar la comunicación entre el *frontend* desarrollado en React, el *backend* implementado en Django REST Framework, la base de datos PostgreSQL, el motor vectorial FAISS y los modelos de OpenAI utilizados para la generación de respuestas y recuperación contextual de información institucional.

2.3.2.2. Diseño de Base de Datos

La gestión de información académica constituye uno de los componentes fundamentales del chatbot inteligente desarrollado en la presente investigación. Para ello, se diseñó una base de datos relacional utilizando PostgreSQL, orientada al almacenamiento seguro y estructurado de información relacionada con usuarios, estudiantes, docentes, asignaturas, horarios, matrículas, calificaciones y registros de eventos del sistema.

El modelo de datos fue desarrollado siguiendo principios de normalización con el propósito de minimizar redundancias, garantizar la integridad referencial y optimizar el acceso a la información. Adicionalmente, todas las entidades principales utilizan identificadores universales únicos (UUID) como claves primarias, fortaleciendo la seguridad y reduciendo la posibilidad de colisiones entre registros.

La estructura implementada permite gestionar tanto la información académica personalizada utilizada por el chatbot como los mecanismos de autenticación, trazabilidad y control de acceso requeridos por la plataforma. De esta manera, la base de datos actúa como una fuente central de información para los distintos componentes del sistema, facilitando la recuperación de datos académicos y el registro de eventos de seguridad.

2.3.2.2.1. Modelo Entidad–Relación

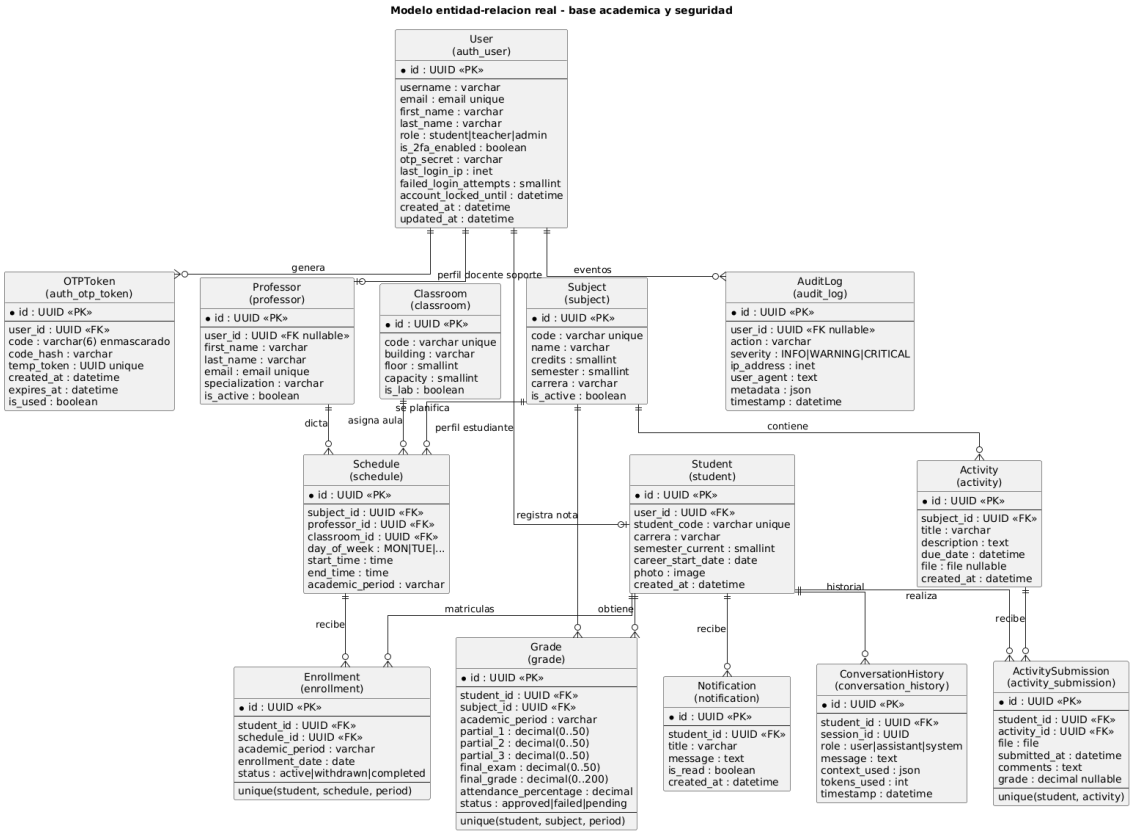
El modelo entidad-relación representa la estructura lógica de los datos utilizados por el sistema. Las entidades principales corresponden a usuarios, estudiantes, docentes, asignaturas, aulas, horarios, matrículas, calificaciones y registros de eventos del sistema.

La entidad User constituye la base del sistema de autenticación y administración de usuarios. A partir de ella se establece una relación uno a uno con la entidad Student, encargada de almacenar la información académica específica de cada estudiante.

Por otra parte, la entidad Schedule actúa como núcleo de la planificación académica, relacionando asignaturas, docentes y aulas dentro de un período académico determinado. Las matrículas permiten asociar estudiantes con horarios específicos, mientras que las calificaciones almacenan el rendimiento académico correspondiente a cada asignatura cursada.

Finalmente, la entidad AuditLog registra los eventos relevantes del sistema, permitiendo mantener trazabilidad sobre accesos, consultas realizadas y acciones ejecutadas dentro de la plataforma.

Ilustración 25 Diagrama entidad-relación de la base de datos académica del chatbot.



Como se observa en la ilustración 25, el modelo entidad-relación integra las entidades académicas necesarias para gestionar estudiantes, docentes, asignaturas, aulas, horarios, matrículas, calificaciones, actividades, entregas, notificaciones e historial conversacional. Esta estructura permite que el chatbot académico recupere información personalizada del estudiante autenticado, mantenga trazabilidad de las interacciones y relacione cada consulta con datos académicos almacenados de forma estructurada en PostgreSQL.

2.3.2.2.2. Descripción de las Entidades Principales

La base de datos se encuentra conformada por diversas entidades que permiten gestionar tanto la información académica como los mecanismos de seguridad implementados en el sistema.

La entidad User almacena la información de autenticación y control de acceso de los usuarios registrados. Student complementa esta información con datos académicos específicos de cada estudiante, incluyendo carrera, semestre y código institucional.

Las entidades *Subject*, *Professor* y *Classroom* representan respectivamente las asignaturas, docentes y aulas utilizadas para la planificación académica. Estas entidades se integran mediante la entidad Schedule, la cual define los horarios de clase para cada período académico.

La entidad *Enrollment* registra las matrículas realizadas por los estudiantes, estableciendo la relación entre cada estudiante y los horarios asignados. Por otra parte, Grade almacena las calificaciones obtenidas por los estudiantes en cada asignatura, incluyendo notas parciales, examen final y porcentaje de asistencia.

La entidad de historial conversacional permite conservar el contexto de interacción de los estudiantes autenticados, facilitando la continuidad de las conversaciones y proporcionando información útil para procesos de análisis, monitoreo y mejora continua del sistema.

Finalmente, la entidad *AuditLog* registra eventos relacionados con la seguridad y el uso del sistema, permitiendo mantener trazabilidad sobre acciones críticas como inicios de sesión, consultas académicas y accesos a información sensible.

2.3.2.3. Diseño de la Interfaz

El diseño de la interfaz fue concebido con el objetivo de proporcionar una experiencia de interacción intuitiva, accesible y eficiente dentro de la plataforma académica inteligente. Para su construcción se definió una guía visual basada en componentes reutilizables, criterios de usabilidad, jerarquía de contenidos, organización por secciones y consistencia gráfica entre las diferentes vistas del sistema.

La propuesta visual consideró dos interfaces principales. La primera corresponde a la interfaz general o pública, destinada a usuarios visitantes, donde se integran elementos institucionales, accesos informativos, menú de navegación, secciones promocionales y el chatbot público. Esta vista fue diseñada con una estructura amplia, visual y de fácil exploración, priorizando el acceso rápido a servicios generales y consultas institucionales.

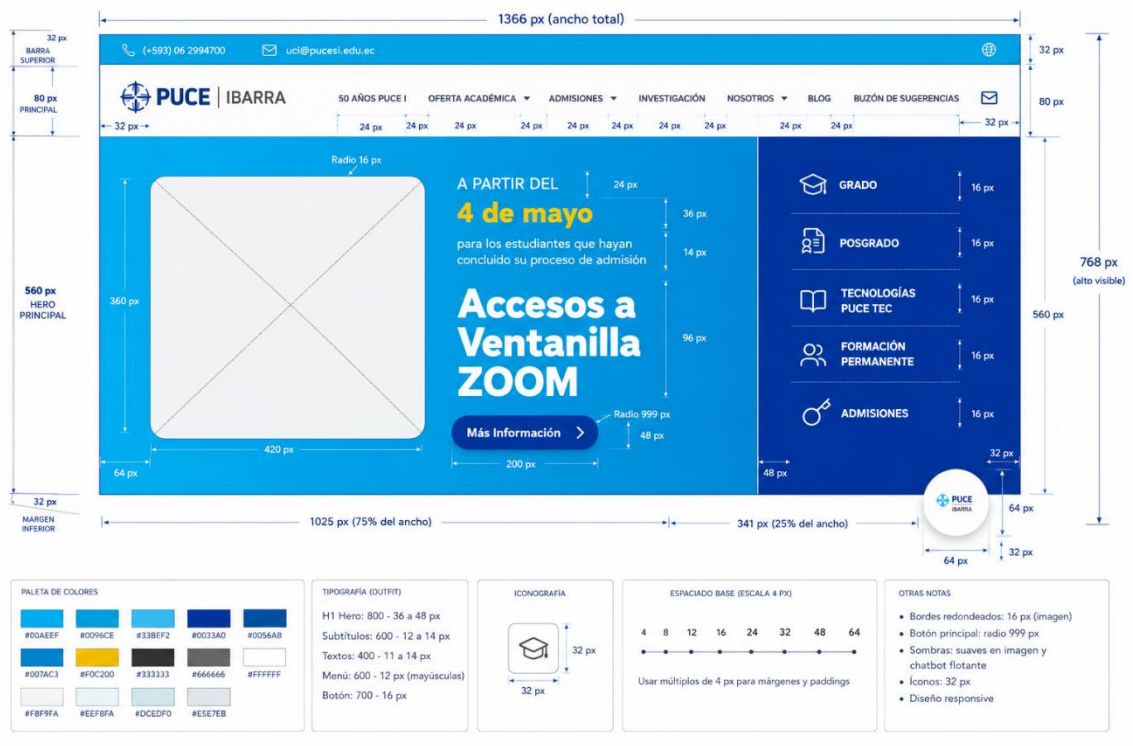
La segunda corresponde a la interfaz personalizada del estudiante, orientada a la consulta de información académica individual. Esta vista fue organizada con una distribución

similar al campus virtual, incorporando un encabezado superior, menú lateral, área central de cursos y una sección lateral de eventos. Su diseño permite acceder de manera ordenada a horarios, asignaturas, actividades, notificaciones, archivos y funcionalidades del chatbot académico.

Para mantener coherencia visual entre ambas interfaces, se definieron lineamientos de diseño relacionados con tipografía, paleta cromática, espaciado, bordes, botones, tarjetas, íconos y comportamiento responsive. Estos criterios sirvieron como base para elaborar los bocetos de interfaz y facilitar su posterior implementación en el *frontend* mediante componentes reutilizables.

Como se muestra en la Ilustración 26, la interfaz general organiza la navegación pública mediante una barra superior de contacto, un menú principal, una sección destacada tipo *hero*, accesos rápidos y el chatbot flotante. Esta estructura permite que el usuario visitante acceda a información institucional y realice consultas generales desde un entorno visual claro y ordenado.

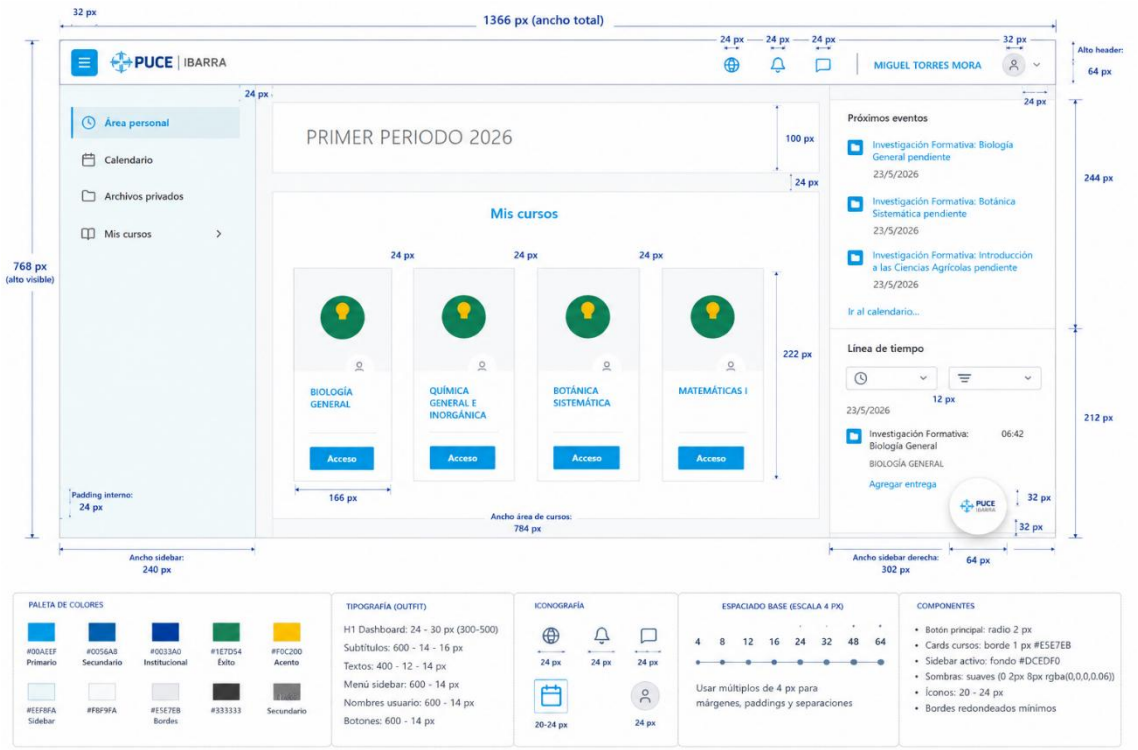
Ilustración 26 Interfaz general de la plataforma académica inteligente.



Por su parte, la Ilustración 27 presenta la interfaz personalizada del estudiante, donde se distribuyen los módulos académicos principales en un entorno de trabajo protegido. En esta vista se visualizan el menú lateral, el área de cursos, los próximos eventos y los

accesos a funcionalidades académicas, manteniendo disponible el chatbot como mecanismo de apoyo conversacional durante la navegación.

Ilustración 27 Interfaz personalizada del estudiante.



2.3.2.3.1. Diseño de navegación

El diseño de navegación fue desarrollado con el objetivo de facilitar el acceso a las funcionalidades principales de la plataforma académica inteligente, permitiendo que el visitante público y el estudiante de primer nivel localicen la información requerida de manera rápida e intuitiva. La estructura se organizó mediante rutas públicas y rutas protegidas, diferenciando el acceso a consultas institucionales generales y el acceso a información académica personalizada mediante autenticación.

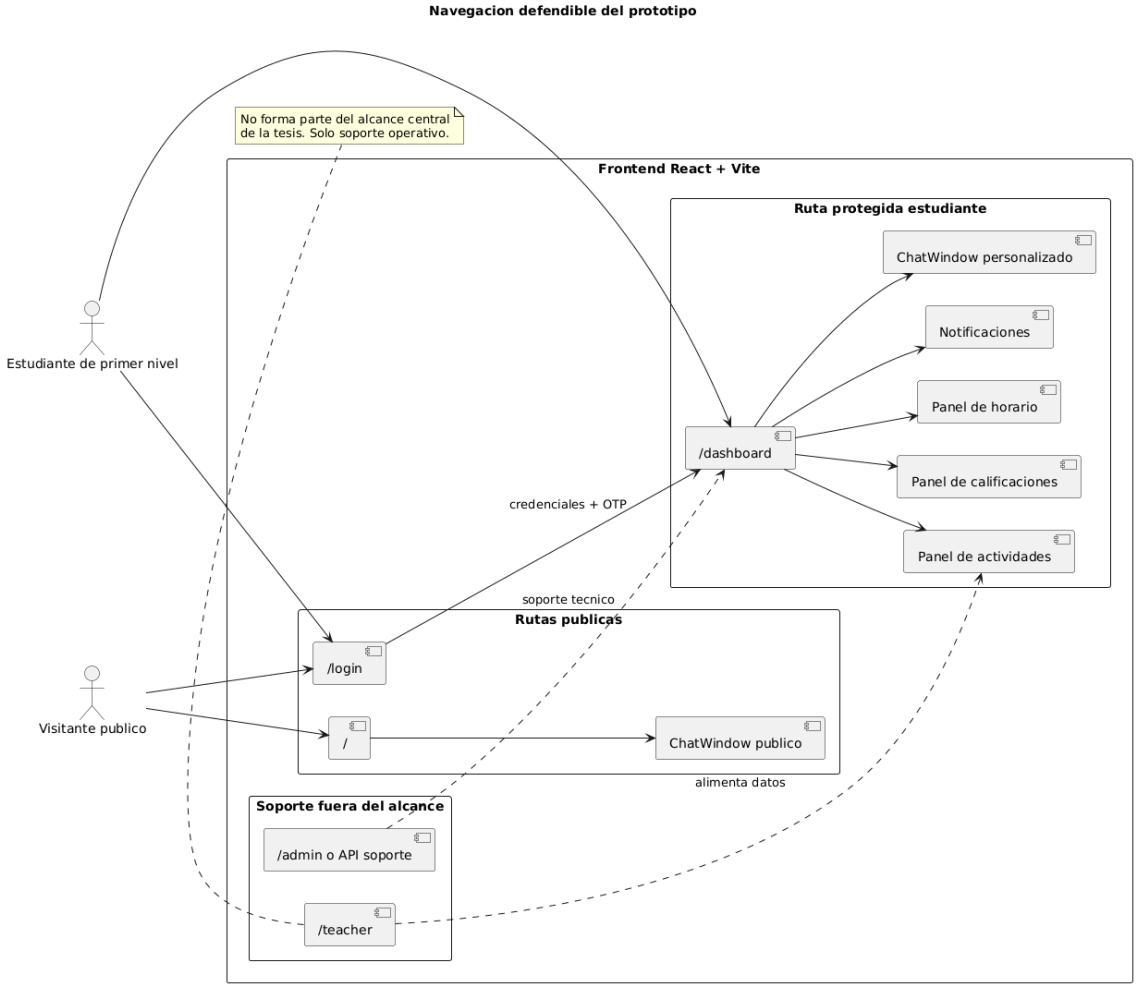
Como se observa en la Ilustración 28, el flujo de navegación inicia en la página pública de la plataforma, desde la cual el visitante puede consultar información institucional mediante el chatbot sin necesidad de iniciar sesión. Cuando la consulta requiere datos académicos personalizados, el sistema dirige al usuario hacia el proceso de autenticación para validar su identidad y habilitar el acceso a los recursos protegidos.

Una vez autenticado, el estudiante de primer nivel accede al *dashboard* académico, donde puede visualizar calificaciones, horario, notificaciones, actividades disponibles y opciones de entrega de tareas. Desde este entorno también puede utilizar el chatbot

autenticado, el cual incorpora contexto académico del estudiante para responder consultas personalizadas de acuerdo con su perfil.

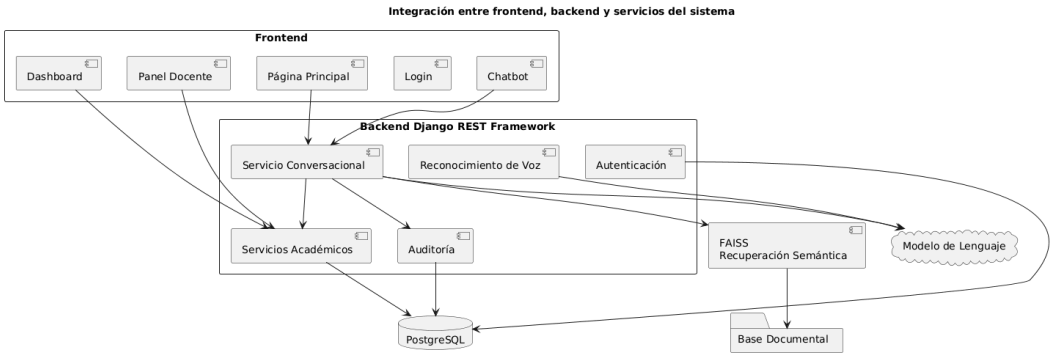
En la misma Ilustración 28 se diferencian las rutas principales del usuario evaluado y las rutas de soporte del prototipo. Las rutas relacionadas con el panel docente o la consulta técnica de registros de eventos no forman parte del recorrido principal de validación; su función dentro del sistema es permitir la carga de actividades, la administración técnica de datos y la revisión de la trazabilidad generada durante el uso del prototipo.

Ilustración 28 Diagrama general de navegación de la plataforma



Por su parte, la Ilustración 29 muestra la integración entre las capas de *frontend* y *backend*, así como los principales servicios que intervienen en el procesamiento de consultas, autenticación, recuperación de información académica y generación de respuestas conversacionales.

Ilustración 29 Integración entre frontend, backend y servicios del sistema



2.3.2.3.2. Interfaz Web General

La interfaz web general constituye el entorno principal de interacción entre los usuarios y la plataforma académica. Desde este espacio es posible acceder a los diferentes módulos del sistema y a las funcionalidades relacionadas con la gestión de información académica.

El diseño fue desarrollado siguiendo principios de usabilidad y consistencia visual, manteniendo una distribución organizada de los elementos de navegación y una estructura orientada a facilitar el acceso a la información disponible dentro de la plataforma.

2.3.2.3.3. Chatbot Flotante Inteligente

El chatbot flotante inteligente constituye el principal mecanismo de interacción conversacional implementado dentro de la plataforma. Su diseño permite mantener acceso permanente al asistente virtual sin afectar la navegación del usuario dentro del sistema.

La interfaz fue desarrollada como un componente flotante anclado a la esquina inferior derecha de la pantalla, permitiendo alternar entre un estado cerrado y un estado abierto mediante una única interacción. El chatbot incorpora capacidades de comunicación mediante texto y voz, así como mecanismos de gestión de sesiones y almacenamiento temporal del historial de conversación.

La implementación busca ofrecer una experiencia similar a las plataformas modernas de mensajería, facilitando la comunicación entre el usuario y el sistema mediante una interfaz intuitiva y de fácil utilización.

2.3.2.3.3.1. Estado Cerrado

Cuando el chatbot se encuentra en estado cerrado, se visualiza únicamente un botón flotante circular ubicado en la esquina inferior derecha de la pantalla, tal como se observa en la Ilustración 30. Este botón permanece disponible durante toda la navegación del usuario y actúa como punto de acceso al asistente virtual.

El componente utiliza elementos visuales institucionales, incluyendo el logotipo de la Pontificia Universidad Católica del Ecuador Sede Ibarra, con el propósito de facilitar su identificación dentro de la interfaz. Asimismo, incorpora efectos visuales de sombra y resaltado que permiten diferenciarlo del resto de elementos presentes en la pantalla.

La selección de un diseño flotante responde a la necesidad de ofrecer acceso inmediato al chatbot sin ocupar espacio permanente dentro de la interfaz principal. De esta manera, el usuario puede acceder al asistente virtual desde cualquier sección de la aplicación sin afectar la visualización del contenido principal.

Ilustración 30 Estado cerrado del chatbot AcadBot PUCESI.

2.3.2.3.3.2. Estado Abierto

Al seleccionar el botón flotante, se despliega el panel principal de conversación del chatbot, como se muestra en la Ilustración 31. Este panel contiene los elementos

necesarios para la interacción entre el usuario y el sistema, incluyendo encabezado institucional, historial de mensajes, controles de entrada y herramientas de interacción por voz.

El encabezado presenta el logotipo institucional, el nombre AcadBot PUCESI y un indicador visual que identifica el contexto de funcionamiento del asistente. Adicionalmente, incorpora un botón de cierre que permite regresar al estado cerrado del chatbot.

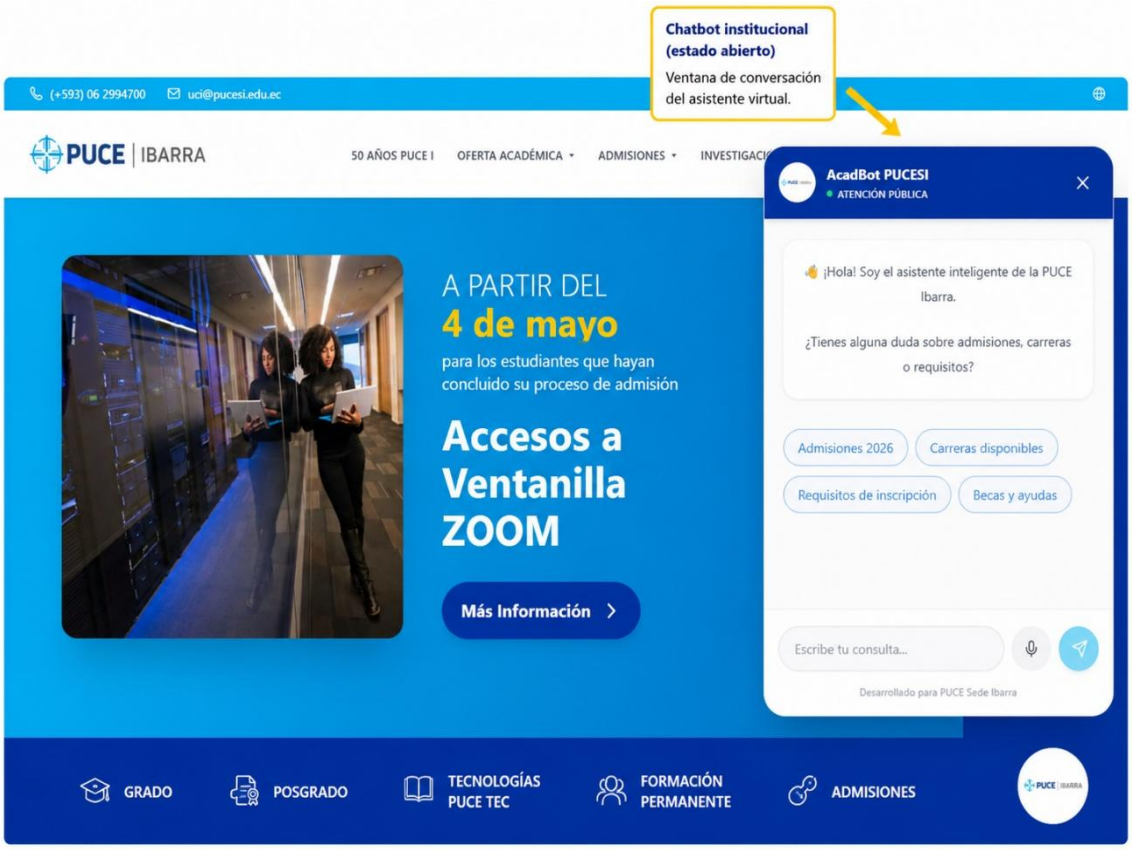
La zona central del panel está destinada a la visualización de mensajes, presentando las consultas realizadas por el estudiante mediante burbujas alineadas a la derecha y las respuestas del asistente mediante burbujas alineadas a la izquierda. Adicionalmente, se incorpora un indicador visual de procesamiento que informa cuando el sistema se encuentra generando una respuesta.

El historial conversacional permite mantener la continuidad de la interacción, conservando el contexto visual de los intercambios previos entre el estudiante y el chatbot. Esta funcionalidad facilita la comprensión del flujo conversacional y contribuye a una experiencia de uso más intuitiva durante la consulta de información académica e institucional.

Cuando el usuario utiliza la modalidad de interacción por voz, el sistema ejecuta un flujo adicional que incluye la captura de audio desde el navegador, la transcripción automática de la consulta, el procesamiento mediante el motor conversacional y la posterior síntesis de voz de la respuesta generada. Este proceso permite establecer una comunicación bidireccional basada completamente en voz, ampliando las posibilidades de interacción disponibles dentro de la plataforma.

Finalmente, la parte inferior del panel contiene los controles de interacción necesarios para el envío de consultas, incluyendo un campo de texto, un botón para reconocimiento de voz y un botón de envío de mensajes. Como complemento, se incorpora una referencia institucional que identifica el sistema como una solución desarrollada para la PUCE Sede Ibarra.

Ilustración 31 Estado abierto del chatbot AcadBot PUCESI.



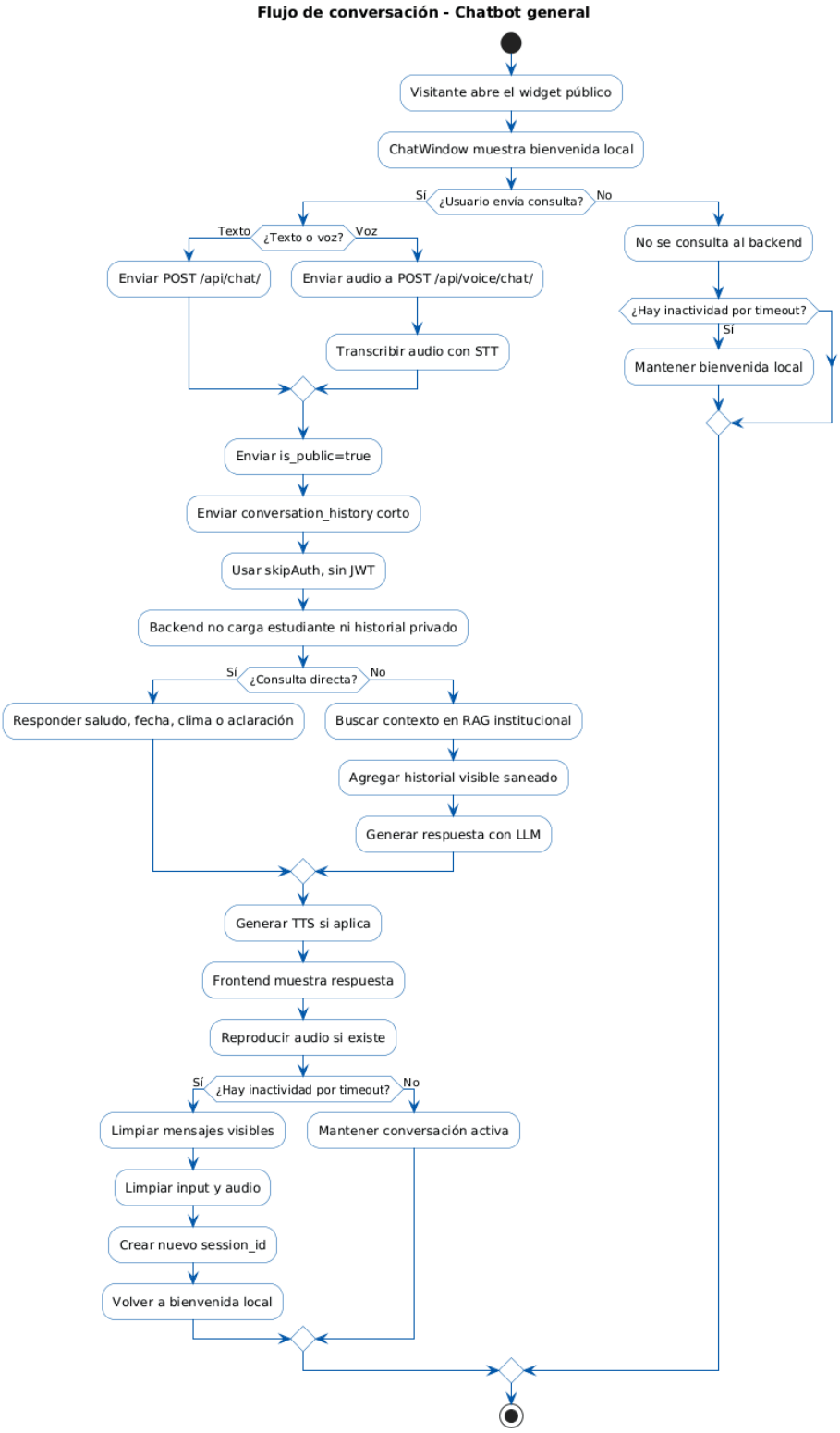
2.3.2.3.3.3. Flujo de Conversación

El flujo de conversación describe la secuencia de interacción entre el usuario y el chatbot académico inteligente dentro del prototipo. Este proceso inicia cuando el usuario formula una consulta mediante texto o voz desde la interfaz conversacional. A partir de esta interacción, el sistema identifica el contexto de uso y diferencia entre dos modalidades principales: el chatbot general, disponible para visitantes o usuarios no autenticados, y el chatbot personalizado, destinado a estudiantes que han iniciado sesión en la plataforma.

En el chatbot general, el flujo inicia cuando el visitante abre el widget público. El componente `*ChatWindow*` muestra un mensaje de bienvenida local y no realiza ninguna petición automática al *backend* hasta que el usuario envía una consulta. Cuando la pregunta se realiza por texto o voz, el *frontend* envía la solicitud con el parámetro `*is_public=true*`, un historial conversacional corto mediante `*conversation_history*` y sin adjuntar *token* JWT. De esta manera, el *backend* procesa la consulta sin cargar perfil académico ni historial privado del usuario. Si la solicitud corresponde a un saludo, fecha, clima o aclaración simple, el sistema genera una respuesta directa; en caso de requerir información institucional, se activa la arquitectura *Retrieval-Augmented Generation*

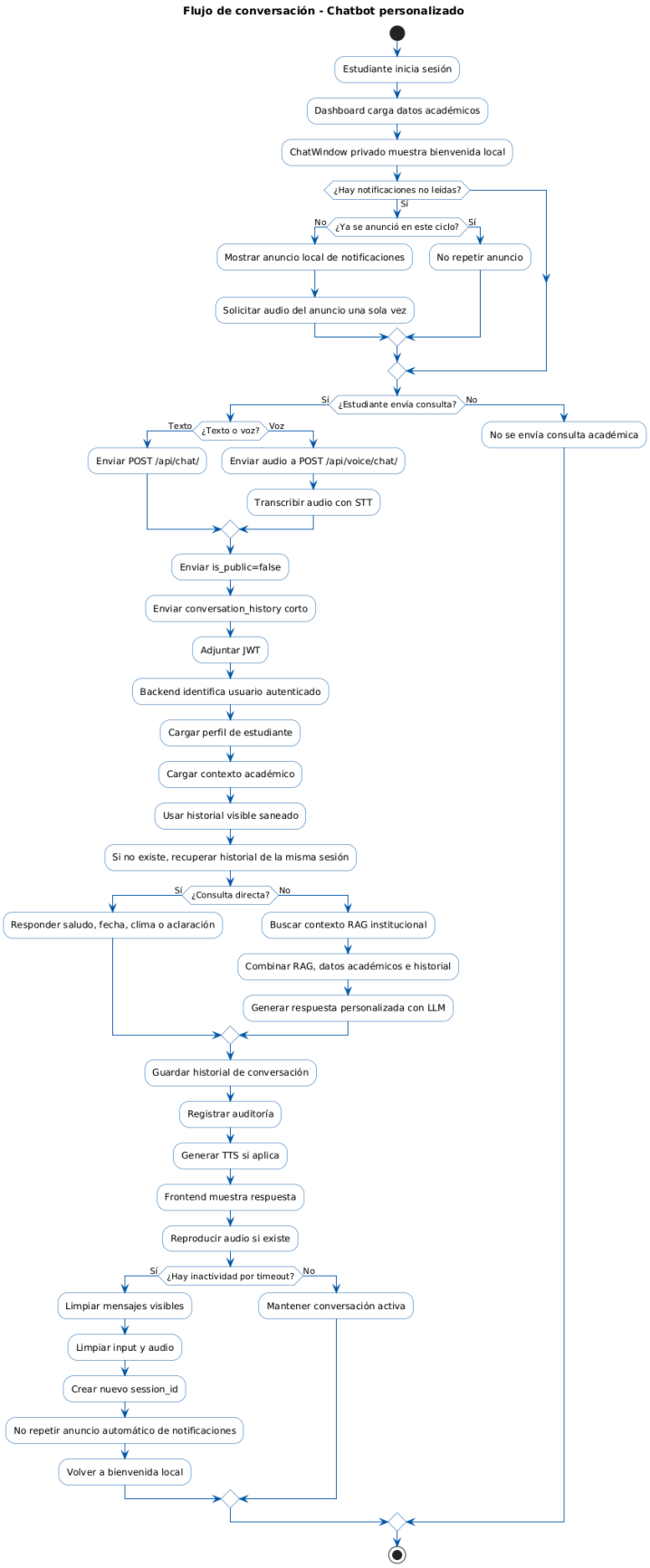
(RAG), se incorpora el historial visible saneado y el modelo de lenguaje genera una respuesta contextualizada. Este comportamiento se representa en la Ilustración 32.

Ilustración 32 Flujo de conversación general.



En el chatbot personalizado, el flujo se ejecuta dentro del *dashboard* del estudiante autenticado. En este caso, el *frontend* envía la consulta con `*is_public=false*`, el *token* JWT y el historial corto de la conversación visible. Posteriormente, el *backend* valida la autenticación del usuario, carga el perfil del estudiante y recupera el contexto académico correspondiente, como nombre, carrera, semestre, horarios, asignaturas, calificaciones, actividades y entregas. Cuando la consulta requiere información personalizada, el sistema utiliza los datos almacenados en PostgreSQL mediante los servicios desarrollados en Django REST Framework. Si la pregunta se relaciona con información institucional o requiere continuidad conversacional, el sistema combina el contexto académico, el historial visible y la recuperación semántica mediante RAG para generar una respuesta personalizada sin inventar datos. Este flujo se muestra en la Ilustración 33.

Ilustración 33 Flujo de conversación personalizado.



Ambos flujos comparten el mismo componente visual `*ChatWindow*`, pero su comportamiento interno cambia según el estado de autenticación del usuario. En el modo público, se prioriza la privacidad y no se almacena historial en la base de datos. En el modo autenticado, la respuesta generada se guarda en el historial de conversación y se registra el evento correspondiente a la consulta realizada. Además, en ambos casos se mantiene un mecanismo de continuidad conversacional mediante el envío de los últimos mensajes visibles, lo que permite interpretar respuestas de seguimiento como "sí", "eso", "el segundo" o "continúa", sin depender exclusivamente del almacenamiento permanente en el servidor.

Finalmente, los dos chatbots incorporan un cierre automático por inactividad configurado por defecto en 30 segundos. Cuando se cumple este tiempo, el *frontend* limpia los mensajes visibles, borra el texto ingresado, detiene cualquier audio en reproducción, genera un nuevo `*session_id*` y vuelve a mostrar el mensaje de bienvenida local. Este mecanismo evita arrastrar contexto conversacional antiguo, mejora la privacidad del usuario y permite que cada nuevo ciclo de interacción inicie de forma ordenada dentro de la plataforma.

2.3.2.3.3.4. Integración con el Entorno Académico

El chatbot fue integrado de manera directa dentro del entorno académico de la plataforma, permitiendo que los estudiantes accedan a sus consultas sin abandonar las diferentes secciones del sistema. Esta integración facilita la interacción con información relacionada con materias, horarios, actividades, notificaciones y servicios institucionales desde una única interfaz conversacional.

La disponibilidad permanente del chatbot dentro de la plataforma contribuye a mejorar la experiencia de usuario, reduciendo la necesidad de navegar entre múltiples módulos para obtener información académica o institucional.

2.3.2.4. Diseño del Plan de Pruebas

Con la finalidad de verificar el comportamiento del chatbot académico inteligente, se diseñó un plan de pruebas orientado a comprobar los módulos principales del sistema y los atributos de calidad definidos durante el desarrollo del proyecto. Esta planificación permitió organizar los casos de prueba antes de la validación final, estableciendo para cada escenario el campo de entrada y el resultado esperado.

El plan se estructuró en dos grupos: pruebas funcionales y pruebas no funcionales. Las pruebas funcionales estuvieron orientadas a comprobar que el sistema ejecutara correctamente sus acciones principales, como responder consultas académicas personalizadas, atender consultas institucionales mediante arquitectura RAG, procesar entradas por texto o voz y restringir el acceso a información privada cuando no existiera una sesión válida. Por otra parte, las pruebas no funcionales se enfocaron en evaluar atributos de calidad del sistema, principalmente usabilidad, claridad de interacción, facilidad de uso, comprensión visual de la interfaz, retroalimentación al usuario, seguridad, autenticación y registros de eventos generados durante el uso del prototipo.

Para mantener una estructura uniforme en todos los casos de prueba, se utilizó la Tabla 8. Plantilla de prueba de aceptación, en la cual se definieron los elementos necesarios para describir cada validación. Esta plantilla permitió registrar el nombre de la prueba, su descripción, los escenarios evaluados, el campo de entrada utilizado, el criterio de aceptación, el resultado esperado y las observaciones correspondientes. Al tratarse de una etapa de diseño del plan de pruebas, no se incluyó el resultado obtenido, debido a que este fue presentado posteriormente en el capítulo de resultados y discusión, una vez ejecutadas las pruebas funcionales y no funcionales.

Tabla 8 Plantilla de prueba de aceptación.

PRUEBA FUNCIONALES/NO FUNCIONALES N°			
Nombre:			
Descripción:			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Esperado***
Observaciones:			

Dentro de las pruebas funcionales, se consideró en primer lugar el chatbot académico personalizado. Este módulo tuvo como finalidad atender consultas asociadas al perfil del estudiante autenticado, tales como horarios, asignaturas matriculadas, aulas, calificaciones, actividades pendientes y promedio académico. Para este componente, se

plantearon escenarios en los que el estudiante ingresaba consultas en lenguaje natural y el sistema debía recuperar información desde la base de datos académica. Este caso se detalla en la Tabla 9. Consultas académicas personalizadas.

Tabla 9 Consultas académicas personalizadas.

PRUEBA DE FUNCIONALIDAD N° 1			
Nombre: Consulta académica personalizada			
Descripción: El chatbot debe permitir consultar información académica personalizada del estudiante autenticado mediante lenguaje natural, incluyendo horarios, asignaturas, aulas, calificaciones, actividades pendientes y promedio académico.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Esperado***
Consulta de horario académico	*¿Cuál es mi horario de hoy?*	*El sistema debe recuperar y mostrar el horario correspondiente al estudiante autenticado.*	
Consulta de calificaciones	*¿Qué calificaciones tengo actualmente?*	*El chatbot debe mostrar las calificaciones registradas para el estudiante autenticado.*	
Consulta de actividades pendientes	*¿Qué actividades tengo pendientes?*	*El sistema debe listar las actividades académicas pendientes*	

		asociadas al estudiante.	
Consulta de asignaturas matriculadas	*¿Qué materias tengo matriculadas?*	*El chatbot debe mostrar las asignaturas registradas en el periodo académico correspondiente.*	
Consulta de aula asignada	*¿En qué aula tengo [MATERIA]?*	*El sistema debe identificar la asignatura consultada y mostrar el aula correspondiente.*	
Consulta de promedio académico	*¿Cuál es mi promedio académico actual?*	*El sistema debe calcular o recuperar el promedio académico del estudiante autenticado.*	
Tiempo de respuesta por texto	*10 consultas académicas personalizadas por texto*	*El tiempo promedio del chatbot personalizado debe ser igual o menor a 3 segundos.*	
Tiempo de respuesta por voz	*10 consultas académicas personalizadas por voz*	*El tiempo promedio del chatbot personalizado*	

		debe ser igual o menor a 3 segundos.	
Observaciones: Esta prueba funcional valida la recuperación de información académica personalizada desde la base de datos, el control de sesión del estudiante y la correspondencia entre la consulta procesada y los datos académicos registrados.			

También se incluyó el chatbot de consultas generales o institucionales, encargado de responder preguntas relacionadas con procesos académicos, servicios universitarios, reglamentos, becas, matrícula y trámites estudiantiles. En este caso, el sistema debía utilizar la arquitectura *Retrieval-Augmented Generation* (RAG) para recuperar información desde la base de conocimiento institucional antes de generar la respuesta. El diseño de esta validación se presenta en la Tabla 10. Consultas institucionales mediante arquitectura RAG.

Tabla 10 Consultas institucionales.

PRUEBA DE FUNCIONALIDAD N° 2			
Nombre: Consultas institucionales mediante RAG			
Descripción: El chatbot debe responder preguntas institucionales utilizando información recuperada desde documentos oficiales indexados.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Esperado***
Consulta sobre biblioteca institucional	*¿Dónde se encuentra la biblioteca?*	*El sistema debe generar una respuesta de orientación institucional relacionada con la biblioteca.*	
Consulta sobre carnet estudiantil	*¿Dónde gestiono mi*	*El chatbot debe orientar al usuario sobre el*	

	carnet estudiantil?	*trámite o dependencia relacionada con la gestión del carnet.*	
Consulta normativa académica	*¿Cuál es la nota mínima para aprobar una materia?*	*El sistema debe recuperar información académica general relacionada con criterios de aprobación.*	
Consulta sobre servicios de cafetería	*¿Dónde se encuentran los servicios de cafetería?*	*El sistema debe entregar información general de orientación sobre los servicios consultados.*	
Consulta sobre pérdida u olvido del carnet	*¿Qué hago si pierdo u olvido mi carnet estudiantil?*	*El chatbot debe orientar al usuario sobre el procedimiento general a seguir.*	
Restricción de información privada	*¿Cuáles son mis calificaciones?*	*El modo público no debe mostrar información personalizada ni acceder a datos académicos privados.*	

Consulta institucional no disponible	*Pregunta sobre información no contenida en la base documental*	*El sistema debe evitar inventar información y orientar al usuario a verificar con la dependencia correspondiente.*	
Tiempo de respuesta por texto	*10 consultas institucionales por texto*	*El tiempo promedio de respuesta del chatbot general debe ser igual o menor a 5 segundos.*	
Tiempo de respuesta por voz	*10 consultas institucionales por voz*	*El tiempo promedio de respuesta del chatbot general debe ser igual o menor a 5 segundos.*	
Observaciones: Esta prueba funcional valida la recuperación semántica mediante FAISS, el uso de *embeddings*, la construcción de contexto y la generación de respuestas fundamentadas mediante el modelo de lenguaje.			

En cuanto a las pruebas no funcionales, se incorporó la evaluación del reconocimiento de voz como parte de la accesibilidad e interacción del sistema. Esta validación permitió establecer escenarios relacionados con consultas habladas, permisos de micrófono, entradas vacías y conversión de voz a texto. El caso correspondiente se presenta en la Tabla 11. Reconocimiento de voz.

Tabla 11 Reconocimiento de voz.

PRUEBA DE USABILIDAD N° 1

Nombre: Interacción mediante reconocimiento de voz			
Descripción: El sistema debe convertir correctamente comandos de voz en texto para su posterior procesamiento conversacional.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Esperado***
Identificación del botón de micrófono	*El usuario observa la interfaz del chatbot*	*El botón de micrófono debe ser visible, reconocible y estar ubicado cerca del campo de entrada*	
Inicio de grabación	*El usuario presiona el botón de micrófono*	*La interfaz debe mostrar una señal clara de que la grabación inició*	
Finalización de grabación	*El usuario suelta o vuelve a seleccionar el botón de micrófono*	*El sistema debe indicar que el audio fue capturado y está siendo procesado*	
Retroalimentación durante procesamiento	*Consulta hablada enviada al sistema*	*La interfaz debe mostrar un mensaje o indicador de carga mientras se*	

		procesa la consulta	
Permiso de micrófono denegado	*El navegador bloquea el acceso al micrófono*	*El sistema debe mostrar un mensaje claro indicando que se requiere permiso de micrófono*	
Audio vacío o no comprensible	*El usuario graba silencio o audio con ruido*	*El sistema debe mostrar un mensaje comprensible solicitando repetir la consulta*	
Respuesta por voz	*El usuario recibe una respuesta generada*	*El sistema debe permitir reproducir la respuesta hablada sin impedir la lectura del texto*	
Observaciones: Esta prueba no funcional evalúa accesibilidad, facilidad de interacción, conversión de voz a texto, manejo de errores de micrófono y continuidad del flujo conversacional.			

Asimismo, se consideró la interfaz conversacional como un componente clave de usabilidad. Esta validación estuvo orientada a comprobar que el chatbot presentara una interacción clara, organizada y comprensible para el usuario, permitiendo enviar consultas, visualizar respuestas, mantener el flujo conversacional y manejar

adecuadamente mensajes extensos. Este caso se describe en la Tabla 12. Interfaz conversacional.

Tabla 12 Interfaz conversacional.

PRUEBA DE USABILIDAD N° 2			
Nombre: Interfaz conversacional inteligente			
Descripción: El *frontend* del chatbot debe gestionar correctamente el flujo de mensajes y respuestas conversacionales del usuario.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Esperado***
Reconocimiento del chatbot	*El usuario ingresa a la página principal*	*El botón flotante debe ser visible y representar claramente el acceso al asistente*	
Apertura del chatbot	*El usuario selecciona el botón flotante*	*La ventana conversacional debe abrirse sin cubrir información esencial de la página*	
Comprensión del campo de texto	*El usuario observa el área de escritura*	*El campo debe presentar una indicación clara sobre qué puede preguntar*	
Indicador de espera	*El chatbot procesa la consulta*	*La interfaz debe mostrar un estado de carga o procesamiento*	
Lectura de respuestas	*El chatbot muestra una respuesta*	*El texto debe presentarse con*	

		tamaño, contraste y organización adecuados	
Continuidad conversacional	*El usuario realiza varias consultas seguidas*	*La interfaz debe conservar los mensajes recientes de forma ordenada*	
Reinicio por inactividad	*El usuario deja de interactuar durante el tiempo configurado*	*El sistema debe limpiar la conversación y mostrar un nuevo mensaje de bienvenida*	
Mensaje de error comprensible	*Ocurre un error de conexión o procesamiento*	*El sistema debe mostrar un mensaje claro, sin lenguaje técnico innecesario*	
Observaciones: Esta prueba no funcional valida criterios de usabilidad, claridad visual, organización del contenido, continuidad conversacional, manejo de errores de entrada y adaptación de la interfaz.			

Finalmente, se diseñaron pruebas no funcionales relacionadas con seguridad, autenticación y registros de eventos. Estas permitieron definir escenarios para verificar el inicio de sesión, el rechazo de credenciales inválidas, la restricción de recursos protegidos, el control de acceso según el usuario y el registro de acciones relevantes dentro del sistema. Este componente se presenta en la Tabla 13. Seguridad, autenticación y registros de eventos.

Tabla 13 Seguridad, autenticación y registros de eventos

PRUEBA DE SEGURIDAD N° 1
Nombre: Seguridad, autenticación y registros de eventos

Descripción: El sistema debe garantizar el acceso seguro mediante autenticación JWT, restringir el acceso a funcionalidades protegidas y registrar los eventos relevantes ejecutados por los usuarios para fines de trazabilidad, control y monitoreo técnico dentro de la plataforma.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Esperado***
Inicio de sesión válido	*Usuario/correo/código estudiantil y contraseña válida.*	*El sistema debe validar las credenciales y continuar con la verificación OTP*	
Verificación mediante OTP	*Código OTP temporal de seis dígitos.*	*El sistema debe validar el código OTP y emitir los tokens JWT de sesión.*	
Rechazo de credenciales inválidas	*Usuario o contraseña incorrectos.*	*El sistema debe rechazar el acceso y no generar sesión.*	
Registro de inicio de sesión	*Login válido completado con OTP.*	*El sistema debe registrar el evento de inicio de sesión.*	
Registro de consulta al chatbot	*Consulta enviada al chatbot.*	*El sistema debe registrar la interacción*	

		como evento trazable dentro de los registros del sistema.	
Registro de consulta por voz	*Audio enviado desde el micrófono del chatbot.*	*El sistema debe registrar la consulta por voz como evento trazable dentro de los registros del sistema.*	
Registro de cierre de sesión	*Acción de cerrar sesión.*	*El sistema debe invalidar la sesión activa y registrar el evento de cierre.*	
Observaciones: Esta prueba no funcional valida autenticación, autorización, protección de recursos, control de acceso por rol, manejo de sesión y trazabilidad de eventos dentro del sistema.			

La organización del plan de pruebas permitió establecer una base clara para la validación posterior del sistema. De esta manera, las Tablas 8 a 13 correspondieron al diseño de los casos de prueba, mientras que los resultados obtenidos durante su ejecución fueron desarrollados en el Capítulo III, dentro del apartado de validación del sistema.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

En el presente capítulo se exponen los resultados obtenidos durante la implementación y validación del sistema propuesto. Se describen las principales interfaces desarrolladas, el funcionamiento de los módulos implementados y las pruebas aplicadas para verificar el cumplimiento de los requerimientos definidos en el capítulo anterior.

3.1. Interfaces del sistema

Como resultado del desarrollo realizado, se obtuvo un sistema capaz de proporcionar consultas académicas personalizadas e información institucional mediante un asistente conversacional. El sistema permite acceder a información personalizada, realizar consultas mediante texto o voz y gestionar procesos de autenticación y monitoreo de actividades.

Las interfaces implementadas evidenciaron los incrementos funcionales desarrollados durante los *Sprints*, ya que integran en un mismo entorno el acceso al chatbot, la autenticación de usuarios, las consultas académicas personalizadas, la interacción por voz y el monitoreo de eventos. Esta integración muestra la correspondencia entre los componentes visuales del sistema y las funcionalidades planificadas en el *Product Backlog*.

A continuación, se presentan las principales interfaces obtenidas como resultado de la implementación del sistema.

3.1.1 Interfaz principal

La Ilustración 34 presenta la interfaz principal de la plataforma académica inteligente. Como resultado del desarrollo realizado, el sistema incorpora un asistente conversacional accesible desde la página principal, permitiendo a los usuarios realizar consultas relacionadas con información institucional mediante lenguaje natural.

A través de esta interfaz, los usuarios pueden acceder a información sobre servicios universitarios, procesos académicos, requisitos de admisión, normativas institucionales, ubicación de dependencias y otros aspectos de interés general para la comunidad universitaria. Asimismo, cuando una consulta requiere información académica

personalizada, el sistema informa al usuario que debe autenticarse previamente para acceder a los datos asociados a su perfil.

El acceso al asistente conversacional se realiza mediante el ícono flotante ubicado en la esquina inferior derecha de la pantalla, señalado en la Ilustración 34. Al seleccionar este ícono, se despliega la ventana de conversación desde la cual el usuario puede interactuar con el chatbot mediante consultas escritas o comandos de voz.

La interfaz principal constituye el punto central de interacción entre los usuarios y el asistente conversacional, facilitando el acceso a la información institucional desde un único espacio dentro de la plataforma.

Ilustración 34 Interfaz principal de la plataforma y chatbot institucional.

3.1.2. Interfaz de consultas institucionales mediante arquitectura RAG

La interfaz de consultas institucionales correspondió al modo público del chatbot, es decir, al escenario en el que el usuario podía realizar consultas generales sin autenticarse. En este entorno, el asistente respondió preguntas relacionadas con servicios institucionales, procesos académicos y orientación dentro del campus, utilizando la base

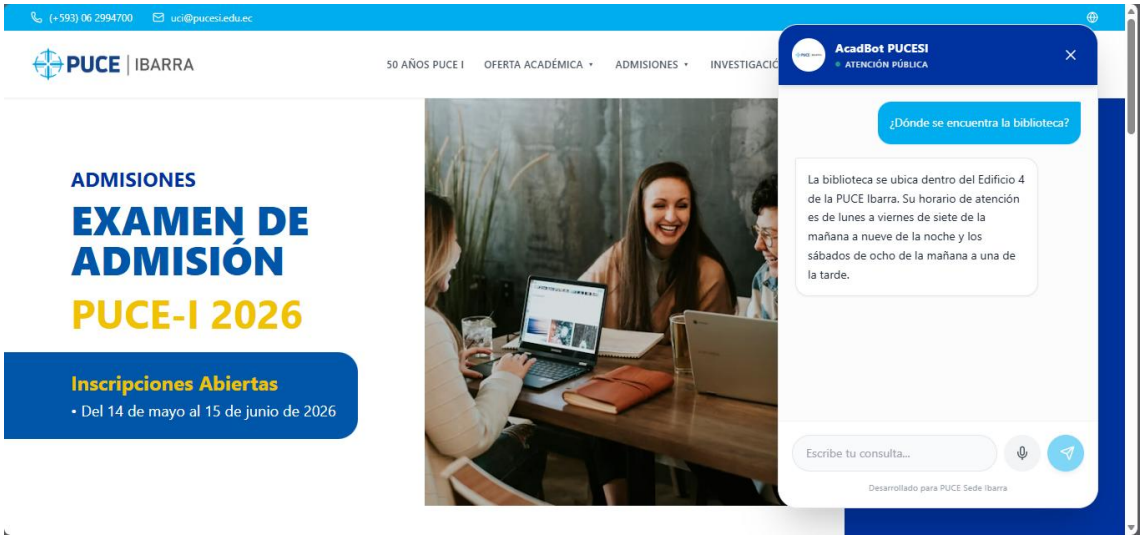
de conocimiento procesada mediante arquitectura *Retrieval-Augmented Generation* (RAG).

En este modo de uso, el sistema no accedió a información académica personalizada del estudiante, sino únicamente a documentos institucionales previamente curados e indexados. Esta separación permitió diferenciar las consultas públicas de las consultas privadas asociadas al estudiante autenticado, manteniendo la protección de los datos personales y académicos.

Las respuestas generadas por el chatbot se fundamentaron en fragmentos recuperados desde la base documental activa. Cuando una consulta requería confirmación institucional o hacía referencia a información que podía variar, el sistema orientó al usuario a verificar el dato con la dependencia correspondiente, evitando presentar información no confirmada como definitiva.

Como se observa en la Ilustración 35, el chatbot respondió una consulta relacionada con la ubicación de la biblioteca institucional. Esta interacción permitió evidenciar que el sistema puede recuperar información de orientación general desde la base de conocimiento institucional y presentarla al usuario en lenguaje natural. En caso de que la ubicación exacta requiera confirmación por cambios internos del campus, la respuesta orienta al estudiante a verificar la información con los puntos oficiales de atención.

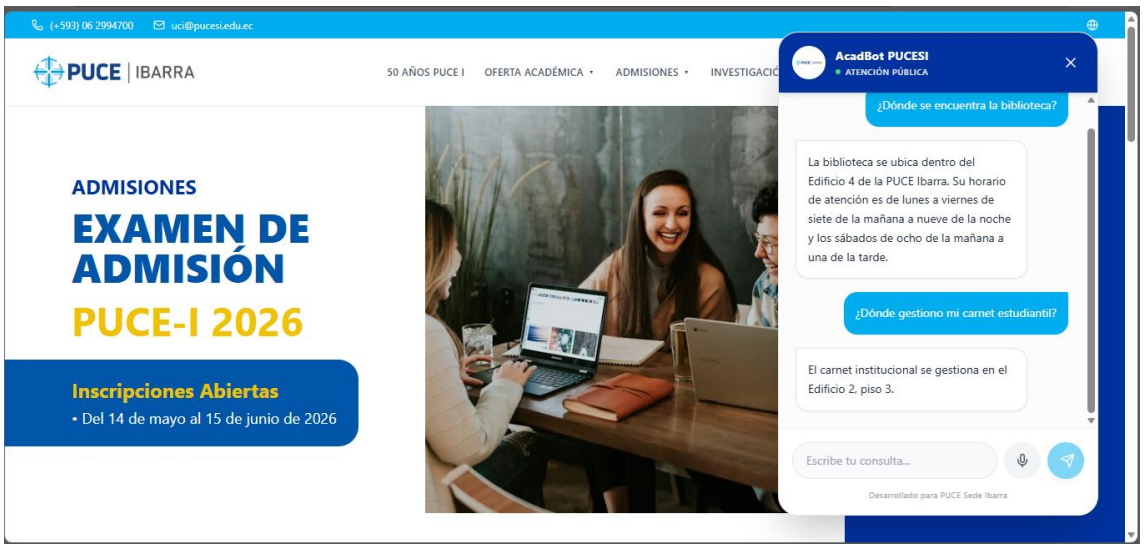
Ilustración 35 Consulta sobre la ubicación de la biblioteca institucional.



La Ilustración 36 muestra una consulta referente al proceso de gestión del carnet estudiantil. A través de esta interacción, el chatbot proporcionó información general sobre el trámite y orientó al usuario hacia las dependencias institucionales

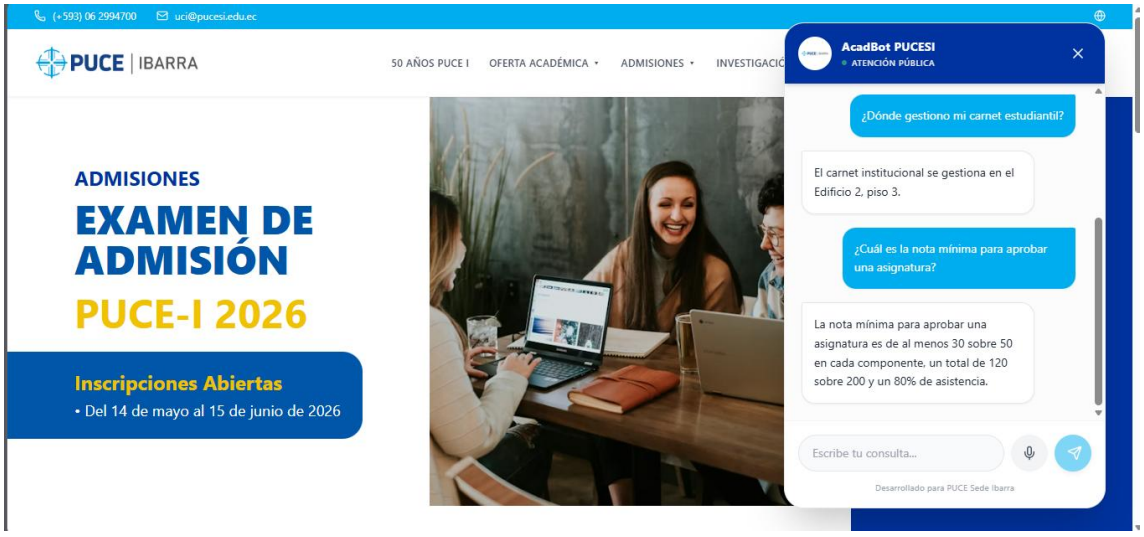
correspondientes cuando el procedimiento requería confirmación administrativa. Este comportamiento resulta adecuado para consultas públicas, debido a que evita entregar datos no verificados como si fueran definitivos.

Ilustración 36 Consulta sobre el proceso de gestión del carnet estudiantil.



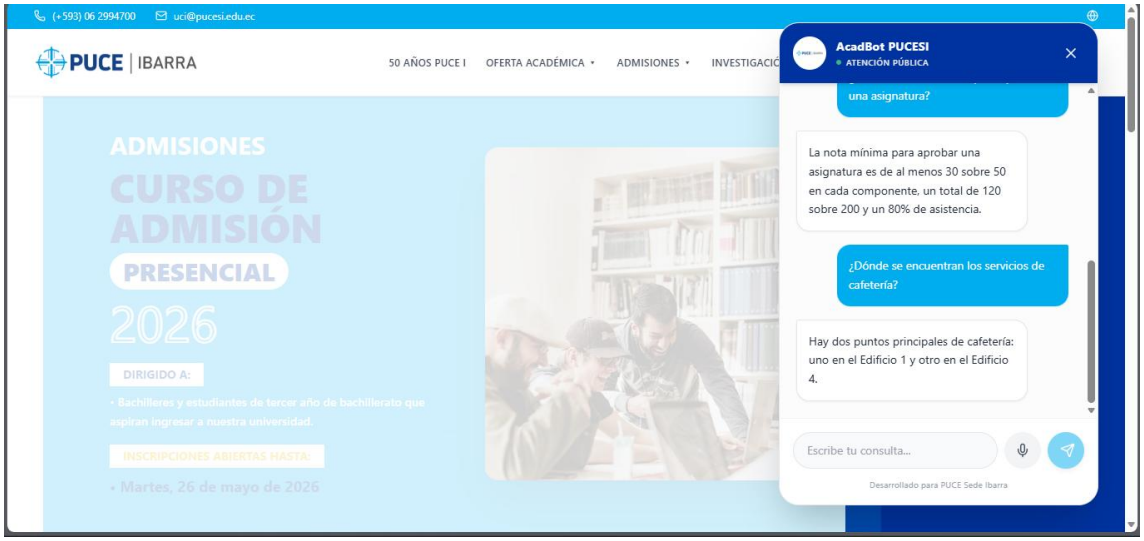
En la Ilustración 37 se presenta una consulta relacionada con la nota mínima requerida para aprobar una asignatura. La respuesta permitió evidenciar que el sistema puede recuperar información normativa o académica general desde la base documental institucional. Este tipo de consulta corresponde al modo público, ya que no requiere acceder al historial académico individual del estudiante.

Ilustración 37 Consulta sobre la nota mínima requerida para aprobar una asignatura.



La Ilustración 38 evidencia una consulta sobre la ubicación de servicios de cafetería dentro o cerca del entorno universitario. En este caso, el chatbot respondió con información de orientación general basada en la documentación institucional activa. Cuando la información corresponde a servicios que pueden cambiar de ubicación, disponibilidad u horario, el sistema debe presentarla como referencia y recomendar la verificación con los canales institucionales correspondientes.

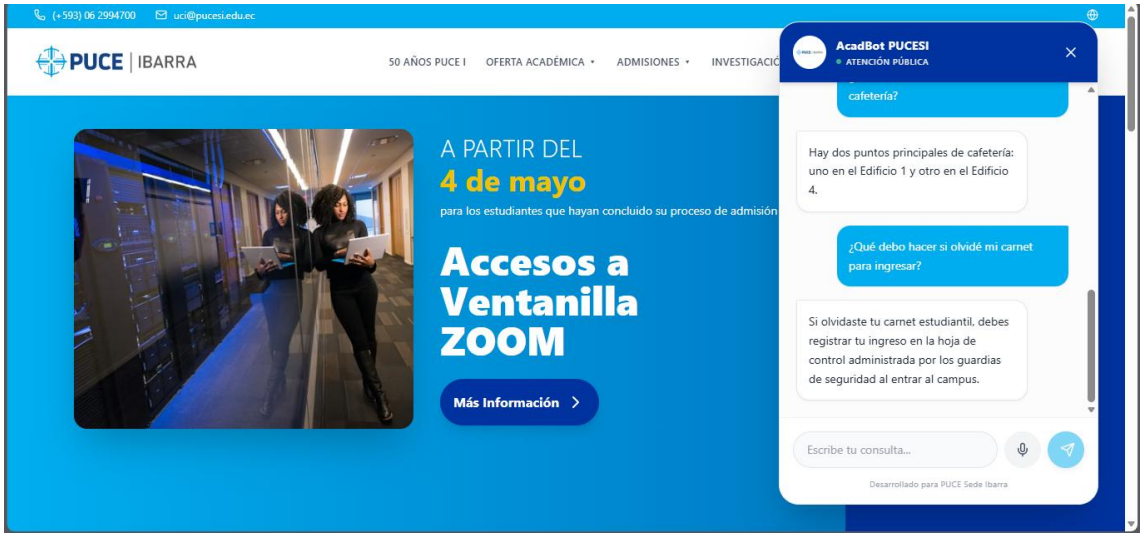
Ilustración 38 Consulta sobre la ubicación de los servicios de cafetería.



Finalmente, la Ilustración 39 muestra una consulta sobre el procedimiento a seguir ante la pérdida u olvido del carnet estudiantil. La respuesta del chatbot permitió orientar al usuario sobre el tipo de gestión que debe realizar, manteniendo un enfoque informativo y derivando la confirmación del trámite a la dependencia institucional responsable. Esta

respuesta demuestra la utilidad del RAG para consultas frecuentes de orientación estudiantil sin requerir autenticación.

Ilustración 39 Consulta sobre el procedimiento ante la pérdida u olvido del carnet estudiantil.



En conjunto, las consultas presentadas evidencian que el chatbot institucional permitió atender preguntas frecuentes mediante recuperación documental, manteniendo una separación clara entre información pública e información académica personalizada. Las respuestas generadas se apoyaron en la base de conocimiento activa del sistema y fueron presentadas en lenguaje natural, lo que facilitó la orientación inicial del estudiante dentro del entorno universitario.

Asimismo, esta sección permitió comprobar que el modo público del chatbot resulta adecuado para consultas generales, pero no expone información privada ni accede a datos académicos individuales. Las consultas personalizadas, como horarios, calificaciones, actividades o notificaciones del estudiante se reservaron para el modo autenticado del sistema.

3.1.3. Interfaz de autenticación y acceso seguro al sistema

La Ilustración 40 presenta la interfaz de autenticación de la plataforma académica inteligente. Esta interfaz permite a estudiantes, docentes y administradores acceder a las funcionalidades personalizadas del sistema mediante el ingreso de sus credenciales de acceso.

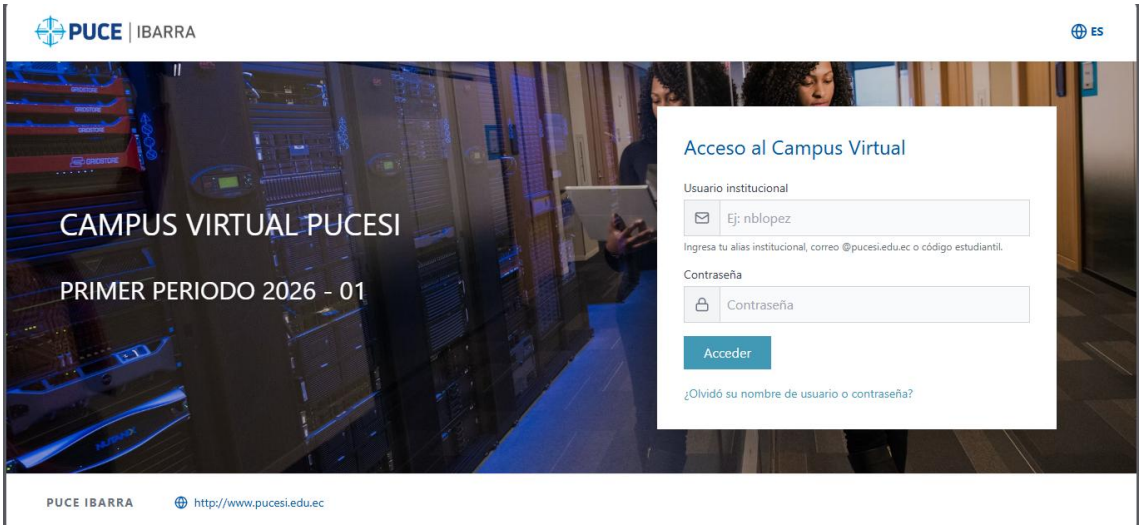
A través de este mecanismo, los usuarios pueden acceder a información académica privada y recursos restringidos de acuerdo con su perfil dentro de la plataforma. Como

parte de las medidas de seguridad implementadas, el sistema incorpora un mecanismo de autenticación de dos factores (2FA), el cual requiere una verificación adicional mediante un código de un solo uso (OTP) antes de conceder el acceso definitivo a la cuenta.

Una vez que las credenciales son validadas correctamente, el usuario es redirigido a la interfaz de verificación mostrada en la Ilustración 41, donde debe ingresar el código OTP generado para completar el proceso de autenticación. Este procedimiento añade una capa adicional de protección frente a accesos no autorizados y fortalece la seguridad de la información académica y administrativa gestionada por la plataforma.

La interfaz de autenticación constituye el punto de acceso a los servicios personalizados del sistema, garantizando que únicamente los usuarios autorizados puedan acceder a los recursos y funcionalidades disponibles según su rol dentro de la plataforma.

Ilustración 40 Interfaz de autenticación y acceso seguro al sistema.



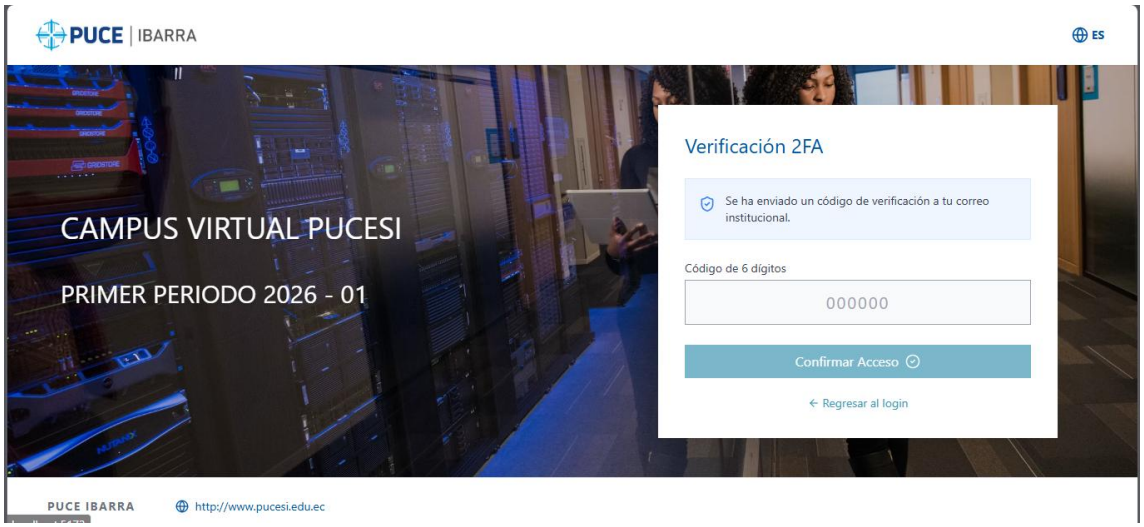
3.1.3.1. Interfaz de verificación mediante autenticación de dos factores (2FA)

La Ilustración 41 presenta la interfaz de verificación de segundo factor implementada en la plataforma. Esta pantalla se muestra después de que el usuario ha ingresado correctamente sus credenciales de acceso y tiene como objetivo validar su identidad mediante un código de un solo uso (OTP).

A través de esta interfaz, el usuario debe ingresar el código temporal generado para su cuenta antes de acceder a los recursos protegidos del sistema. Una vez validado el código, la plataforma habilita el acceso a las funcionalidades correspondientes según el perfil autenticado.

La incorporación de este mecanismo fortalece la seguridad del proceso de autenticación al requerir una segunda evidencia de identidad además de la contraseña, reduciendo el riesgo de accesos no autorizados y mejorando la protección de la información académica y administrativa gestionada por el sistema.

Ilustración 41 Interfaz de verificación mediante código OTP (2FA).

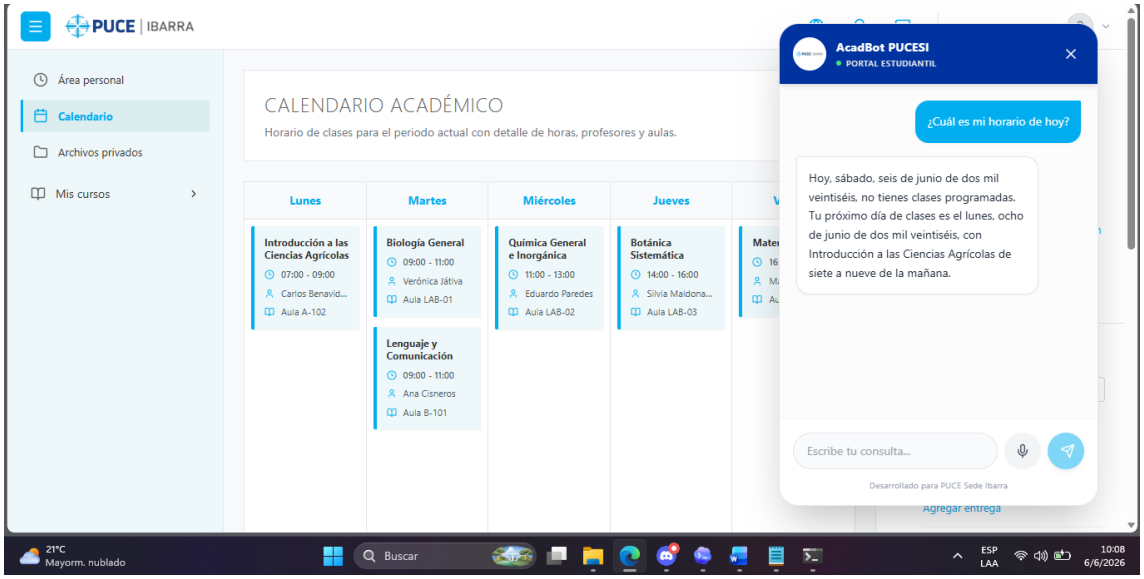


3.1.4. Interfaz de consultas académicas personalizadas

La interfaz de consultas académicas personalizadas integra diferentes mecanismos de acceso a información académica asociada al estudiante autenticado. A través de esta funcionalidad, el usuario puede realizar consultas relacionadas con horarios de clases, calificaciones, actividades pendientes, asignaturas matriculadas, promedio académico, ubicación de aulas y planificación académica futura, utilizando lenguaje natural como medio de interacción.

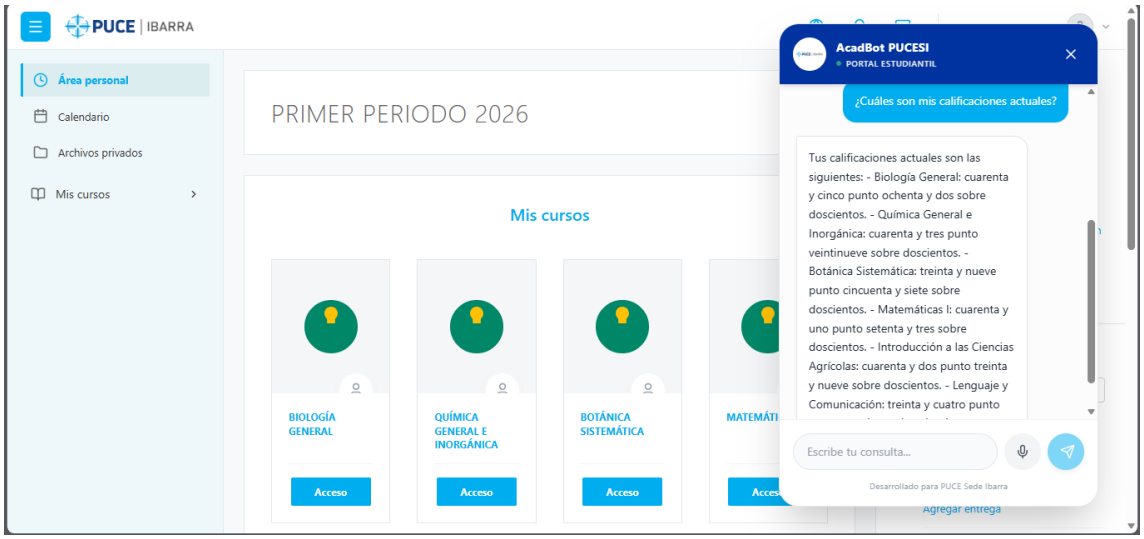
Como se observa en la Ilustración 42, la interfaz permite visualizar la información correspondiente al horario académico del estudiante mediante una respuesta generada por el chatbot a partir de los datos registrados en el sistema.

Ilustración 42 Respuesta del chatbot sobre el horario académico del estudiante.



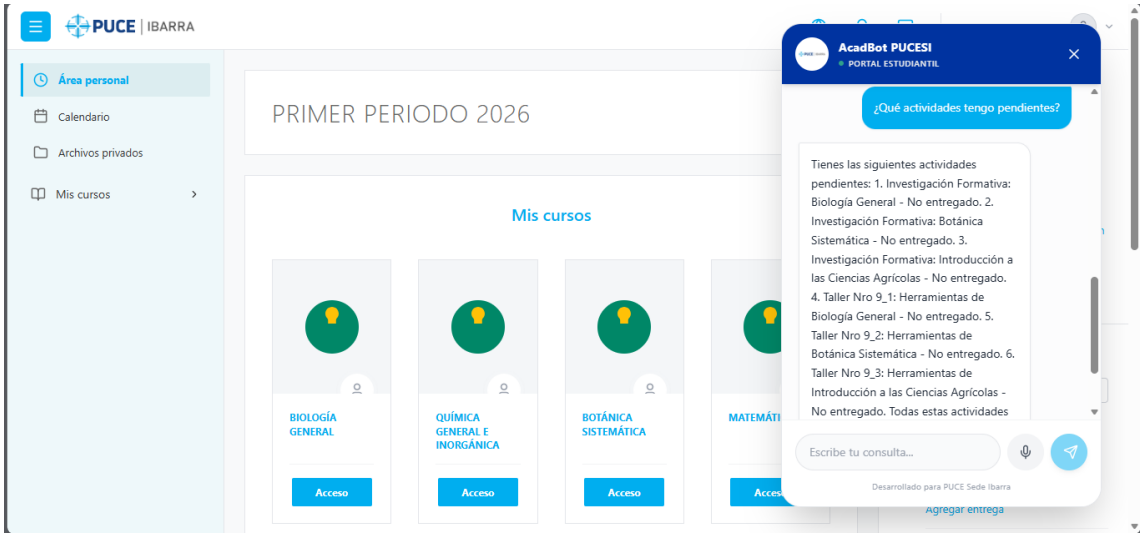
En la Ilustración 43 se evidencia la consulta de calificaciones académicas, donde el chatbot recupera y presenta las notas asociadas al perfil del estudiante autenticado.

Ilustración 43 Respuesta del chatbot sobre las calificaciones actuales del estudiante.



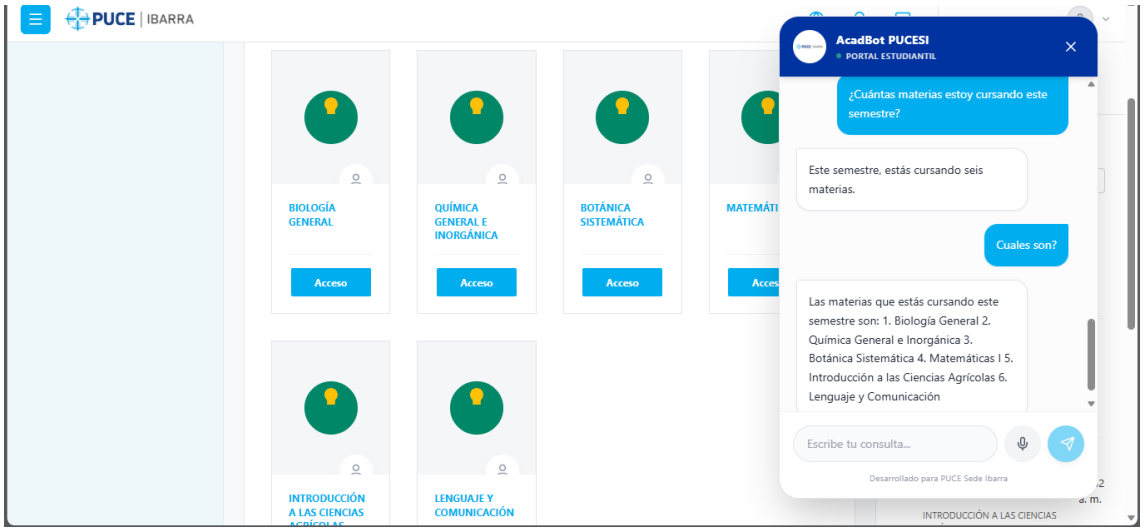
La Ilustración 44 muestra el resultado de una consulta relacionada con actividades académicas pendientes, permitiendo al estudiante identificar tareas o compromisos registrados dentro del sistema.

Ilustración 44 Respuesta del chatbot sobre las actividades académicas pendientes.



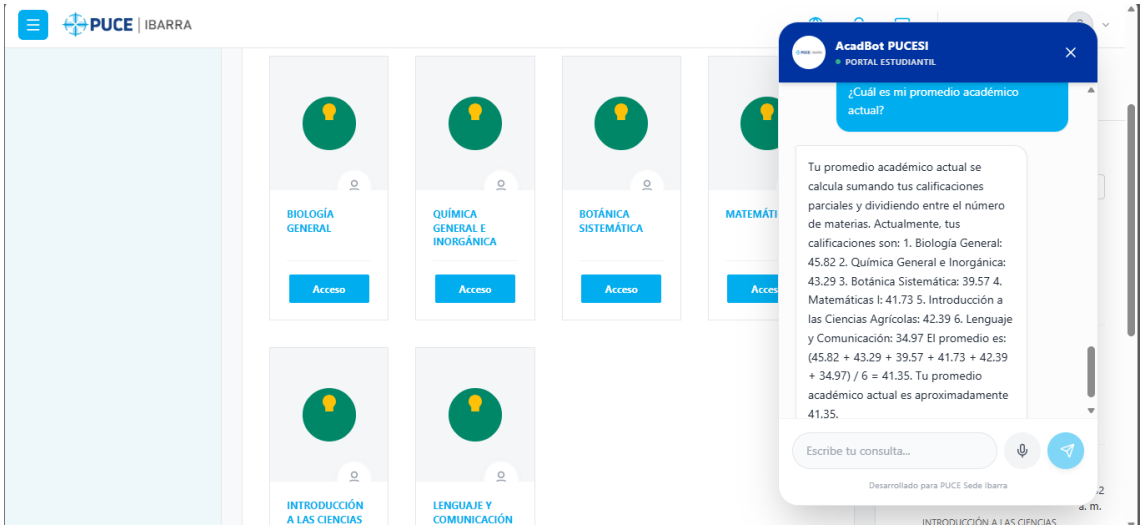
Como se aprecia en la Ilustración 45, el chatbot respondió a una consulta sobre las asignaturas matriculadas durante el periodo académico, proporcionando información personalizada del estudiante.

Ilustración 45 Respuesta del chatbot sobre las asignaturas matriculadas en el semestre.



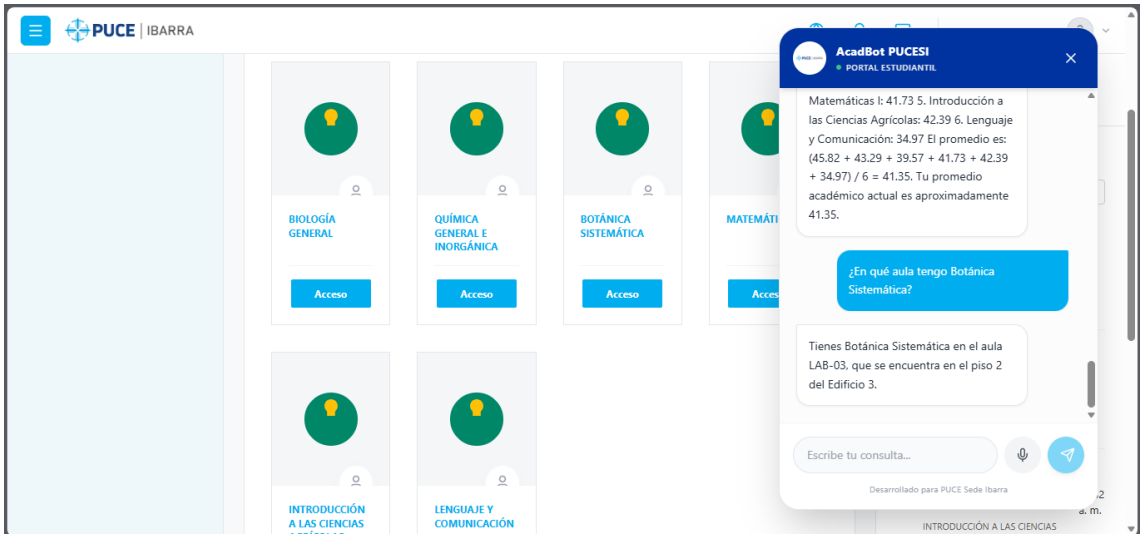
La Ilustración 46 presenta el resultado de una consulta orientada a obtener el promedio académico actual del estudiante, permitiendo visualizar un indicador general de rendimiento académico.

Ilustración 46 Respuesta del chatbot sobre el promedio académico actual.



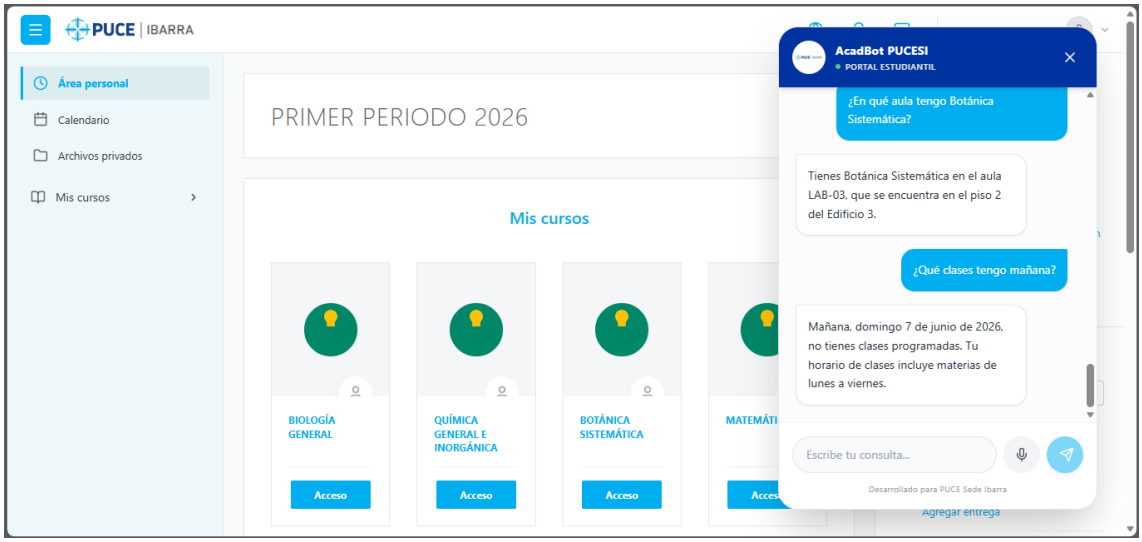
En la Ilustración 47 se observa una consulta relacionada con la ubicación del aula asignada a una materia específica, información que facilita la organización y movilidad del estudiante dentro de la institución.

Ilustración 47 Respuesta del chatbot sobre el aula asignada a una asignatura específica.



Finalmente, la Ilustración 48 muestra una consulta sobre las clases programadas para el día siguiente, demostrando la capacidad del sistema para recuperar información académica prospectiva y apoyar la planificación de actividades estudiantiles.

Ilustración 48 Respuesta del chatbot sobre las clases programadas para el día siguiente.



3.1.5. Interfaz de interacción por voz

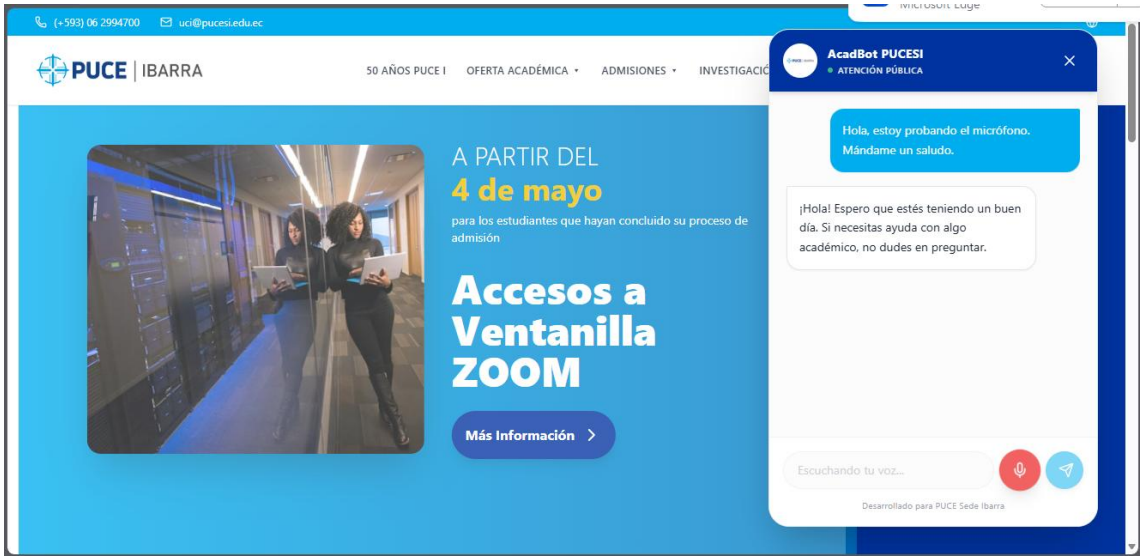
La Ilustración 49 presenta la funcionalidad de interacción por voz incorporada al asistente conversacional de la plataforma. Esta característica permite que los usuarios realicen consultas académicas e institucionales mediante comandos hablados, proporcionando una alternativa al mecanismo tradicional de interacción basado en texto.

A través de esta interfaz, el usuario puede utilizar el micrófono para formular preguntas de manera natural y obtener respuestas generadas por el asistente conversacional. Para ello, debe mantener presionado el botón del micrófono mientras realiza la consulta y liberarlo una vez finalizada la intervención. Posteriormente, el sistema procesa la información recibida y genera la respuesta correspondiente dentro de la conversación.

La funcionalidad de voz se encuentra integrada con los mismos servicios utilizados para las consultas textuales, permitiendo acceder a información académica personalizada e información institucional desde una única interfaz conversacional. De esta manera, el usuario puede interactuar con el sistema utilizando diferentes modalidades de entrada sin modificar el flujo general de uso de la plataforma.

La incorporación de esta funcionalidad amplía las opciones de interacción disponibles para los usuarios y contribuye a mejorar la accesibilidad del sistema, facilitando el acceso a la información mediante una experiencia conversacional más natural y flexible.

Ilustración 49 Interfaz de interacción por voz integrada al asistente conversacional.



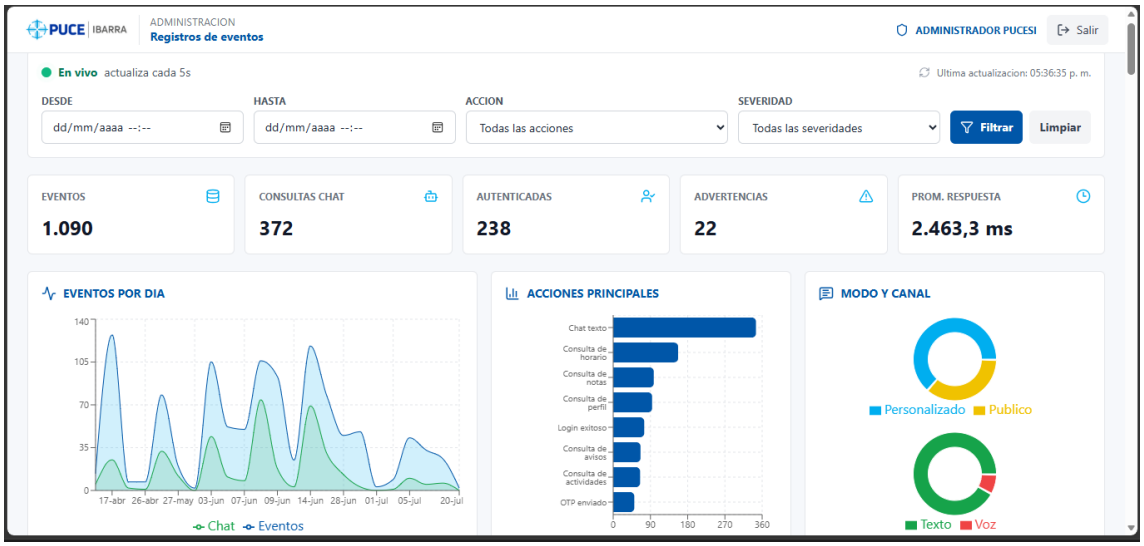
3.1.6. Interfaz de registros de eventos y monitoreo del sistema

La Ilustración 50 presenta la interfaz de registros de eventos y monitoreo disponible dentro de la plataforma académica inteligente. Esta funcionalidad permite visualizar los eventos generados durante el uso del sistema, facilitando el seguimiento y la trazabilidad técnica de las actividades realizadas por los usuarios.

A través de esta interfaz es posible consultar información relacionada con accesos a la plataforma, consultas efectuadas al asistente conversacional y otras acciones registradas durante la operación normal del sistema. Los registros mostrados permiten disponer de un historial organizado de eventos para fines de supervisión y control.

La interfaz de registros de eventos y monitoreo constituye un mecanismo de apoyo para la gestión técnica de la plataforma, al proporcionar información relevante sobre las actividades realizadas dentro del entorno académico y facilitar el seguimiento de los eventos generados durante el uso del sistema.

Ilustración 50 Interfaz de registros de eventos y monitoreo del sistema.



3.2. Pruebas del sistema

La validación del sistema se realizó con el propósito de comprobar el funcionamiento de los componentes principales del prototipo desarrollado. Para ello, se ejecutaron pruebas funcionales y no funcionales orientadas a verificar la interacción conversacional, la recuperación de información institucional mediante arquitectura RAG, las consultas académicas personalizadas, la interacción por voz, la autenticación segura y el registro de eventos.

Las pruebas se desarrollaron en un entorno controlado, utilizando datos académicos simulados correspondientes a estudiantes de primer nivel de la carrera de Ingeniería. Esta decisión permitió comprobar el comportamiento técnico del sistema sin utilizar información académica real de estudiantes. La validación se centró en la experiencia del visitante público y del estudiante autenticado, manteniendo los módulos docentes y administrativos como componentes de soporte operativo fuera del alcance funcional principal.

Cada caso de prueba fue evaluado considerando la funcionalidad evaluada, el resultado esperado, el resultado obtenido y el estado final de la prueba. Los resultados se organizaron en las Tablas 14, 15, 16, 17 y 18, correspondientes a la validación de consultas institucionales mediante arquitectura RAG, consultas académicas personalizadas, interfaz conversacional, interacción por voz, seguridad y registros de eventos.

A continuación, se presentan los resultados obtenidos durante la ejecución de las pruebas funcionales y no funcionales realizadas sobre la plataforma.

3.2.1. Pruebas Funcionales

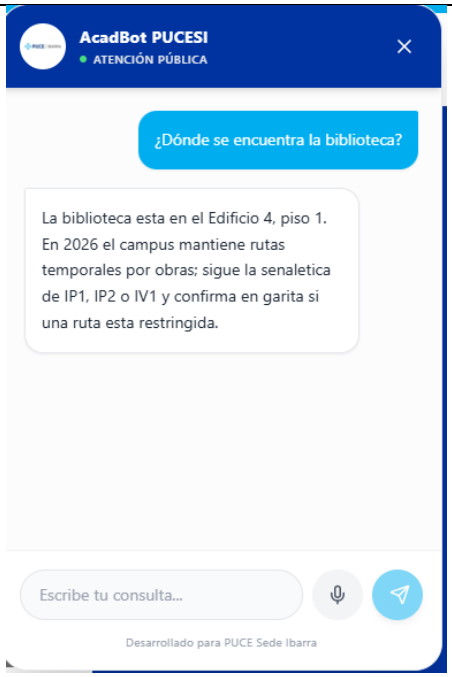
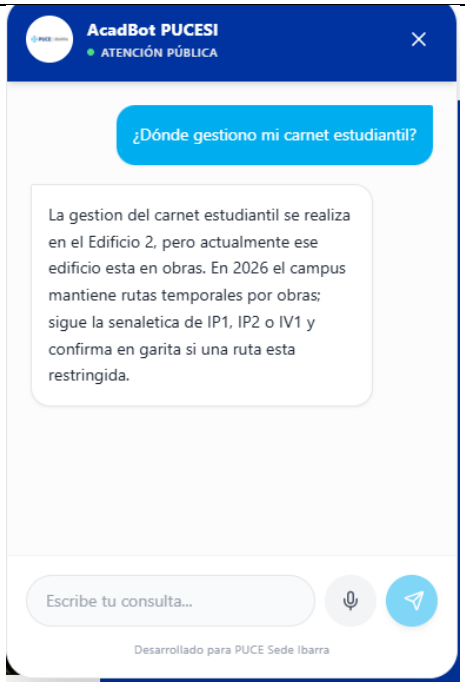
3.2.1.1. Validación de consultas institucionales mediante arquitectura RAG

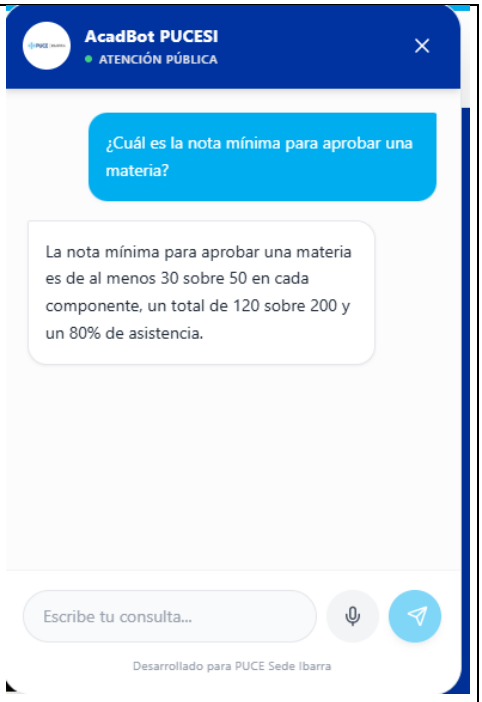
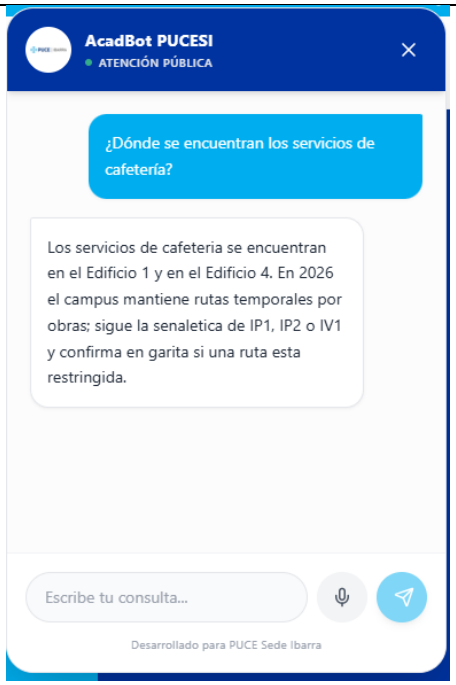
La validación de consultas institucionales mediante arquitectura RAG tuvo como finalidad comprobar que el sistema pudiera recuperar información relevante desde la base documental activa y utilizarla como contexto para generar respuestas institucionales en lenguaje natural. Estas pruebas se aplicaron en el modo público del chatbot, sin requerir autenticación del usuario, debido a que las consultas evaluadas correspondieron a información general de orientación académica e institucional. Además de verificar la pertinencia de las respuestas, se incorporó la medición del tiempo de respuesta como parte del comportamiento funcional del flujo de consulta, considerando que el chatbot general debía responder en un tiempo promedio máximo de 5 segundos. Para ello, se ejecutaron 10 pruebas por texto, obteniendo un promedio de 4.875,40 ms, equivalente a 4,88 segundos, por lo que el criterio se cumplió por promedio. En cambio, en la interacción por voz se obtuvo un promedio de 6.227,80 ms, equivalente a 6,23 segundos, superando el tiempo esperado debido a procesos adicionales como captura de audio, transcripción, procesamiento de la consulta, generación de respuesta y reproducción del resultado.

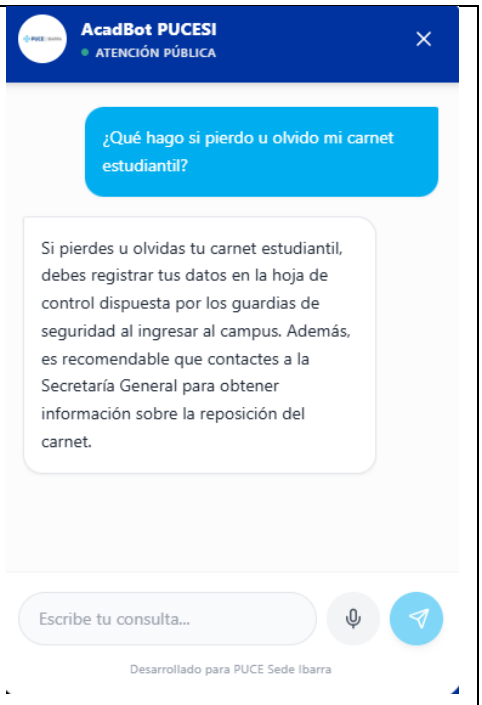
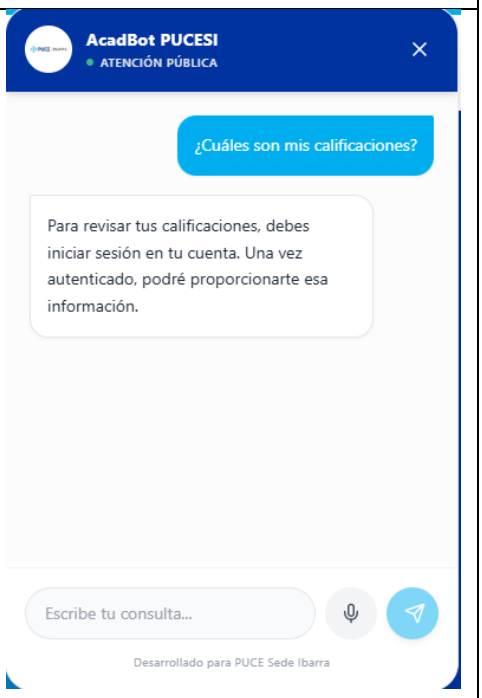
Durante esta validación se consideraron consultas relacionadas con servicios universitarios, procesos administrativos, normativa académica y orientación dentro del entorno institucional. En cada caso se verificó que el chatbot recuperara información desde documentos previamente procesados y que la respuesta generada fuera coherente con la consulta realizada. Los resultados de esta validación se presentan en la Tabla 14.

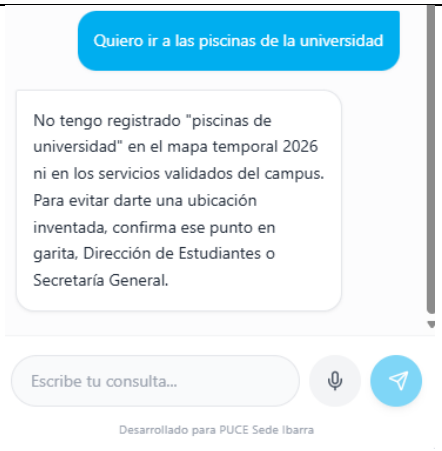
Tabla 14 Resultados de validación de consultas institucionales mediante arquitectura RAG.

PRUEBA DE FUNCIONALIDAD N° 1			
Nombre: Consultas institucionales mediante RAG			
Descripción: El chatbot debe responder preguntas institucionales utilizando información recuperada desde documentos oficiales indexados.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Obtenido***

Consulta sobre biblioteca institucional	*¿Dónde se encuentra la biblioteca?*	*El sistema debe generar una respuesta de orientación institucional relacionada con la biblioteca.*	
Consulta sobre carnet estudiantil	*¿Dónde gestiono mi carnet estudiantil?*	*El chatbot debe orientar al usuario sobre el trámite o dependencia relacionada con la gestión del carnet.*	

Consulta normativa académica	*¿Cuál es la nota mínima para aprobar una materia?*	*El sistema debe recuperar información académica general relacionada con criterios de aprobación.*	
Consulta sobre servicios de cafetería	*¿Dónde se encuentran los servicios de cafetería?*	*El sistema debe entregar información general de orientación sobre los servicios consultados.*	

Consulta sobre pérdida u olvido del carnet	*¿Qué hago si pierdo u olvido mi carnet estudiantil?*	*El chatbot debe orientar al usuario sobre el procedimiento general a seguir.*	
Restricción de información privada	*¿Cuáles son mis calificaciones ?*	*El modo público no debe mostrar información personalizada ni acceder a datos académicos privados.*	

Consulta institucional no disponible	*Pregunta sobre información no contenida en la base documental*	*El sistema debe evitar inventar información y orientar al usuario a verificar con la dependencia correspondiente .*	
Tiempo de respuesta por texto	*10 consultas institucionales por texto*	*El tiempo promedio de respuesta del chatbot general debe ser igual o menor a 5 segundos.*	*Se obtuvo un promedio de 4.875,40 ms, equivalente a 4,88 segundos, correspondiente al 97,51 % del límite establecido.*
Tiempo de respuesta por voz	*10 consultas institucionales por voz*	*El tiempo promedio de respuesta del chatbot general debe ser igual o menor a 5 segundos.*	*Se obtuvo un promedio de 6.227,80 ms, equivalente a 6,23 segundos, correspondiente al 124,56 % del límite establecido.*
Observaciones: Esta prueba funcional valida la recuperación semántica mediante FAISS, el uso de *embeddings*, la construcción de contexto y la generación de respuestas fundamentadas mediante el modelo de lenguaje.			

Los resultados de la Tabla 14 evidencian que el chatbot institucional cumplió funcionalmente con la recuperación de información mediante arquitectura RAG, la generación de respuestas contextualizadas y la restricción de acceso a información privada desde el modo público. En cuanto al tiempo de respuesta, las consultas por texto cumplieron el criterio establecido para el chatbot general, con un promedio de 4,88 segundos frente al límite máximo de 5 segundos. Sin embargo, las consultas por voz

superaron el límite definido, con un promedio de 6,23 segundos. Este resultado se explica porque la interacción por voz incorpora etapas adicionales de captura, transcripción, procesamiento del lenguaje, generación de respuesta y reproducción del audio, además de depender de la latencia de red y del tiempo de respuesta de servicios externos.

Como se observa en la Tabla 14, los casos de prueba permitieron comprobar que el chatbot institucional respondió de forma adecuada ante consultas públicas relacionadas con orientación estudiantil, servicios y procesos académicos generales. Los resultados obtenidos evidenciaron que el sistema utilizó la base documental activa para generar respuestas contextualizadas, evitando acceder a información privada del estudiante.

En conjunto, los resultados presentados en la Tabla 14 permiten confirmar que la arquitectura RAG fue útil para responder consultas institucionales frecuentes. Además, el sistema mantuvo un comportamiento prudente ante información que puede variar con el tiempo, orientando al usuario a confirmar ciertos datos con las dependencias institucionales correspondientes cuando fue necesario.

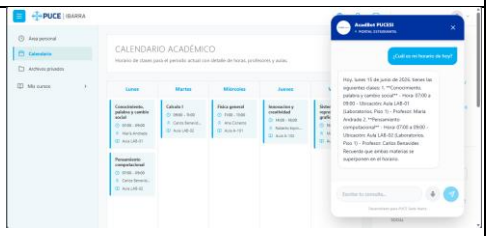
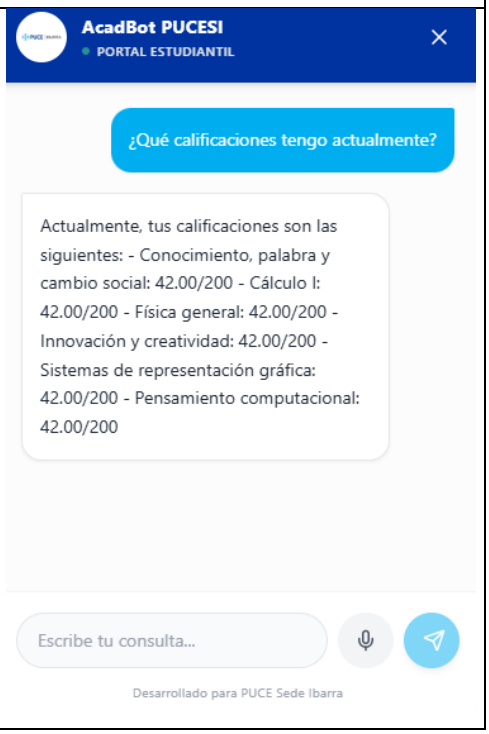
3.2.1.2. Validación de consultas académicas personalizadas

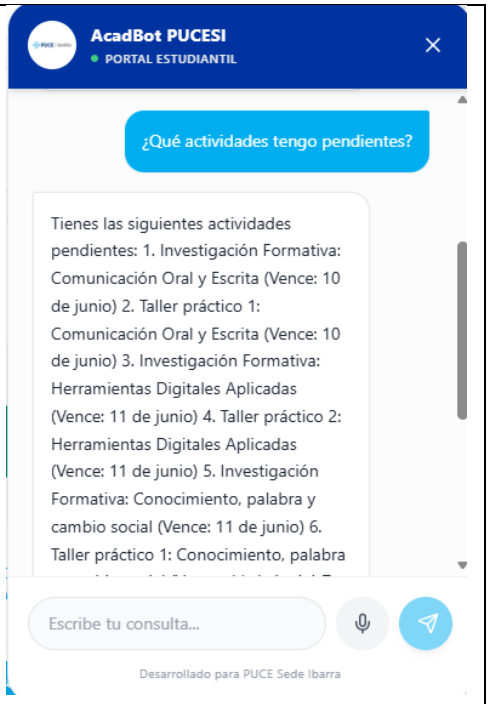
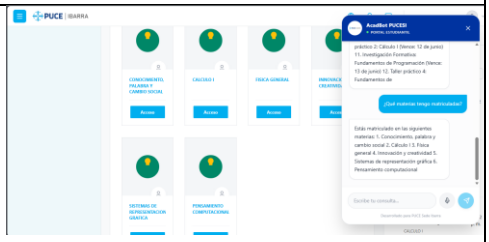
La validación de consultas académicas personalizadas tuvo como finalidad comprobar que el chatbot pudiera recuperar información específica del estudiante autenticado y generar respuestas relacionadas con su contexto académico, tales como horarios, asignaturas, aulas, calificaciones, actividades pendientes y notificaciones. Estas pruebas se realizaron dentro del entorno autenticado del sistema, utilizando datos académicos simulados almacenados en PostgreSQL. Además de validar la recuperación correcta de información personalizada, se midió el tiempo de respuesta del flujo conversacional, considerando que el chatbot personalizado debía responder en un tiempo promedio máximo de 3 segundos. A partir de las 10 pruebas realizadas, el tiempo promedio por texto fue de 4.875,40 ms, equivalente a 4,88 segundos, mientras que el promedio por voz fue de 6.227,80 ms, equivalente a 6,23 segundos. Por tanto, el sistema cumplió funcionalmente con la entrega de información académica personalizada, pero no alcanzó el criterio de eficiencia establecido para el tiempo de respuesta.

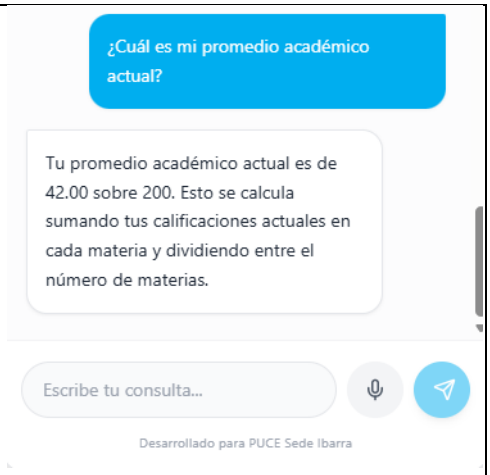
Durante esta validación se evaluaron consultas relacionadas con horarios, calificaciones, asignaturas, actividades pendientes, notificaciones, promedio académico y ubicación de aulas. En cada caso se verificó que el sistema respondiera únicamente con información

vinculada al estudiante autenticado, manteniendo la separación entre información pública e información privada. Los resultados obtenidos se presentan en la Tabla 15.

Tabla 15 Resultados de validación de consultas académicas personalizadas.

<table>
<tr><th colspan="4">PRUEBA DE FUNCIONALIDAD N° 2</th></tr>
<tr><td colspan="4">Nombre: Consulta académica personalizada</td></tr>
<tr><td colspan="4">Descripción: El chatbot debe permitir consultar información académica personalizada del estudiante autenticado mediante lenguaje natural, incluyendo horarios, asignaturas, aulas, calificaciones, actividades pendientes y promedio académico.</td></tr>
<tr><td colspan="4">Escenarios</td></tr>
<tr><td>Escenario evaluado</td><td>Campo de Entrada</td><td>Criterio de aceptación</td><td>Resultado Obtenido</td></tr>
<tr><td>Consulta de horario académico</td><td>¿Cuál es mi horario de hoy?</td><td>El sistema debe recuperar y mostrar el horario correspondiente al estudiante autenticado.</td><td>  </td></tr>
<tr><td>Consulta de calificaciones</td><td>¿Qué calificaciones tengo actualmente?</td><td>El chatbot debe mostrar las calificaciones registradas para el estudiante autenticado.</td><td> 
</td></tr>
</table>

Consulta de actividades pendientes	*¿Qué actividades tengo pendientes?*	*El sistema debe listar las actividades académicas pendientes asociadas al estudiante.*	
Consulta de asignaturas matriculadas	*¿Qué materias tengo matriculadas?*	*El chatbot debe mostrar las asignaturas registradas en el periodo académico correspondiente.*	
Consulta de aula asignada	*¿En qué aula tengo [MATERIA]?*	*El sistema debe identificar la asignatura consultada y mostrar el aula correspondiente.*	

Consulta de promedio académico	*¿Cuál es mi promedio académico actual?*	*El sistema debe calcular o recuperar el promedio académico del estudiante autenticado.*	
Tiempo de respuesta por texto	*10 consultas académicas personalizadas por texto*	*El tiempo promedio del chatbot personalizado debe ser igual o menor a 3 segundos.*	*Se obtuvo un promedio de 4.875,40 ms, equivalente a 4,88 segundos, correspondiente al 162,51 % del límite establecido.*
Tiempo de respuesta por voz	*10 consultas académicas personalizadas por voz*	*El tiempo promedio del chatbot personalizado debe ser igual o menor a 3 segundos.*	*Se obtuvo un promedio de 6.227,80 ms, equivalente a 6,23 segundos, correspondiente al 207,59 % del límite establecido.*
Observaciones: Esta prueba funcional valida la recuperación de información académica personalizada desde la base de datos, el control de sesión del estudiante y la correspondencia entre la consulta procesada y los datos académicos registrados.			

Los resultados de la Tabla 15 evidencian que el chatbot personalizado cumplió funcionalmente con la recuperación de información académica desde PostgreSQL, permitiendo responder consultas sobre horarios, asignaturas, aulas, calificaciones, actividades pendientes y promedio académico. Asimismo, se comprobó que el sistema restringió el acceso a información privada cuando no existía una sesión autenticada válida. Sin embargo, los tiempos de respuesta no cumplieron el criterio definido para el chatbot personalizado, cuyo límite máximo era de 3 segundos. Las consultas por texto

registraron un promedio de 4,88 segundos, mientras que las consultas por voz alcanzaron un promedio de 6,23 segundos. Este comportamiento se relaciona con la intervención del *backend*, la consulta a la base de datos, la validación de sesión, el historial conversacional, la generación de respuesta en lenguaje natural y, en el caso de voz, los procesos adicionales de captura, transcripción y reproducción de audio.

Como se observa en la Tabla 15, los casos de prueba permitieron comprobar que el sistema respondió funcionalmente a consultas académicas personalizadas utilizando información asociada al estudiante autenticado, como horarios, calificaciones, asignaturas, actividades y aulas. Sin embargo, los tiempos promedio registrados en las consultas académicas personalizadas superaron el límite esperado de 3 segundos. Por esta razón, los resultados deben interpretarse como funcionalmente correctos respecto a la entrega de información académica, pero no conformes respecto al criterio de eficiencia definido para el tiempo de respuesta.

En conjunto, los resultados presentados en la Tabla 15 permiten confirmar que el sistema mantuvo una separación adecuada entre el modo público y el modo autenticado. Las consultas personalizadas solo fueron atendidas cuando existió una sesión válida, lo que contribuyó a proteger la información académica individual del estudiante.

3.2.2. Pruebas no funcionales

3.2.2.1. Validación de la interfaz conversacional

La validación de la interfaz conversacional tuvo como finalidad comprobar que el usuario pudiera interactuar correctamente con el chatbot desde los dos escenarios principales definidos para el prototipo: la página pública y el *dashboard* académico del estudiante autenticado. En esta prueba se verificó el envío de consultas textuales, la visualización de respuestas, la apertura y cierre del widget conversacional, la actualización del historial visible y la disponibilidad del asistente tanto en modo público como en modo personalizado.

En el modo público, se comprobó que el visitante pudiera realizar consultas institucionales generales sin iniciar sesión. En el modo autenticado, se verificó que el estudiante de primer nivel pudiera utilizar el chatbot desde su *dashboard* académico, manteniendo una experiencia visual coherente con el resto de la plataforma. Los resultados de esta validación se presentan en la Tabla 16.

Tabla 16 Resultados de validación de la interfaz conversacional.

PRUEBA DE USABILIDAD N° 1			
Nombre: Interfaz conversacional inteligente			
Descripción: El *frontend* del chatbot debe gestionar correctamente el flujo de mensajes y respuestas conversacionales del usuario.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Obtenido***
Reconocimiento del chatbot	*El usuario ingresa a la página principal*	*El botón flotante debe ser visible y representar claramente el acceso al asistente*	*El usuario identifica que puede iniciar una conversación*
Apertura del chatbot	*El usuario selecciona el botón flotante*	*La ventana conversacional debe abrirse sin cubrir información esencial de la página*	*El usuario accede al chatbot sin perder contexto*
Comprensión del campo de texto	*El usuario observa el área de escritura*	*El campo debe presentar una indicación clara sobre qué puede preguntar*	*El usuario entiende que puede escribir consultas académicas o institucionales*
Indicador de espera	*El chatbot procesa la consulta*	*La interfaz debe mostrar un estado de carga o procesamiento*	*El usuario entiende que el sistema está generando una respuesta*
Lectura de respuestas	*El chatbot muestra una respuesta*	*El texto debe presentarse con tamaño, contraste y organización adecuados*	*El usuario puede leer la respuesta sin dificultad*

Continuidad conversacional	*El usuario realiza varias consultas seguidas*	*La interfaz debe conservar los mensajes recientes de forma ordenada*	*El usuario comprende el hilo de la conversación*
Reinicio por inactividad	*El usuario deja de interactuar durante el tiempo configurado*	*El sistema debe limpiar la conversación y mostrar un nuevo mensaje de bienvenida*	*El usuario entiende que se inició una nueva sesión*
Mensaje de error comprensible	*Ocurre un error de conexión o procesamiento*	*El sistema debe mostrar un mensaje claro, sin lenguaje técnico innecesario*	*El usuario comprende que debe intentar nuevamente*
Observaciones: Esta prueba no funcional valida criterios de usabilidad, claridad visual, organización del contenido, continuidad conversacional, manejo de errores de entrada y adaptación de la interfaz.			

Como se observa en la Tabla 16, los casos de prueba ejecutados permitieron validar el funcionamiento general de la interfaz conversacional. Los resultados obtenidos evidenciaron que el sistema permitió enviar consultas, mostrar respuestas dentro del widget, conservar el historial visible durante la sesión activa y acceder al chatbot desde los entornos público y autenticado.

En conjunto, los resultados presentados en la Tabla 16 permitieron confirmar que la interfaz conversacional funcionó correctamente en los escenarios evaluados. Se comprobó que el usuario pudo identificar el botón flotante, abrir el chatbot, escribir consultas, visualizar respuestas, reconocer estados de carga, mantener continuidad en la conversación y comprender los mensajes de error. A partir de estos resultados, se evidenció una integración adecuada entre la capa frontend desarrollada en React y los servicios backend encargados del procesamiento conversacional.

3.2.2.2. Validación de interacción por voz

La validación de interacción por voz tuvo como finalidad comprobar que el usuario pudiera formular consultas habladas y recibir respuestas generadas por el sistema en formato textual y hablado. Esta prueba permitió verificar el funcionamiento del flujo de captura de audio, transcripción automática, procesamiento conversacional y síntesis de respuesta.

Durante la validación se evaluaron consultas institucionales y académicas realizadas mediante audio. En cada caso se verificó que el sistema pudiera interpretar el contenido de la consulta, procesarla mediante los servicios correspondientes y devolver una respuesta coherente al usuario. Los resultados obtenidos se presentan en la Tabla 17.

Tabla 17 Resultados de validación de interacción por voz.

PRUEBA DE USABILIDAD N° 2			
Nombre: Interacción mediante reconocimiento de voz			
Descripción: El sistema debe convertir correctamente comandos de voz en texto para su posterior procesamiento conversacional.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Obtenido***
Identificación del botón de micrófono	*El usuario observa la interfaz del chatbot*	*El botón de micrófono debe ser visible, reconocible y estar ubicado cerca del campo de entrada*	*El usuario identifica fácilmente dónde iniciar una consulta por voz*
Inicio de grabación	*El usuario presiona el botón de micrófono*	*La interfaz debe mostrar una señal clara de que*	*El usuario comprende que el sistema está escuchando*

		la grabación inició	
Finalización de grabación	*El usuario suelta o vuelve a seleccionar el botón de micrófono*	*El sistema debe indicar que el audio fue capturado y está siendo procesado*	*El usuario comprende que debe esperar la respuesta*
Retroalimentación durante procesamiento	*Consulta hablada enviada al sistema*	*La interfaz debe mostrar un mensaje o indicador de carga mientras se procesa la consulta*	*El usuario no percibe que el sistema quedó bloqueado*
Permiso de micrófono denegado	*El navegador bloquea el acceso al micrófono*	*El sistema debe mostrar un mensaje claro indicando que se requiere permiso de micrófono*	*El usuario entiende la causa del problema y puede corregirla*
Audio vacío o no comprensible	*El usuario graba silencio o audio con ruido*	*El sistema debe mostrar un mensaje comprensible solicitando repetir la consulta*	*El usuario sabe qué acción realizar sin abandonar el flujo*
Respuesta por voz	*El usuario recibe una*	*El sistema debe permitir*	*El usuario puede leer y escuchar la respuesta*

	respuesta generada	*reproducir la respuesta hablada sin impedir la lectura del texto*	
Observaciones: Esta prueba no funcional evalúa accesibilidad, facilidad de interacción, conversión de voz a texto, manejo de errores de micrófono y continuidad del flujo conversacional.			

Como se observa en la Tabla 17, las pruebas ejecutadas permitieron comprobar el funcionamiento del flujo de interacción por voz. Los resultados obtenidos evidenciaron que el sistema transcribió las consultas habladas, procesó el contenido mediante el motor conversacional y generó respuestas coherentes según el tipo de consulta realizada.

En conjunto, los resultados presentados en la Tabla 17 permiten confirmar que la interacción por voz funcionó como un mecanismo complementario al ingreso textual. Esta funcionalidad aportó una alternativa adicional de uso para el estudiante, manteniendo la integración con las consultas institucionales mediante RAG y con las consultas académicas personalizadas cuando existió una sesión autenticada.

3.2.2.3. Validación de seguridad y registros de eventos

La validación de seguridad y registros de eventos tuvo como finalidad comprobar que el sistema protegiera el acceso a la información académica personalizada y registrara eventos relevantes durante el uso del prototipo. Estas pruebas se enfocaron en verificar el inicio de sesión, la autenticación mediante OTP, la restricción de acceso a recursos protegidos, el control de roles y la generación de registros de eventos.

Durante esta validación se consideraron escenarios relacionados con credenciales correctas e incorrectas, verificación de segundo factor, bloqueo de acceso no autorizado, control de recursos protegidos y registro de eventos del sistema. Los resultados obtenidos se presentan en la Tabla 18.

Tabla 18 Resultados de validación de seguridad y registros de eventos.

PRUEBA DE SEGURIDAD N° 1

Nombre: Seguridad, autenticación y registros de eventos			
Descripción: El sistema debe garantizar el acceso seguro mediante autenticación JWT, restringir el acceso a funcionalidades protegidas y registrar los eventos relevantes ejecutados por los usuarios para fines de trazabilidad y monitoreo técnico.			
Escenarios			
Escenario evaluado	***Campo de Entrada***	***Criterio de aceptación***	***Resultado Obtenido***
Inicio de sesión válido	*Usuario/correo/código estudiantil y contraseña válida.*	*El sistema debe validar las credenciales y continuar con la verificación OTP*	*El usuario accede al segundo factor de autenticación*
Verificación mediante OTP	*Código OTP temporal de seis dígitos.*	*El sistema debe validar el código OTP y emitir los tokens JWT de sesión.*	*El usuario accede al dashboard académico*
Rechazo de credenciales inválidas	*Usuario o contraseña incorrectos.*	*El sistema debe rechazar el acceso y no generar sesión.*	*El usuario permanece fuera del sistema*
Registro de inicio de sesión	*Login válido completado con OTP.*	*El sistema debe registrar el evento de inicio de sesión.*	*El evento queda almacenado en los registros del sistema*

Registro de consulta al chatbot	*Consulta enviada al chatbot.*	*El sistema debe registrar la interacción como evento trazable dentro de los registros del sistema.*	*La consulta queda registrada para trazabilidad*
Registro de consulta por voz	*Audio enviado desde el micrófono del chatbot.*	*El sistema debe registrar la consulta por voz como evento trazable dentro de los registros del sistema.*	*La interacción por voz queda registrada*
Registro de cierre de sesión	*Acción de cerrar sesión.*	*El sistema debe invalidar la sesión activa y registrar el evento de cierre.*	*El cierre queda registrado correctamente*
Observaciones: Esta prueba no funcional valida autenticación, autorización, protección de recursos, control de acceso por rol, manejo de sesión y trazabilidad de eventos dentro del sistema.			

Como se observa en la Tabla 18, las pruebas ejecutadas permitieron comprobar que los mecanismos de seguridad implementados protegieron el acceso a la información académica personalizada. Los resultados obtenidos evidenciaron que el sistema validó credenciales, aplicó verificación OTP, restringió recursos protegidos y controló el acceso según el rol del usuario.

En conjunto, los resultados presentados en la Tabla 18 permiten confirmar que el registro de eventos funcionó como mecanismo de trazabilidad técnica del prototipo. Los eventos generados durante autenticación, consultas, accesos protegidos y cierre de sesión quedaron registrados, lo que permitió verificar el comportamiento del sistema ante acciones relevantes de seguridad y uso.

Criterios de aceptación de las historias de usuario

Una vez finalizada cada historia de usuario, se verificó el cumplimiento de los criterios de aceptación definidos en el *Product Backlog*. Esta validación permitió comprobar que el sistema respondiera adecuadamente a las consultas académicas personalizadas, consultas institucionales, interacción por voz, uso de la interfaz conversacional, autenticación segura y generación de registros de eventos. Los resultados evidenciaron que los criterios funcionales fueron cumplidos, debido a que el chatbot logró recuperar información desde PostgreSQL, utilizar la arquitectura RAG para consultas institucionales, procesar entradas por voz, restringir el acceso a información personalizada sin autenticación y registrar eventos relevantes dentro del sistema.

Para medir los tiempos de respuesta asociados a las historias de usuario de consulta, se realizaron 10 pruebas por modalidad. En el chatbot institucional, las consultas por texto registraron un tiempo promedio de 4.875,40 ms, equivalente a 4,88 segundos, por lo que cumplieron el criterio máximo establecido de 5 segundos. Sin embargo, las consultas institucionales por voz alcanzaron un promedio de 6.227,80 ms, equivalente a 6,23 segundos, superando el tiempo esperado. En el chatbot personalizado, cuyo criterio máximo era de 3 segundos, las consultas por texto y por voz también superaron el límite definido. Por tanto, el sistema cumplió funcionalmente con la entrega de información institucional y académica personalizada, pero presentó observaciones relacionadas con la eficiencia del tiempo de respuesta.

Este comportamiento se relacionó con la intervención de varios componentes técnicos durante el procesamiento de cada consulta, entre ellos la latencia de conexión a Internet, el tiempo de respuesta de los servicios externos de inteligencia artificial, la recuperación de información desde PostgreSQL o desde el índice documental, la construcción del contexto para el modelo de lenguaje y el procesamiento realizado por el *backend*. En la modalidad por voz, el tiempo fue mayor debido a que el flujo incorporó etapas adicionales de captura de audio, transcripción, procesamiento conversacional y reproducción de la

respuesta. Por esta razón, las historias de usuario fueron aceptadas funcionalmente, pero con observaciones orientadas a la optimización de los tiempos de respuesta.

CONCLUSIONES

El desarrollo del prototipo de chatbot académico personalizado permitió cumplir con el objetivo general de la investigación, al integrar en una sola solución consultas institucionales de carácter público y consultas académicas personalizadas para estudiantes de primer nivel de la carrera de Ingeniería de la PUCE-I. La propuesta permitió centralizar el acceso a información relacionada con horarios, asignaturas, aulas, calificaciones, actividades pendientes y orientación general, fortaleciendo la atención académica inicial dentro de un entorno controlado.

La implementación de una base de conocimiento institucional mediante arquitectura *Retrieval-Augmented Generation* permitió responder consultas generales utilizando información previamente organizada y recuperada como contexto antes de generar una respuesta. Este enfoque contribuyó a mejorar la pertinencia de las respuestas institucionales y a reducir el riesgo de generar información alejada del contexto universitario. No obstante, se identificó que la calidad de las respuestas depende directamente de la amplitud, vigencia y cobertura de los documentos incorporados a la base de conocimiento.

La base de datos académica implementada en PostgreSQL permitió gestionar información simulada de estudiantes de primer nivel, lo que facilitó la validación técnica de consultas personalizadas sin utilizar datos académicos reales. A partir de esta estructura, el chatbot pudo entregar respuestas asociadas al perfil del estudiante autenticado, diferenciando adecuadamente la información pública de la información privada.

La integración del *backend* desarrollado en Django REST *Framework* con la interfaz web construida en React permitió establecer una arquitectura modular, organizada mediante servicios REST y orientada a la comunicación entre los distintos componentes del sistema. Esta estructura facilitó la interacción del usuario con el chatbot institucional, el *dashboard* estudiantil, los servicios académicos, la recuperación de información y los mecanismos de seguridad.

La incorporación del reconocimiento de voz funcionó como un mecanismo complementario de interacción, permitiendo que el estudiante realizara consultas habladas dentro del asistente conversacional. Esta funcionalidad amplió las formas de acceso al sistema sin reemplazar la entrada por texto, aportando una experiencia más flexible para el usuario. Sin embargo, su desempeño puede verse condicionado por

factores externos como ruido ambiental, claridad de pronunciación y permisos del navegador.

Los mecanismos de seguridad implementados, como autenticación, verificación OTP, control de acceso y registro de eventos, permitieron fortalecer la protección de la información académica personalizada y mejorar la trazabilidad de las acciones realizadas dentro del sistema. Esto resultó especialmente importante debido a que el prototipo diferencia entre visitantes públicos y estudiantes autenticados.

La validación del prototipo permitió comprobar el funcionamiento de los componentes principales del sistema mediante pruebas funcionales y no funcionales aplicadas en un entorno controlado. Los resultados evidenciaron que la solución propuesta es técnicamente viable para apoyar la orientación académica de estudiantes de primer nivel; sin embargo, también se identificaron observaciones relacionadas con la optimización de tiempos de respuesta, la necesidad de ampliar la cobertura de la base de conocimiento institucional y el fortalecimiento de evidencias de validación. Debido a que se utilizaron datos simulados y no se realizó un despliegue productivo institucional, los resultados corresponden a una validación técnica inicial del prototipo.

RECOMENDACIONES

- Realizar una prueba piloto con estudiantes reales del primer nivel de la carrera de Ingeniería, previa autorización institucional, con el fin de evaluar el comportamiento del chatbot en un contexto de uso más cercano al entorno académico real. Esta prueba permitiría identificar mejoras en la experiencia de usuario, la claridad de las respuestas, los tiempos de interacción y la aceptación del sistema.
- Se recomienda mantener actualizada y ampliar progresivamente la base de conocimiento institucional utilizada por la arquitectura RAG, incorporando documentos oficiales vigentes, reglamentos, procedimientos académicos, calendarios, servicios universitarios y contenidos relacionados con otras áreas de interés para los estudiantes, como bienestar estudiantil, pastoral universitaria, identidad católica institucional, becas, actividades culturales, deportivas y servicios de acompañamiento. Esta ampliación permitiría que el chatbot responda consultas que actualmente podrían quedar fuera de la cobertura documental, mejorando la pertinencia, confiabilidad y utilidad de las respuestas generadas dentro del contexto universitario.
- Optimizar los tiempos de respuesta del sistema, debido a que las historias de usuario fueron aceptadas funcionalmente, pero presentaron observaciones relacionadas con la eficiencia. Esta mejora debería enfocarse en los componentes que influyeron en la latencia registrada, como la conexión a Internet, los servicios externos de inteligencia artificial, la recuperación de información desde PostgreSQL, la consulta al índice documental RAG, el procesamiento del *backend* y el flujo de voz. Para ello, podrían aplicarse estrategias como almacenamiento en caché para consultas frecuentes, optimización de consultas a la base de datos, ajuste de los fragmentos recuperados en RAG, monitoreo de latencia por componente y mejora de los procesos de transcripción y reproducción de audio.
- Fortalecer los mecanismos de seguridad antes de una posible implementación institucional, considerando políticas de expiración de *tokens*, control de intentos fallidos, gestión segura de credenciales, protección de *endpoints*, respaldo de información y revisión periódica de los registros de eventos. Estas medidas

permitirían mejorar la confidencialidad, integridad y trazabilidad de la información académica.

- Ampliar progresivamente el alcance del prototipo para atender a estudiantes de niveles superiores y/o a otras carreras. Este escalamiento se deberá realizar después de validar su funcionamiento con el grupo objetivo inicial.
- Complementar la validación técnica con instrumentos de evaluación de satisfacción y usabilidad aplicados a estudiantes, con el fin de conocer su percepción sobre la utilidad, facilidad de uso, claridad de respuestas e interacción por voz. Esto permitiría obtener evidencia más completa sobre el impacto del chatbot en la orientación académica inicial.
- Documentar los procedimientos de administración, actualización de información y mantenimiento del índice RAG, con el propósito de facilitar la sostenibilidad del sistema en caso de que sea retomado, ampliado o implementado en un entorno institucional.

REFERENCIAS BIBLIOGRÁFICAS

Bibliografía

Belov, R. (11 de September de 2023). *Integrative models of client-server technology interaction in web development*. Obtenido de Zenodo: https://zenodo.org/records/15035220

CAO, H. (19 de June de 2024). *Arxiv*. Obtenido de Recent advances in universal text embeddings: A Comprehensive Review of Top-Performing Methods on the MTEB Benchmark: https://arxiv.org/html/2406.01607v2

Cormac McGrath., A. F.-P. (24 de August de 2024). *Springer Nature Link*. Obtenido de Generative AI chatbots in higher education: a review of an emerging research area: https://link.springer.com/article/10.1007/s10734-024-01288-w

DeBellis, M. D. (01 de February de 2025). *Integrating Ontologies and Large Language Models to Implement Retrieval Augmented Generation*. Obtenido de ResearchGate: https://www.researchgate.net/figure/Standard-RAG-architecture-RAG-retrieval-augmented-generation_fig3_388593710

Galina Ilieva1., T. Y.-B. (6 de September de 2023). *MDPI*. Obtenido de Effects of Generative Chatbots in Higher Education: https://www.mdpi.com/2078-2489/14/9/492

Labadze, L. G. (31 de October de 2023). *Springer Nature Link*. Obtenido de Role of AI chatbots in education: systematic literature review: https://link.springer.com/article/10.1186/s41239-023-00426-1

Lambebo, E., & Chen, H.-L. (06 de December de 2024). Chatbots in higher education: A systematic review. *Interactive Learning Environments, 33*(4), 2781-2807. doi:10.1080/10494820.2024.2436931

Matthijs Douze, A. G.-E. (16 de January de 2024). *arXiv*. Obtenido de The Faiss library: https://arxiv.org/abs/2401.08281

Pedro Filipe Oliveira., P. M. (11 de December de 2023). *MDPI*. Obtenido de Introducing a Chatbot to the Web Portal of a Higher Education Institution to Enhance Student Interaction †: https://www.mdpi.com/2673-4591/56/1/128

Pranab Sahoo, A. K. (5 de February de 2024). *Cornell University*. Obtenido de A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications: https://arxiv.org/abs/2402.07927

Wayne Xin Zhao, K. Z. (31 de March de 2023). Obtenido de A Survey of Large Language Models: https://arxiv.org/abs/2303.18223

Yildiz Durak, H., & Onan, A. (21 de May de 2024). Predicting the use of chatbot systems in education: A comparative approach using PLS-SEM and machine learning algorithms. *Current Psychology, 43*, 23656-23674. doi:10.1007/s12144-024-06072-8

Yunfan Gao, Y. X. (18 de December de 2023). *Cornell University*. Obtenido de Retrieval-Augmented Generation for Large Language Models: A Survey: https://arxiv.org/abs/2312.10997

Zubiaga, A. (12 de January de 2024). *National Library of Medicine*. Obtenido de Natural language processing in the era of large language models: https://pmc.ncbi.nlm.nih.gov/articles/PMC10820986/

ANEXOS

Anexo 1 Carta de aceptación

A quien pueda Interesar

Por medio de la presente, yo, Dulce Milagro Rivero como Líder del Grupo de Investigación en Sistemas Inteligentes (GISI) confirmo que he revisado el Trabajo de Titulación titulado "**CHATBOT PERSONALIZADO MEDIANTE EL USO DE VOZ PARA LA ATENCIÓN GENERAL DE ESTUDIANTES DE PRIMER NIVEL EN LA CARRERA DE INGENIERÍA EN LA PUCE-I**", elaborado por el estudiante **Nick Bryan López Reina** como requisito previo para la obtención del título de Ingeniero en Tecnologías de la Información en la PUCE-SI.

Una vez revisado el trabajo de grado, se ha verificado que el sistema cumple satisfactoriamente con los objetivos establecidos en el proyecto.

Sin otro particular,

Quedo atenta a cualquier consulta adicional.

Dulce Milagro Rivero Albarrán
Firmado digitalmente por Dulce Milagro Rivero Albarrán
Fecha: 2026.07.22 08:45:52 -05'00'

Dra. Dulce Milagro Rivero
Líder del Grupo de Investigación GISI
CI. 1757608961