Trabajo de Grado en la Modalidad Tesis para Optar por el Título de Ingeniero en Sistemas

Desarrollo e Implementación de un Chatbot con Inteligencia artificial, en el Instituto Tecnológico del Putumayo, para la extracción y respuesta automática a partir de documentos.

Estudiantes:

Yan Carlos Cuaran Imbacuan y Maicol Libardo Jossa Campaña

Facultad de Ingeniería y Ciencias Básicas, Instituto Tecnológico del Putumayo

54211: Ingeniería de Sistemas

Asesor:

Jhoni Ricardo Cerón Chaves

6 de junio de 2024

Resumen

El presente trabajo tiene como objetivo desarrollar e implementar un chatbot con inteligencia artificial en el Instituto Tecnológico del Putumayo (ITP) para la extracción y respuesta automática de información a partir de documentos. Utilizando la metodología A.D.D.I.E. para el desarrollo y SCRUM para la gestión del proyecto, se ha diseñado un chatbot capaz de manejar consultas de forma precisa y eficiente, mejorando la accesibilidad a la información pública de la institución. El chatbot se integra en la página principal del ITP, facilitando la obtención de respuestas a preguntas frecuentes relacionadas con los documentos institucionales como los estatutos, con el fin de reducir la carga de trabajo del personal administrativo y promoviendo la autonomía de la comunidad académica. Como resultado, Se espera que el resultado final sea una experiencia de usuario más eficiente y satisfactoria en el entorno educativo del ITP, demostrando la utilidad y efectividad de los chatbots con inteligencia artificial en la gestión de información y apoyo a la comunidad educativa.

Palabras Clave: Chatbot, Inteligencia Artificial, Langchain, LLM, RAG, A.D.D.I.E, SCRUM, Procesamiento de Documentos, Tecnología, Educación.

Abstract

The objective of this work is to develop and implement an artificial intelligence chatbot at the Instituto Tecnológico del Putumayo (ITP) for the automatic extraction and response of information from documents. Using the A.D.D.I.E. methodology for development and SCRUM for project management, a chatbot capable of handling queries accurately and efficiently has been designed, improving accessibility to the institution's public information. The chatbot is integrated into the ITP main page, facilitating the obtaining of answers to frequently asked questions related to institutional documents such as statutes, in order to reduce the workload of administrative staff and promoting the autonomy of the academic community. As a result, the final result is expected to be a more efficient and satisfactory user experience in the ITP educational environment, demonstrating the usefulness and effectiveness of artificial intelligence chatbots in information management and support to the educational community.

Keywords: Chatbot, Artificial Intelligence, Langchain, LLM, RAG, A.D.D.I.E., SCRUM, Document Processing, Technology, Education.

Índice General

Capítulo 1: Información General *11*

1.1. Introducción *11*

1.2. Estructura del Trabajo *13*

1.3. Antecedentes *14*

1.4. Justificación *15*

1.5. Planteamiento del Problema *16*

1.6. Objetivos *17*

1.6.1. Objetivo General *17*

1.6.2. Objetivos Específicos *17*

Capítulo 2: Estado del Arte *17*

2.1. Marco Teórico *17*

2.1.1. Inteligencia Artificial (IA) *17*

2.1.1. Clasificación de la IA *18*

2.1.2. Chatbot *18*

2.1.3. Procesamiento del Lenguaje Natural (PNL) *19*

2.2. Marco Conceptual *19*

2.2.1. Aplicaciones de la IA *19*

2.2.2. IA en la educación *19*

2.2.3. PLN en la Educación *19*

2.2.4. Metodología A.D.D.I.E *19*

2.2.5. 20

2.2.6. Gestión de la Información en Instituciones Educativas *20*

2.2.7. OpenAI ChatGPT y APIs 20

2.2.8. Python como Herramienta para el Desarrollo de IA 21

2.2.9. Streamlit 21

2.2.10. Mongo DB 22

2.2.11. Elasticsearch 22

2.3. Marco Legal 22

2.4. Marco Contextual 24

Capítulo 3: Metodología 25

3.1. Enfoque y Tipo de Investigación 25

3.2. Etapa de Análisis 26

3.2.1. Sprint 1: Configuración Inicial y Desarrollo del Backend 33

3.2.2. Sprint 2: Desarrollo del Frontend y Diseño de la UI 34

3.2.3. Sprint 3: Integración de APIs de IA y Procesamiento de Lenguaje Natural 35

3.2.4. Sprint 4: Pruebas y Optimización 36

3.2.5. Sprint 5: Pruebas de Usabilidad y Preparación para Despliegue 37

3.3. Etapa de Diseño 38

3.4. Etapa de desarrollo 46

3.4.1. Herramientas de desarrollo 46

3.4.2. Desarrollo del Backend 47

3.4.3. Desarrollo del Frontend 54

3.5. Etapa de Implementación 60

3.6. Etapa de Evaluación 66

Capítulo 4: Conclusiones y Trabajo Futuro *78*

Conclusiones *78*

Trabajo Futuro *79*

Capítulo 5: Gestión del Proyecto *80*

Presupuesto *80*

Planificación del Proyecto *82*

APÉNDICES 87

Apéndice A. Autorización de Implementación – En el ITP *87*

ANEXOS 88

Anexo 1. Formulario de Usabilidad del Chatbot *88*

Anexo 2. Resultados Revisión Sistemática *92*

Anexo 3. Investigacion modelos LLM open Source y limitaciones *101*

Anexo 4. Manual de usuario. *106*

Anexo 5. Manual de Programador *119*

Índice de Tablas

Tabla 1 *OpenAI modelos calidad - precio* *31*

Tabla 2 *Roles de los miembros del equipo* *32*

Tabla 3 *Requisitos Funcionales y No Funcionales* *38*

Tabla 4 *Recursos del Programa* *61*

Tabla 5 *Especificaciones de la Aplicación* *63*

Tabla 6 *Recursos y Presupuesto* *81*

Tabla 7 *Cronograma de actividades* *82*

Índice de Figuras

Figura 1 *Etapas modelo A.D.D.I.E* *25*

Figura 2 *Metodología SCRUM* *26*

Figura 3 *OpenAI API* *31*

Figura 4 *Diagrama de Caso de Uso – Funcionamiento General* *40*

Figura 5 *Diagrama de Caso de Uso – Modos del Chatbot* *41*

Figura 6 *Diagrama de Caso de Uso – Secuencia de funcionamiento* *43*

Figura 7 *Interfaz principal* *44*

Figura 8 *interfaz ChatPDF* *45*

Figura 9 *Plantillas Streamlit* *45*

Figura 10 *Plantillas Streamlit* *46*

Figura 11 *Creación del proyecto en Visual Code* *48*

Figura 12 *Creación y configuración de Mongo DB* *48*

Figura 13 *Estructura de la base de datos.* *50*

Figura 14 *Configuración Mongo db con el backend* *50*

Figura 15 *Instalación de Elasticsearch* *51*

Figura 16 *Configuración Elasticsearch con el backend* *51*

Figura 17 *Generación de Api Key* *52*

Figura 18 *Integración de Api Key en el backend* *52*

Figura 19 *Integración del modelo de IA* *53*

Figura 20 *Interfaz Principal del chatbot* *55*

Figura 21 *Categorías preguntas frecuentes (Estatuto 2)* **¡Error! Marcador no definido.**

Figura 22 *Categorías preguntas frecuentes (Estatuto 1)* **¡Error! Marcador no definido.**

Figura 23 *Interfaz principal, Realizar una solicitud* *57*

Figura 24 *Interfaz principal, modo avanzado* *57*

Figura 25 *Interfaz principal, barra lateral* *58*

Figura 26 *Interfaz Principal, Guía chatbot* *59*

Figura 27 *Interfaz Secciones, configuración* *59*

Figura 28 *Interfaz Secciones, Configuración Administrador* *60*

Figura 29 *Diagrama de clases* *62*

Figura 30 *Configuración VPN* *64*

Figura 31 *Características de la Máquina Virtual* *64*

Figura 32 *Alojamiento del Código* *65*

Figura 33 *Ejecución del Chatbot en la Máquina Virtual* *65*

Figura 34 *Participantes 8vo semestre* *67*

Figura 35 *Participantes 10mo semestre* *68*

Figura 36 *Pregunta - ¿Carrera a la que pertenece?* *69*

Figura 37 *Pregunta - ¿Semestre al cual pertenece?* *69*

Figura 38 *Pregunta - ¿Ha tenido interacción con algún tipo de Chatbot o herramienta de respuesta automatizada?* *70*

Figura 39 *Pregunta - Si en la pregunta anterior su respuesta fue sí, por favor, cuéntenos como ha sido su experiencia con chatbots* *70*

Figura 40 *Pregunta - ¿Considera que esta herramienta optimiza el trabajo a los administrativos?* *71*

Figura 41 *Pregunta - ¿Considera que esta herramienta sería de gran utilidad para nuestros usuarios??* *71*

Figura 42 *Pregunta - ¿Cuál cree que sería el nivel de uso del Chatbot?* *72*

Figura 43 *Pregunta - ¿Cuál cree que sería el nivel de uso del Chatbot?* *72*

Figura 44 *Pregunta - ¿Qué tan fácil fue para ti encontrar la información que buscabas utilizando el Chatbot?* *73*

Figura 45 *Pregunta - ¿El Chatbot resolvió tus dudas o consultas de manera eficiente?* *74*

Figura 46 *Pregunta - ¿El diseño y la interfaz del Chatbot son intuitivos y fáciles de usar?* *74*

Figura 47 *Pregunta - ¿Consideras que los tiempos de respuesta del Chatbot son adecuados?* *75*

Figura 48 *Pregunta - ¿Tuviste algún problema técnico al utilizar el Chatbot? (Por ejemplo, mensajes de error, respuestas incorrectas, etc.)* *76*

Figura 49 *Pregunta - ¿Recomendarías el uso del Chatbot a otros estudiantes del ITP?* *77*

Figura 50 *Pregunta - ¿Tienes alguna sugerencia específica para mejorar la funcionalidad, la usabilidad o el contenido del Chatbot?* *77*

Capítulo 1: Información General

1.1. Introducción

En la actualidad, los sistemas de Chatbots con Inteligencia Artificial (IA) han surgido como herramientas innovadoras para interactuar con los usuarios de manera eficiente e incluso se ha considerado como una de las tareas esenciales en inteligencia artificial y lenguaje natural modelado el modelado de conversaciones (Ayanouz et al., 2020). Estos sistemas han sido integrados en diversas plataformas, como aplicaciones de mensajería y sitios web, proporcionando respuestas instantáneas y personalizadas (Garibay, 2020). La evolución de los chatbots no solo se ha visto impulsada por los avances en IA, sino también por su capacidad para simular conversaciones coherentes y atender solicitudes de información de manera autónoma, lo que ha incrementado su popularidad en diferentes ámbitos, incluido el educativo (Álvarez et al., 2018).

En el Instituto Tecnológico del Putumayo (ITP), se ha identificado una problemática significativa en la gestión y acceso a la información relevante para la comunidad educativa. Estudiantes, docentes y personal administrativo enfrentan dificultades para encontrar información específica debido a la dispersión de la documentación en múltiples fuentes. Además, la dependencia de los horarios de atención del personal administrativo limita la disponibilidad de esta información, lo que puede resultar en una sobrecarga de trabajo (Peña-Torres et al., 2021).

Para abordar esta problemática, se ha desarrollado e implementado un chatbot con inteligencia artificial en el ITP. Este proyecto tiene como objetivo principal mejorar la accesibilidad a la información pública mediante la extracción y respuesta automática a partir de documentos. Utilizando la metodología A.D.D.I.E. para un desarrollo sistemático y SCRUM para una gestión ágil del proyecto, se diseñó un chatbot capaz de manejar consultas de manera precisa y eficiente. El chatbot se integra en la página principal del ITP, facilitando la obtención de respuestas inmediatas a preguntas frecuentes relacionadas con documentos

institucionales, como los estatutos, con el fin de reducir la carga de trabajo del personal administrativo y promover la autonomía de la comunidad académica.

El desarrollo de este chatbot incluyó un enfoque mixto (cuantitativo y cualitativo), evaluando su impacto en la comunidad educativa del ITP en términos de usabilidad y satisfacción. Se espera que esta herramienta no solo mejore la accesibilidad a la información, sino que también contribuya a una experiencia de usuario más eficiente y satisfactoria, demostrando la utilidad y efectividad de los chatbots con inteligencia artificial en la gestión de información educativa.

Este proyecto representa un avance significativo en la integración de tecnologías emergentes en el sector educativo, promoviendo la autonomía estudiantil y optimizando los recursos administrativos. A través de la implementación de este chatbot, el ITP busca mejorar la calidad de los servicios educativos ofrecidos y fomentar una cultura de innovación tecnológica.

1.2. Estructura del Trabajo

En el Capítulo 1 se aborda la **Información General** (antecedentes, explicación de la problemática y pregunta de investigación, por último, los objetivos propuestos)

En el Capítulo 2 se aborda el **Estado del Arte** (contexto y profundización sobre los chatbots e inteligencia artificial).

En el Capítulo 3 se aborda la **Metodología** (tipo de investigación, enfoque, población objetivo, e iteración por las etapas del modelo en cascada modificado).

En el Capítulo 4 se aborda la **Conclusiones y Trabajo Futuro** (conclusiones finales del trabajo realizado y se indican posibles mejoras sobre las funcionalidades actuales y nuevas líneas de trabajo para mejora del chatbot)

En el Capítulo 5 se aborda la **Gestión del Proyecto** (recursos usados y presupuesto, cronograma de trabajo).

1.3. Antecedentes

La inteligencia artificial (IA) y el procesamiento del lenguaje natural han transformado la manera en que interactuamos con la tecnología, permitiendo el desarrollo de chatbots que proporcionan respuestas inmediatas y personalizadas. Estos avances han sido particularmente relevantes en el ámbito educativo, donde los chatbots pueden mejorar significativamente el acceso a la información y apoyar el aprendizaje de los estudiantes (Álvarez et al., 2018; Garibay, 2020).

Investigaciones previas han demostrado la efectividad de los chatbots en diversos contextos educativos. (Múnera Torres et al., 2022) destacaron cómo los asistentes virtuales pueden proporcionar información relevante y facilitar el acceso a recursos educativos, fomentando la autonomía de los estudiantes. De manera similar, (Peña-Torres et al., 2021) identificaron que la dispersión de la documentación y la necesidad de interactuar con múltiples fuentes para obtener respuestas representan obstáculos significativos, causando sobrecarga en el personal administrativo y académico.

En el Instituto Tecnológico del Putumayo (ITP), estas dificultades han sido evidentes, subrayando la necesidad de soluciones innovadoras para mejorar la gestión de la información. La implementación de un chatbot con IA en el ITP se enmarca en esta tendencia de utilizar tecnología avanzada para resolver problemas de accesibilidad a la información, optimizando así los recursos disponibles y promoviendo una mayor autonomía entre los miembros de la comunidad académica.

Estudios como los de (Garibay, 2020; Múnera Torres et al., 2022) han proporcionado una base sólida para entender cómo los chatbots pueden integrarse en el entorno educativo, demostrando su capacidad para mejorar la eficiencia y la satisfacción de los usuarios. La experiencia en otras instituciones educativas ha mostrado que los chatbots pueden no solo facilitar el acceso a la información, sino también transformar la interacción educativa, haciendo que el aprendizaje sea más accesible y personalizado.

1.4. Justificación

La implementación de un chatbot con inteligencia artificial en el Instituto Tecnológico del Putumayo (ITP) responde a una necesidad de modernizar y optimizar el acceso a la información en la institución. Actualmente, el acceso a documentos relevantes se ve limitado por la dispersión de la información y la dependencia de horarios específicos para la atención, lo cual no solo limita la autonomía de los estudiantes, sino que también sobrecarga al personal administrativo, afectando la eficiencia operativa y la calidad del servicio educativo (Moral-Sánchez et al., 2023).

El uso de chatbots en el ámbito educativo ha demostrado ser una herramienta efectiva para proporcionar soporte continuo y personalizado, facilitando el aprendizaje y mejorando la satisfacción del usuario. Según estudios recientes, los chatbots pueden ofrecer respuestas inmediatas y precisas, lo que se traduce en una experiencia de usuario más eficiente y satisfactoria (Henriquez Miranda et al., 2024). Además, la inteligencia artificial ha demostrado ser capaz de manejar consultas complejas, proporcionando información relevante y oportuna que mejora la comprensión de los conceptos por parte de los estudiantes (Ocaña-Fernandez et al., 2019).

En el contexto del ITP, la implementación de un chatbot con IA no solo mejorará el acceso a la información, sino que también fomentará una cultura de innovación y adaptación tecnológica dentro de la institución. Este proyecto está alineado con las tendencias globales de digitalización y modernización del sector educativo, destacando la importancia de integrar tecnologías avanzadas para enfrentar los desafíos educativos actuales y futuros (Moral-Sánchez et al., 2023).

El chatbot, diseñado para responder automáticamente a preguntas frecuentes y extraer información de documentos institucionales, proporcionará un acceso más rápido y eficiente a la información, reduciendo la carga de trabajo del personal administrativo y promoviendo la autonomía de la comunidad académica. Este enfoque permitirá a los estudiantes y al personal

acceder a la información relevante de manera rápida, mejorando la experiencia educativa y administrativa en el ITP (Henriquez Miranda et al., 2024).

1.5. Planteamiento del Problema

El tema de interés de la presente investigación es la falta de acceso eficiente y rápido a la información relevante por parte de la comunidad educativa en el contexto de la institución. En la actualidad, los estudiantes, docentes y administrativos a menudo enfrentan dificultades al tratar de encontrar información específica como los servicios que se ofrecen, las políticas y reglamentos institucionales, los requisitos de graduación, entre otros. Este problema surge de la dispersión de la información en múltiples fuentes, lo que dificulta su búsqueda.

Además, se identifica como una problemática relacionada con la sobrecarga de trabajo que cae sobre el personal administrativo y académico para responder preguntas comunes. Esto no sólo consume tiempo y recursos valiosos que podrían destinarse a otras tareas, sino que también limita la autonomía de los estudiantes al hacerlos dependientes del horario de atención de las oficinas administrativas.

Por lo tanto, el objetivo de esta investigación es explorar el uso de chatbot con inteligencia Artificial como una solución potencial a estos problemas. Este podría permitir a la comunidad educativa buscar y obtener respuestas rápidas y precisas a través de documentos sin tener que consultar a diferentes personas u otros medios. Esto resultara en una mayor autonomía estudiantil, un ahorro de tiempo, recursos, y una mejora en la experiencia general de la comunidad en conocer sobre él instituto.

A raíz de esta situación, surge la siguiente pregunta de investigación: ¿Cómo puede la implementación de chatbot en el ITP mejorar el acceso a la información relevante para la comunidad educativa y contribuir a sus conocimientos y necesidades?

1.6. Objetivos

1.6.1. Objetivo General

Diseñar, desarrollar e implementar un chatbot con inteligencia artificial que facilite el acceso a la información relevante del instituto tecnológico, mediante un sistema de preguntas de manera escrita y verbal, con el fin de mejorar la experiencia de la comunidad académica y fomentar la autonomía.

1.6.2. Objetivos Específicos

- Realizar una revisión de literatura sobre la implementación de chatbots con inteligencia artificial, específicamente en el manejo de documentos, con el fin de comprender las mejores prácticas, desafíos y beneficios asociados a su uso.
- Diseñar un chatbot que sea capaz de manejar consultas de forma precisa y eficiente.
- Desarrollar e implementar el chatbot en el entorno de producción, asegurando su correcta integración, funcionamiento y compatibilidad con los sistemas actuales de la página principal del instituto tecnológico del putumayo.
- Evaluar la eficiencia del chatbot en la provisión de información relevante a la comunidad educativa, y cómo facilita su acceso a la misma.

Capítulo 2: Estado del Arte

2.1. Marco Teórico

2.2.1. Inteligencia Artificial (IA)

La inteligencia artificial (IA) se refiere a la capacidad de las máquinas para realizar tareas que requieren inteligencia humana, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones, también la han considerado como una disciplina de la informática que tiene por objetivo elaborar máquinas y sistemas que puedan desempeñar tareas que requieren una inteligencia humana. Desde sus inicios en la década de 1950, la IA ha

evolucionado significativamente, integrándose en diversas aplicaciones que van desde el procesamiento del lenguaje natural hasta la conducción autónoma (Abeliuk & Gutiérrez, 2021; Estupiñán Ricardo et al., 2021).

2.1.1. Clasificación de la IA

La IA se clasifica de la siguiente manera:

Inteligencia artificial débil: considera que los ordenadores únicamente pueden simular que razonan y solamente pueden actuar de forma inteligente. Los partidarios de la inteligencia artificial débil consideran que no será nunca posible construir ordenadores conscientes y que un programa es una simulación de un proceso cognitivo, pero no un proceso cognitivo en sí mismo.

Inteligencia artificial fuerte: especifica que un ordenador puede tener una mente y estados mentales y que, por tanto, un día será posible construir uno con todas las capacidades de la mente humana. Este ordenador será capaz de razonar, imaginar, etcétera (Vidal Ledo et al., 2019).

2.1.2. Chatbot

Los chatbots son programas de software diseñados para simular conversaciones humanas a través de interfaces de chat. Existen chatbots basados en reglas y chatbots basados en IA, estos últimos utilizan técnicas avanzadas de procesamiento del lenguaje natural para comprender y generar respuestas más precisas y contextuales (Garibay, 2020).

Desde el primer chatbot ELIZA desarrollado por Joseph Weizenbaum en 1966, los chatbots han avanzado considerablemente. Hoy en día, tecnologías como el aprendizaje profundo y el PLN han permitido desarrollar chatbots mucho más sofisticados y útiles en una variedad de contextos, incluyendo la educación y el servicio al cliente (Romero et al., 2020).

Los chatbots se utilizan en numerosos sectores, como el servicio al cliente, marketing y educación, proporcionando una herramienta eficaz para la interacción y el soporte continuo (Álvarez et al., 2018).

2.1.3. Procesamiento del Lenguaje Natural (PNL)

El procesamiento del lenguaje natural (PLN) es una rama de la IA que se enfoca en la interacción entre las computadoras y el lenguaje humano. Incluye componentes como la sintaxis, la semántica, la pragmática y el discurso (Moral-Sánchez et al., 2023).

Entre las técnicas de PLN utilizadas en chatbots se encuentran el análisis de sentimientos, el reconocimiento de entidades nombradas (NER) y la generación de lenguaje natural (NLG). Estas técnicas permiten a los chatbots entender y responder de manera adecuada a las consultas de los usuarios (Henriquez Miranda et al., 2024).

2.2. Marco Conceptual

2.2.1. Aplicaciones de la IA

La IA tiene aplicaciones en múltiples sectores, incluyendo la educación, donde se utiliza para personalizar el aprendizaje y mejorar la eficiencia operativa (Ocaña-Fernandez et al., 2019).

2.2.2. IA en la educación

La implementación de IA en la educación ha demostrado beneficios significativos, tales como la personalización del aprendizaje y la provisión de soporte continuo a los estudiantes. Los chatbots educativos, por ejemplo, pueden ofrecer respuestas inmediatas y personalizadas, mejorando la experiencia de aprendizaje (Henriquez Miranda et al., 2024).

2.2.3. PLN en la Educación

El PLN mejora la interacción y comprensión en los chatbots educativos, facilitando respuestas más precisas y contextuales a las consultas de los estudiantes (Garibay, 2020).

2.2.4. Metodología A.D.D.I.E

La metodología A.D.D.I.E. se utiliza para el diseño y desarrollo de programas educativos, y consta de cinco fases: Análisis, Diseño, Desarrollo, Implementación y Evaluación. Esta metodología asegura que el proyecto se planifique y ejecute de manera sistemática y eficiente (Morales González & Morales González, 2022) .

2.2.5. Metodología SCRUM

SCRUM es una metodología ágil que se enfoca en la entrega rápida y continua de software funcional. Se basa en iteraciones cortas llamadas Sprints y reuniones diarias para monitorear el progreso y ajustar el plan según sea necesario. Esta metodología es especialmente útil en proyectos de desarrollo de software debido a su flexibilidad y capacidad de respuesta a cambios (Cesar et al., 2018).

2.2.6. Gestión de la Información en Instituciones Educativas

La dispersión de documentos y la sobrecarga de trabajo para el personal administrativo son problemas comunes en las instituciones educativas. Esto resulta en tiempos de respuesta prolongados y una experiencia de usuario subóptima (Peña-Torres et al., 2021)

Las soluciones tecnológicas, como los sistemas de gestión documental y los chatbots, pueden mejorar significativamente la gestión de la información. Los chatbots, en particular, pueden proporcionar respuestas rápidas y precisas a las consultas, liberando al personal administrativo para otras tareas (Álvarez et al., 2018).

La implementación de tecnología avanzada, como los chatbots, ha demostrado mejorar la eficiencia operativa y la satisfacción del usuario en diversas instituciones educativas (Henriquez Miranda et al., 2024).

2.2.7. OpenAI ChatGPT y APIs

OpenAI es una organización que desarrolla y promueve la inteligencia artificial amigable para el beneficio de las personas. Sus modelos de IA, como GPT-3, proporcionan capacidades avanzadas de generación y comprensión de lenguaje natural, siendo una herramienta valiosa para el desarrollo de chatbots. Los chatbots desarrollados con GPT-3 pueden ser utilizados en diversos contextos, incluyendo servicio al cliente, creación de contenidos y traducción de idiomas, generando respuestas en múltiples idiomas. OpenAI ChatGPT está disponible a través de la API OpenAI, lo que permite a los desarrolladores acceder e integrar el chatbot en sus

aplicaciones y sistemas, facilitando así la implementación de soluciones innovadoras y eficientes en distintos sectores (Aydin & Karaarslan, 2022).

Las APIs (Application Programming Interfaces) permiten la integración de diferentes sistemas y servicios, facilitando la comunicación y el intercambio de datos entre aplicaciones. Son fundamentales para la implementación de chatbots avanzados que requieren acceso a múltiples fuentes de información (Monroy, 2022).

2.2.8. Python como Herramienta para el Desarrollo de IA

Python es un lenguaje de programación ampliamente utilizado en el desarrollo de soluciones de IA debido a su simplicidad y la amplia gama de bibliotecas disponibles, como TensorFlow, PyTorch y scikit-learn. Estas herramientas facilitan el desarrollo e implementación de modelos de IA complejos.

Python facilita el desarrollo de chatbots gracias a sus poderosas bibliotecas y frameworks. Herramientas como Flask y Django permiten crear aplicaciones web que integran chatbots de manera eficiente (Monroy, 2022).

2.2.9. Streamlit

Streamlit es un framework de código abierto que permite la creación rápida de aplicaciones web interactivas en Python. Es especialmente útil para desarrollar prototipos y aplicaciones de datos, facilitando la visualización y la interacción con modelos de IA. (Pokhrel et al., 2024).

Con Streamlit, es posible desarrollar aplicaciones en línea con facilidad. El marco admite varias bibliotecas de Python, incluidos NumPy, pandas, scikit-learn, Keras, PyTorch y SymPy (LaTeX). Usando Streamlit, los desarrolladores pueden crear aplicaciones web interactivas y visualmente atractivas sin preocuparse por el código subyacente (Musthafa & Kumar, 2023).

2.2.10. Mongo DB

MongoDB es la base de datos más popular entre las bases de datos NoSQL. Para la construcción de almacenes de datos, es una herramienta excelente, especialmente por su capacidad de utilizar completamente la llamada "arquitectura de clúster sin fragmentación". Es una base de datos de código abierto, lo que la hace ideal para construir almacenes de datos de alto rendimiento.

MongoDB es ampliamente utilizado en aplicaciones web y móviles, análisis de big data, y sistemas de gestión de contenido. Su capacidad para manejar grandes volúmenes de datos no estructurados lo hace ideal para el desarrollo de aplicaciones modernas que requieren flexibilidad y escalabilidad (Anjali Chauhan, 2019).

2.2.11. Elasticsearch

Elasticsearch es un motor de búsqueda y análisis de texto. Es utilizado para buscar, analizar y visualizar grandes volúmenes de datos en tiempo real. Elasticsearch permite realizar búsquedas complejas y análisis de datos con gran velocidad y precisión, gracias a su capacidad de indexar y buscar a través de grandes volúmenes de información (Shah et al., 2022).

2.3. Marco Legal

En cuanto al ámbito legal, el presente proyecto se regirá bajo ciertas normas o legalidades:

Para la implementación de un chatbot con IA en una institución educativa en Colombia, se deben tener en cuenta varias leyes y normativas. En particular, la Ley 1581 de 2012, también conocida como la Ley de Protección de Datos Personales, establece los requisitos para el tratamiento de los datos personales en Colombia. Esta ley otorga a los individuos varios derechos, incluyendo el derecho a conocer, actualizar y rectificar su información personal, y a solicitar pruebas de la autorización para el tratamiento de sus datos personales.

Además, la Ley 1712 de 2014, conocida como la Ley de Transparencia y del Derecho de Acceso a la Información Pública Nacional, establece que las instituciones públicas deben facilitar el acceso a la información pública. Aunque esta ley no se aplica directamente a las instituciones educativas privadas, la implementación de un chatbot con IA puede ayudar a mejorar la transparencia y el acceso a la información.

También es relevante la Ley 1341 de 2009 o Ley de TIC, que define un marco para el desarrollo y la regulación de las tecnologías de la información y las comunicaciones en Colombia. Esta ley fomenta la utilización y el aprovechamiento de las TIC para contribuir al desarrollo económico, social y cultural del país.

Ley 170 de 1994 por medio de la cual se aprueba el Acuerdo por el que se establece la Organización Mundial de Comercio (OMC).

Artículo 10: 1. “Los programas de ordenador, sean programas fuente o programas objeto, serán protegidos como obras literarias en virtud del Convenio de Berna (1971).” En Colombia, el registro se hace en la Dirección Nacional de Derechos de Autor del Ministerio del Interior. Para el registro se aporta una copia de la obra y se llena el formulario dispuesto para ello. El servicio es gratuito. Como el software se compara a la obra literaria también se debe registrar ante la Dirección Nacional De Derechos de Autor. El registro no es constitutivo de los derechos, sino solamente declarativo y no es obligatorio.

Ley 1273 de 2009 expedida por el Congreso de la República de Colombia, por medio de la cual se modifica el Código Penal, se crea un nuevo bien jurídico tutelado -denominado "de la protección de la información y de los datos"- y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones. (Secretaría Jurídica Distrital, 2009).

De acuerdo a la ley 1266 del 31 de diciembre de 2008, en la cual se habla acerca del tratamiento de datos de las personas, ya sea para su divulgación en los diferentes aspectos que se puedan denotar, los cuales no deben afectar a las personas en ningún sentido y en

dado caso que suceda se proceda a corregir; de igual forma el tratamiento de datos dentro de software debe ser de forma confidencial sin afectar la integridad de las mismas. (Presidencia de la República, s.f.)

2.4. Marco Contextual

El uso de la tecnología en la educación ha crecido exponencialmente en los últimos años. En particular, el uso de chatbots con IA ha demostrado ser prometedor para mejorar la experiencia educativa. Varios estudios han mostrado que los chatbots pueden mejorar la eficiencia de la educación y la satisfacción de los estudiantes al proporcionar un acceso más fácil a la información y un aprendizaje más personalizado. Este proyecto tiene como finalidad, facilitar el acceso de la información del Instituto Tecnológico del Putumayo, a la comunidad académica y a promover la autonomía en el aprendizaje.

2.4.1.1. Espacio de investigación

2.4.1.2. Instituto Tecnológico del Putumayo

El Instituto Tecnológico del Putumayo (ITP) se encuentra ubicado en la ciudad de Mocoa, capital del departamento del Putumayo, en la República de Colombia. Su dirección actual es: Barrio Luis Carlos Galán, Mocoa, Putumayo. Esta institución de educación superior es un pilar fundamental en la formación técnica y profesional en la región, contribuyendo significativamente al desarrollo educativo y socioeconómico del Putumayo.

Capítulo 3: Metodología

2.5. Enfoque y Tipo de Investigación

El enfoque de esta investigación es mixto, combinando métodos cuantitativos y cualitativos para proporcionar una comprensión integral del impacto y la efectividad de la implementación de un chatbot con inteligencia artificial en el Instituto Tecnológico del Putumayo (ITP). Este enfoque permite no solo medir el desempeño y la eficiencia del chatbot mediante datos cuantitativos, sino también explorar las experiencias y percepciones de los usuarios a través de datos cualitativos.

Para llevar a cabo esta investigación, se utilizaron dos metodologías complementarias: A.D.D.I.E. para el desarrollo del chatbot y SCRUM para la gestión del proyecto.

La metodología A.D.D.I.E, es un enfoque secuencial que se utiliza en el desarrollo de software para guiar el proceso de creación de una aplicación o programa. Este enfoque consta de 5 etapas como se observa en la **Figura 1**, esta metodología tiene un propósito específico y se lleva a cabo de manera ordenada para garantizar un desarrollo eficiente y de calidad *(Morales González & Morales González, 2022)*.

Figura 1

Etapas modelo A.D.D.I.E

A.D.D.I.E				
Análisis	**Diseño**	**Desarrollo**	**Implementación**	**Evaluación**
Esta etapa consiste en la recopilación de información y definición de necesidades. Aquí es donde se pone en claro el "qué", "por qué" y "para quién" de un proyecto. Se investigan a fondo los objetivos, el público objetivo y los recursos disponibles. También es una etapa de oportunidad para identificar posibles desafíos y diseñar estrategias para superarlos.	En esta etapa se toma la información recopilada durante la fase de análisis y se utilizan para planificar cómo será el proyecto en términos de estructura, contenido, interacción, etc. Es como diseñar la arquitectura o planos de una casa antes de empezar a construirla.	Esta etapa, es donde las ideas cobran vida. Utilizando el plan o diseño como guía, se comienza a crear el contenido, a desarrollar las interacciones y a construir el proyecto. Es la etapa de "construcción" en el proceso.	Aquí es donde el proyecto, se coloca en acción. Esta etapa es la prueba real del valor y la efectividad del trabajo que se ha realizado hasta ahora.	Esta es la etapa de reflexión y ajuste. Es donde se evalúa el éxito del proyecto, se recoge los resultados obtenidos y se hacen las modificaciones necesarias. Esta etapa es crucial para el aprendizaje y la mejora continua. Es la oportunidad de ver lo que funcionó, lo que no funcionó y cómo se puede hacer mejor.

Elaboración propia.

En cuanto a la gestión del proyecto se llevó a cabo mediante, el uso de la metodología Scrum, esta se concibe en su definición como un marco de trabajo, que provee las herramientas necesarias para realizar proyectos de desarrollo de software de una forma ágil y efectiva.

La estructura de SRUM se basa en ciclos “sprint”, que suelen durar entre dos a cuatro semanas, en nuestro caso los Sprint tendrán un lapso de tiempo de dos semanas, donde al final de cada Sprint, se tendrá una entrega funcional del chatbot, se revisará el backlog y priorizará al inicio del proyecto y antes de cada sprint.

Figura 2

Metodología SCRUM



Nota. Metodología Scrum, por Nuria, 2023, boardmix (https://boardmix.com/es/knowledge/what-is-scrum/)

2.6. Etapa de Análisis

En esta etapa se, se recopilo y se analizó la información sobre chatbots que utilizan IA en el procesamiento y manejo de documentos con el fin de identificar, algunas de las características, funcionalidades y casos de éxito de los chatbots en distintos campos para

comprender como se pueden aplicar de manera efectiva y eficiente, este proceso se realizó mediante una revisión sistemática, para ello se realizó los siguientes pasos:

Se definió el **Objetivo general de la revisión Sistemática**:

- Investigar y analizar la literatura sobre la eficiencia y eficacia de chatbots con inteligencia artificial, con un enfoque especifico en el manejo de informacion y documentos.

Se realizaron los **Objetivos específicos de la revisión Sistemática**:

- Identificar estudios, artículos y documentos que describan o evalúen la Implementación y desempeño de chatbots con inteligencia artificial, específicamente en lo que respecta al manejo de informacion y documentos.
- Analizar las mejores prácticas, desafíos y beneficios asociados con la implementación de chatbots, para extraer recomendaciones que puedan aplicarse a la página principal del instituto tecnológico del putumayo.

Se agregaron **Criterios de inclusión**:

- Estudios publicados en revistas científicas.
- Trabajos centrados en chatbots que utilizan algún tipo de inteligencia artificial.
- Publicaciones que incluyan evaluaciones de desempeño, eficiencia, o eficacia de chatbots en el manejo de información.
- Estudios publicados en los últimos 5 años para asegurar la relevancia tecnológica.
- Trabajos en idiomas inglés y español para facilitar la interpretación de resultados.

Se tomo en cuenta también los **Criterios de exclusión**:

- Estudios publicados hace más de 5 años.
- Trabajos en idiomas diferentes al inglés y al español.
- Documentos que traten el tema de inteligencia artificial sin especificar su relación o aplicación en chatbots.
- Publicaciones que no tengan referencias bibliográficas.

Algunas de las **Fuentes de Búsqueda** que usamos fueron las siguientes:

- Bases de datos académicas (PubMed, IEEE, Scopus, Web of Science)
- Google Scholar
- Repositorios institucionales o bases de datos de tesis
- Bibliotecas digitales específicas en el campo de la informática o la inteligencia artificial.

Para facilitar la búsqueda fue necesario realizar algunas combinaciones de palabras clave y estas fueron algunas **Estrategia de Búsqueda**:

Básicas:

- "Chatbots" AND "Manejo de Información" AND "Desempeño" AND "Inteligencia Artificial"

  Documentos encontrados: 77

- "Beneficios" AND "chatbots" AND "Implementación" AND "Manejo de Informacion" AND "Inteligencia Artificial"

  Documentos encontrados: 103

- "Mejores prácticas" AND "chatbots" AND "Manejo de información" AND Inteligencia Artificial"

  Documentos encontrados: 23

Mas especificas (Artículos):

- "Implementación" AND "Chatbots" AND "Inteligencia Artificial" AND "Eficacia"

Documentos encontrados: 24

- "Best Practices" AND "chatbots" AND "Information Management" AND "Artificial Intelligence"

Documentos encontrados: 123

Por último, se centró la investigación, específicamente en el tema que estamos trabajando y se definieron las preguntas de investigación:

- ¿Cómo los chatbots con inteligencia artificial han contribuido a mejorar el acceso a la información y la experiencia del usuario?
- ¿Qué algoritmos se han destacado en la literatura para la creación de los chatbots con inteligencia artificial y han demostrado ser los más eficaces?
- ¿Qué metodologías y herramientas se han utilizado con éxito para implementar chatbots con inteligencia artificial en sistemas ya existentes, en diferentes campos?
- ¿En qué campos, aparte de la educación, se han implementado chatbots con inteligencia artificial con éxito?
- ¿Qué modelos avanzados de procesamiento de lenguaje natural en chatbots, han demostrado ser los más eficientes y eficaces en su uso?

Además, para el desarrollo de la revisión se utilizaron herramientas como bases de datos, como Google Scholar, Scopus, IEEE, etc. También, se utilizó el gestor de búsqueda conocido como Mendeley.

De este modo se obtuvieron resultados de la revisión sistemática, estos fueron positivos e importantes, para el desarrollo del proyecto. (**Ver Anexo 1. Resultados Revisión Sistemática**)

Una vez realizada la revisión sistemática, se encontraron varias clases de chatbots que operaban de diferentes formas. Algunos operaban con modelos previamente entrenados mediante machine learning, mientras que otros estaban desarrollados para responder a preguntas y consultas previamente determinadas(Garibay, 2020). Los avances tecnológicos en cuanto al procesamiento del lenguaje natural y las respuestas generadas en tiempo real son recientes y están en constante evolución.

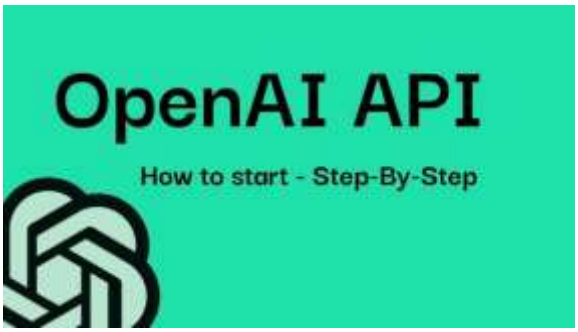
Se identificaron artículos que hacían uso de APIs, las cuales son actualmente las más utilizadas debido a que permiten el acceso a los modelos de IA más avanzados y su fácil accesibilidad. A raíz de esto, se exploraron diferentes formas de adaptar la inteligencia artificial al chatbot, basándonos en los diferentes artículos y recomendaciones. Inicialmente, se realizaron pruebas con la API de OpenAI, que mostró ser una solución viable para el proyecto.

También se evaluaron modelos de lenguaje locales (LLM) como Llama2, GPT-4all, Phi3, Mistral y Gemma. Aunque estos modelos son open-source y no incurren en costos de uso, presentaron varios desafíos, principalmente la gran cantidad de recursos computacionales necesarios para trabajar con múltiples usuarios simultáneamente. Esta limitación hizo inviable su uso en el contexto del proyecto, en el momento en el que realizamos la investigación, ya que el ITP requiere una solución que soporte múltiples usuarios de manera eficiente (**Anexo 3. Investigacion modelos LLM open Source y limitaciones**).

Al realizar pruebas sencillas con diferentes APIs de inteligencia artificial, se encontró que estas soluciones permitían trabajar con varios usuarios y obtener respuestas rápidamente. Por lo tanto, se optó por integrar la inteligencia artificial mediante APIs, asegurando así un rendimiento adecuado y una experiencia de usuario satisfactoria.

Figura 3

OpenAI API



Nota. Adaptado de How to start using OpenAI API, por Ofir Ziv, 2023, Medium (https://medium.com/@ofirziv/how-to-start-using-openai-api-step-by-step-38ff1ee4b15b).

En la Figura 3 se presenta la opción que se optó por usar, para la integración de la inteligencia artificial en nuestro proyecto, el api de OpenAI, específicamente en el uso del modelo de IA gpt-3.5-turbo por sus beneficios calidad y precio por solicitud.

Tabla 1

OpenAI modelos calidad - precio

Modelo IA	Contexto	Entrada (1000 tokens USD)	Salida (1000 tokens USD)
gpt-4o	128k	$0,005	$0,015
gpt-4	32k	$0,06	$0,12
gpt-3.5-turbo	16k	$0,0005	$0,0015
GPT-3.5-Turbo-Instruct	4k	$0,0015	$0,002

Entre otras de las opciones que se planteó usar están el api de Google con su modelo de IA Gemini y el api de Anthropic con su modelo Claude 2, sin embargo, se descartaron por sus costos de uso y escasa documentación relacionada con el procesamiento de información.

Para llevar una buena gestión del proyecto, estuvo presente la metodología SCRUM, se identificó los objetivos a alcanzar para el desarrollo del producto o programa final. Para esto se tomó como referencia las historias de usuario, que representan las necesidades que deben cubrir las funcionalidades del programa, y de este modo satisfacer las exigencias del cliente.

En cuanto a la realización del proceso de extracción de información relativa a las necesidades a cubrir, se llevó a cabo reuniones entre los miembros del equipo de desarrollo.

Tabla 2

Roles de los miembros del equipo

Roles	
Product Owner	Yan Carlos Cuaran Imbacuan
Scrum Master	Maicol Libardo Jossa
Team Member	Yan Cuaran, Maicol Jossa

Este proceso, gracias a la metodología Scrum, no limitará la fase inicial del proyecto, como ocurriría en un desarrollo clásico generalmente, si no que se tratará de un proceso iterativo en constante evolución para poder amoldarse a los requerimientos necesarios para el funcionamiento del programa.

Historias de usuario

Las historias de usuario se definirán utilizando el siguiente modelo:

Historia de Usuario	
ID	HUXX

Nombre	
Prioridad	
Riesgo	
Descripción	
Validación	

Significado de cada campo:

- ID: Se trata del identificador único asignado a este elemento del proyecto, se seguirá el formato HUXX para las historias de usuario.
- Nombre: Es nombre corto utilizado para describir muy brevemente la historia de usuario.
- Prioridad: Es la preferencia de cara al desarrollo de la historia de usuario respecto a las demás.
  - Valores: Alta, media y baja.
- Riesgo: Se trata de la importancia de la historia de usuario en relación al conjunto del proyecto. Cuantificando de este modo el daño provocado en caso de fallo.
  - Valores: Alto, medio y bajo.
- Descripción: Breve explicación de las intenciones de la historia de usuario. Debe dejar clara la idea de la propia historia.
- Validación: Son las condiciones que deben cumplirse una vez la historia está completamente desarrollada para que se pueda por finalizada.

2.6.1. Sprint 1: Configuración Inicial y Desarrollo del Backend

Historia de Usuario	
ID	HU01
Nombre	Configuración del entorno de desarrollo
Prioridad	Alta
Riesgo	Medio
Descripción	Como desarrollador, quiero configurar el entorno de desarrollo para asegurar que todas las dependencias y herramientas necesarias estén disponibles y configuradas correctamente.
Validación	El entorno debe estar configurado con Python, Streamlit, MongoDB, Elasticsearch y todas las dependencias necesarias instaladas. Debe estar documentado el proceso de configuración para que cualquier miembro del equipo pueda replicarlo.

Historia de Usuario	
ID	HU02
Nombre	Implementación de la estructura básica del backend
Prioridad	Alta
Riesgo	Medio
Descripción	Como desarrollador, quiero implementar la estructura básica del backend para manejar las solicitudes y respuestas del chatbot.
Validación	La estructura del backend debe estar definida y configurada. Debe estar documentada la lógica de manejo de solicitudes y respuestas.

Historia de Usuario	
ID	HU03
Nombre	Configuración de la base de datos MongoDB y el buscador vectorial Elasticsearch
Prioridad	Alta
Riesgo	Bajo

Descripción	Como desarrollador, quiero configurar la base de datos MongoDB para almacenar la información y registros de interacción del chatbot, además de configurar el buscador vectorial Elasticsearch para realizar la extracción y vectorización de informacion de la base de datos.
Validación	La base de datos debe estar configurada y conectada al backend. Elasticsearch debe estar configurado y conectado con el backend. Debe estar documentada la estructura de la base de datos y el buscador vectorial y su integración con el backend.

2.6.2. Sprint 2: Desarrollo del Frontend y Diseño de la UI

Historia de Usuario	
ID	HU04
Nombre	Diseño y desarrollo de la interfaz de usuario
Prioridad	Alta
Riesgo	Medio
Descripción	Como usuario del chatbot, quiero una interfaz intuitiva y accesible para interactuar con el sistema de manera eficiente.
Validación	La interfaz debe ser responsiva y funcionar en diferentes dispositivos. Deben estar implementados los componentes de interfaz utilizando HTML, CSS y Librerías de diseño de Streamlit.

Historia de Usuario	
ID	HU05
Nombre	Implementación de la comunicación entre el frontend y el backend
Prioridad	Alta
Riesgo	Medio
Descripción	Como desarrollador, quiero implementar la comunicación entre el frontend y el backend para que las solicitudes del usuario sean procesadas correctamente.
Validación	Deben estar configuradas las llamadas API para la comunicación con el backend. La integración debe estar documentada y probada.

2.6.3. Sprint 3: Integración de APIs de IA y Procesamiento de Lenguaje Natural

Historia de Usuario	
ID	HU06
Nombre	Integración de la API de OpenAI para procesamiento de lenguaje natural
Prioridad	Alta
Riesgo	Alto
Descripción	Como desarrollador, quiero integrar la API de OpenAI para procesar y generar respuestas utilizando inteligencia artificial.
Validación	La API de OpenAI debe estar integrada y configurada correctamente. Las respuestas generadas deben ser precisas y relevantes.

Historia de Usuario	
ID	HU07
Nombre	Implementación de módulos de IA para generación de respuestas
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario del chatbot, quiero recibir respuestas precisas y contextuales a mis consultas.
Validación	Los módulos de IA deben estar implementados y probados. Las respuestas deben ser evaluadas y ajustadas según sea necesario.

2.2.4. Sprint 4: Pruebas y Optimización

Historia de Usuario	
ID	HU08
Nombre	Realización de pruebas de funcionalidad y rendimiento
Prioridad	Alta
Riesgo	Medio

Descripción	Como desarrollador, quiero realizar pruebas de funcionalidad y rendimiento para asegurar que el chatbot funcione correctamente bajo diferentes condiciones.
Validación	Deben ejecutarse pruebas unitarias e integradas. Deben realizarse pruebas de carga y rendimiento.

Historia de Usuario	
ID	HU09
Nombre	Optimización de la interfaz y la lógica del chatbot
Prioridad	Media
Riesgo	Bajo
Descripción	Como desarrollador, quiero optimizar la interfaz y la lógica del chatbot para mejorar su rendimiento y usabilidad.
Validación	La interfaz debe ser rápida y eficiente. La lógica del chatbot debe estar optimizada para responder rápidamente.

2.2.5. Sprint 5: Pruebas de Usabilidad y Preparación para Despliegue

Historia de Usuario	
ID	HU10
Nombre	Realización de pruebas de usabilidad con usuarios reales
Prioridad	Alta
Riesgo	Medio
Descripción	Como usuario del chatbot, quiero que la interfaz sea fácil de usar y que las respuestas sean útiles y precisas.
Validación	Deben realizarse pruebas de usabilidad con usuarios reales. El feedback debe ser recopilado y analizado para realizar ajustes necesarios.

Historia de Usuario	
ID	HU12

Nombre	Funcionamiento en Windows
Prioridad	Alta
Riesgo	Alto
Descripción	Como usuario quiero que el programa funcione en el sistema operativo Windows 10
Validación	Necesito que el programa funcione en Windows 10. Diseño de la arquitectura del programa.

	Historia de Usuario
ID	HU11
Nombre	Preparación para el despliegue en el entorno de producción
Prioridad	Alta
Riesgo	Alto
Descripción	Como desarrollador, quiero preparar el entorno de producción para asegurar un despliegue exitoso del chatbot.
Validación	Deben configurarse el servidor y la infraestructura necesaria. Deben realizarse pruebas finales en el entorno de producción.

2.3. Etapa de Diseño

Basándose en la revisión sistemática, los requerimientos y el análisis de los diferentes modelos de IA, además de su integración mediante APIs, se definieron los requisitos funcionales y no funcionales del chatbot.

Tabla 3

Requisitos Funcionales y No Funcionales

Funcionales	No funcionales
1. Capacidad de Responder Consultas:	1. Escalabilidad:

- El chatbot debe ser capaz de responder consultas relacionadas con documentos institucionales. - Debe proporcionar respuestas precisas y relevantes basadas en el contenido de los documentos. 2. Procesamiento de Lenguaje Natural (PLN): - Debe integrar funciones de procesamiento de lenguaje natural para comprender y generar respuestas precisas. - Debe ser capaz de manejar diferentes tipos de preguntas, incluyendo preguntas abiertas y cerradas. 3. Disponibilidad: - El chatbot debe estar disponible 24/7 y ser accesible desde la página principal del ITP. 4. Sección de Administrador:	- El sistema debe ser escalable para manejar un gran volumen de consultas simultáneamente sin degradar el rendimiento. - Debe soportar la adición de nuevos módulos y funcionalidades sin afectar el desempeño. 2. Privacidad y Seguridad: - Debe garantizar la privacidad y seguridad de los datos de los usuarios. - Debe cumplir con las regulaciones de protección de datos. 3. Tiempo de Respuesta: - El tiempo de respuesta del chatbot debe ser inferior a 5 segundos en la mayoría de los casos. - Debe optimizarse para proporcionar respuestas rápidas incluso durante picos de alta demanda.

- Debe tener una sección de administrador para realizar los cambios pertinentes, como actualizar FAQs, categorías y documentos PDF. - Debe permitir la gestión del software y control de acceso. 5. Interfaz de Usuario: - Debe tener una interfaz de usuario intuitiva y fácil de usar para todos los niveles de usuarios. - Debe proporcionar una guía interactiva para ayudar a los nuevos usuarios a familiarizarse con el sistema. 6. Historial de Consultas: - Debe mantener un historial de consultas realizadas por los usuarios para referencia futura. - Debe permitir a los usuarios acceder y revisar sus consultas anteriores.	4. Interfaz de Usuario Intuitiva y Amigable: - El chatbot debe tener una interfaz de usuario intuitiva y amigable. - Debe ser accesible para todo tipo de usuarios. 5. Documentación del Sistema: - Debe haber documentación completa del sistema, incluyendo manuales de usuario y programador. - La documentación debe estar actualizada y accesible tanto para usuarios finales como para administradores del sistema. 6. Mantenimiento y Actualizaciones: - El sistema debe ser fácil de mantener y actualizar. - Debe permitir la implementación de actualizaciones.

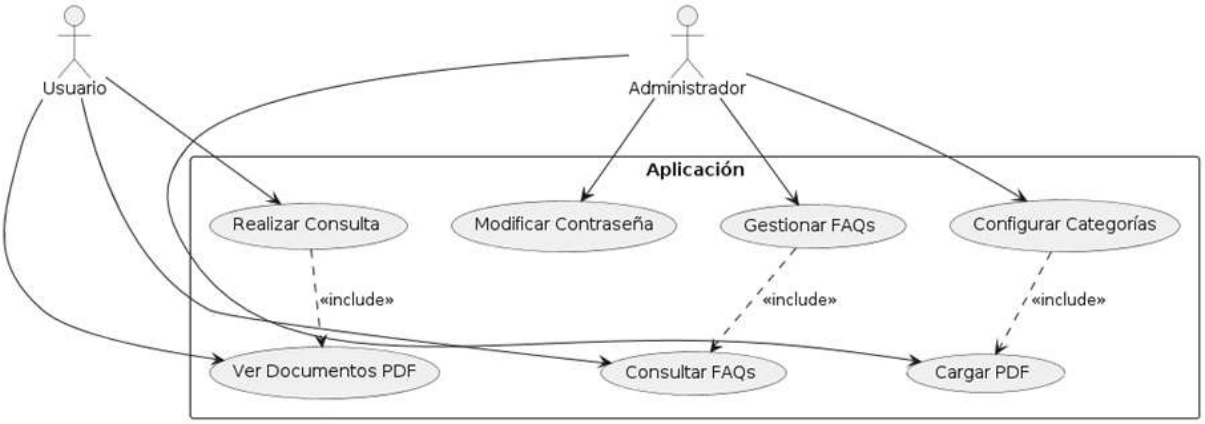
	7. Rendimiento y Eficiencia: – Debe ser eficiente en el uso de recursos del sistema, como memoria y CPU. – Debe garantizar que las consultas y operaciones se ejecuten de manera óptima.

Para documentar el funcionamiento lógico del chatbot y su interacción con los usuarios, se utilizará el Lenguaje Unificado de Modelado (UML). Su objetivo principal es proporcionar a los arquitectos de sistemas, ingenieros de software y desarrolladores de software herramientas efectivas para el análisis, diseño e implementación de sistemas basados en software (Laguia et al., 2022). En este caso se usaron para desarrollar los diagramas de caso de uso para representar gráficamente la funcionalidad del chatbot y cómo los usuarios interactúan con él.

Los siguientes diagramas de caso de uso (Figura 4, Figura 5 y Figura 6) se representa gráficamente la funcionalidad del chatbot inteligente y como el usuario interactuaría con el programa.

Figura 4

Diagrama de Caso de Uso – Funcionamiento General



En este caso de uso, tanto el usuario como el administrador pueden utilizar la aplicación de las siguientes maneras:

Usuario

El usuario puede:

- **Realizar Consulta:** El usuario puede enviar preguntas a la aplicación para obtener respuestas.
- **Ver Documentos PDF:** Como parte del proceso de consulta, el usuario puede acceder y ver documentos PDF relevantes a sus consultas.
- **Consultar FAQs:** El usuario puede buscar y consultar preguntas frecuentes para obtener información rápida y relevante.

Administrador

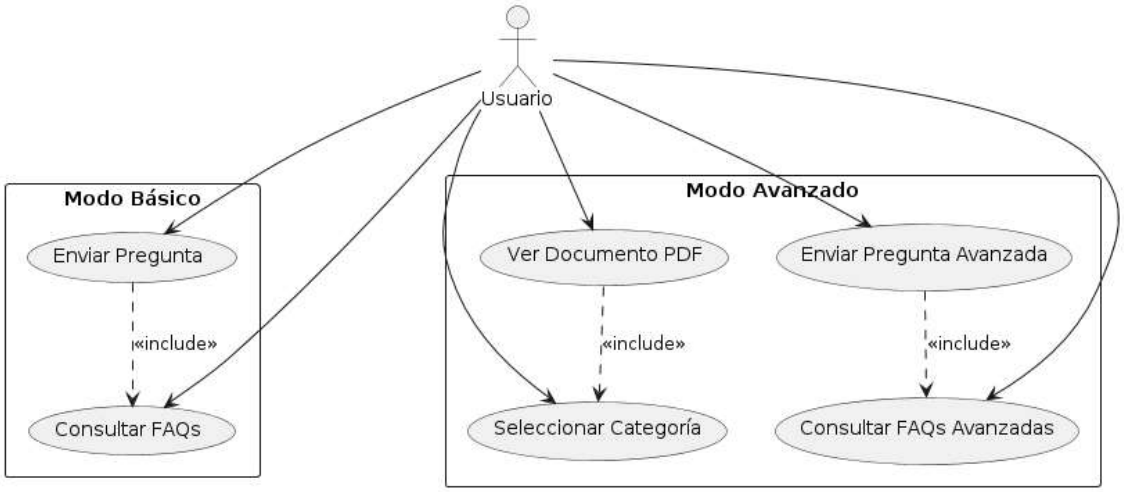
El administrador puede realizar las actividades que realizan los usuarios y además unos procesos extras como:

- **Modificar Contraseña:** El administrador puede cambiar la contraseña de acceso a las secciones protegidas de la aplicación.

- **Gestionar FAQs:** El administrador puede añadir, editar o eliminar preguntas frecuentes para mantener la base de datos actualizada.
- **Configurar Categorías:** El administrador puede añadir o eliminar categorías que organizan los documentos PDF y las FAQs.
- **Cargar PDF:** El administrador puede cargar nuevos documentos PDF en la base de datos de la aplicación.

Figura 5

Diagrama de Caso de Uso – Modos del Chatbot



En este caso de uso, el usuario puede interactuar con la aplicación en dos modos distintos: Modo Básico y Modo Avanzado. Aquí se describe cómo el usuario puede utilizar cada modo:

Modo Básico

En el modo básico, el usuario puede:

- **Ver Documentos PDF:** El usuario puede visualizar los documentos PDF que son relevantes para sus consultas.
- **Enviar Pregunta:** El usuario puede formular preguntas generales a la aplicación.

- **Consultar FAQs:** Como parte del proceso de enviar una pregunta, la aplicación puede incluir la consulta de preguntas frecuentes (FAQs) para proporcionar respuestas rápidas y predefinidas a consultas comunes.

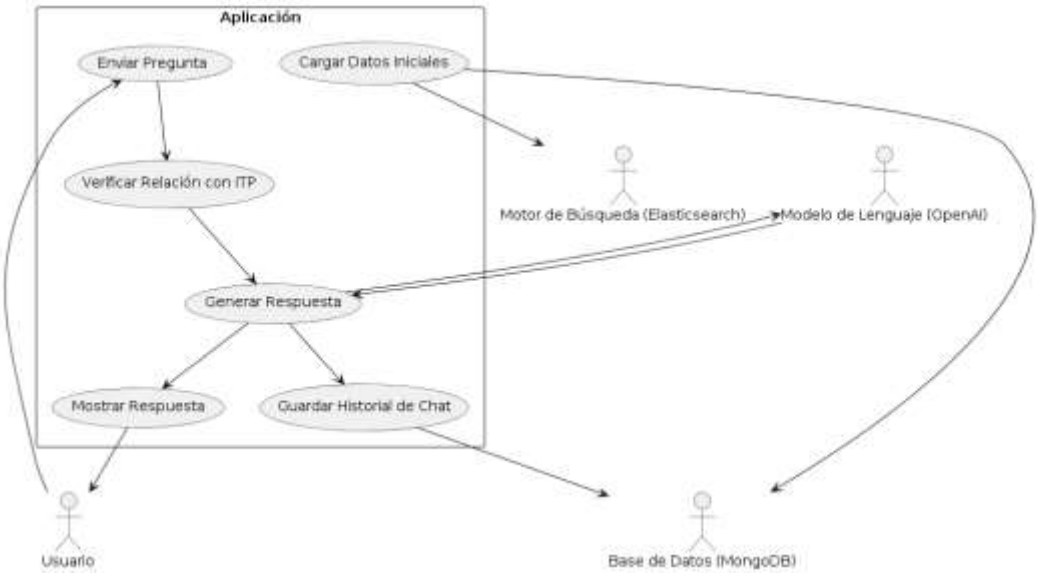
Modo Avanzado

En el Modo Avanzado, el usuario tiene acceso a funcionalidades más detalladas y específicas:

- **Ver Documento PDF:** El usuario puede visualizar documentos PDF que son relevantes para sus consultas.
- **Seleccionar Categoría:** El usuario puede seleccionar una categoría específica para centrar sus consultas en un documento o tema en específico y preguntas frecuentes.
- **Enviar Pregunta:** El usuario puede formular preguntas, obteniendo respuestas más detalladas y específicas sobre el documento o categoría seleccionada.
- **Consultar FAQs:** Como parte del proceso de enviar una pregunta, la aplicación puede incluir la consulta de preguntas frecuentes (FAQs) para proporcionar respuestas rápidas y predefinidas a consultas comunes**.**

Figura 6

Diagrama de Caso de Uso – Secuencia de funcionamiento



En este caso de uso, se describe el proceso completo que sigue la aplicación tras la consulta de un usuario:

Usuario:

- **Enviar Pregunta:** El usuario formula una pregunta a la aplicación.

Aplicación:

La aplicación sigue los siguientes pasos tras recibir una pregunta del usuario:

- **Enviar Pregunta:** La pregunta enviada por el usuario es recibida por la aplicación.
- **Verificar Relación con ITP:** La aplicación verifica si la pregunta está relacionada con el Instituto Tecnológico del Putumayo (ITP). Esta verificación puede implicar la consulta de palabras clave y temas relevantes almacenados.
- **Generar Respuesta:**

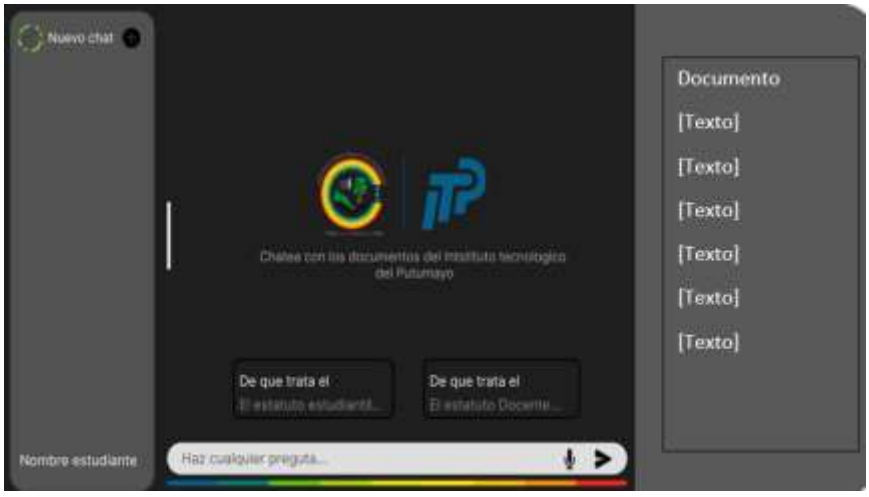
- **Motor de Búsqueda (Elasticsearch):** La aplicación puede consultar el motor de búsqueda para encontrar documentos relevantes que contengan información relacionada con la pregunta del usuario.
- **Modelo de Lenguaje (OpenAI):** La aplicación envía la pregunta y los documentos relevantes al modelo de lenguaje para generar una respuesta.

- **Guardar Historial de Chat:** La respuesta generada, junto con la pregunta del usuario, se guarda en la base de datos (MongoDB) para mantener un historial de las interacciones.
- **Mostrar Respuesta:** La respuesta generada se muestra al usuario en la interfaz de la aplicación.

El desarrollo de la interfaz, se realizó mediante bosquejos, que fueron realizados en aplicaciones web como Figma, Figura 7. Además de inspiración de diseños de otras páginas como ChatPDF Figura 8 y de plantillas básicas predeterminas de Streamlit (Figura 9 y Figura 10).

Figura 7

Interfaz principal

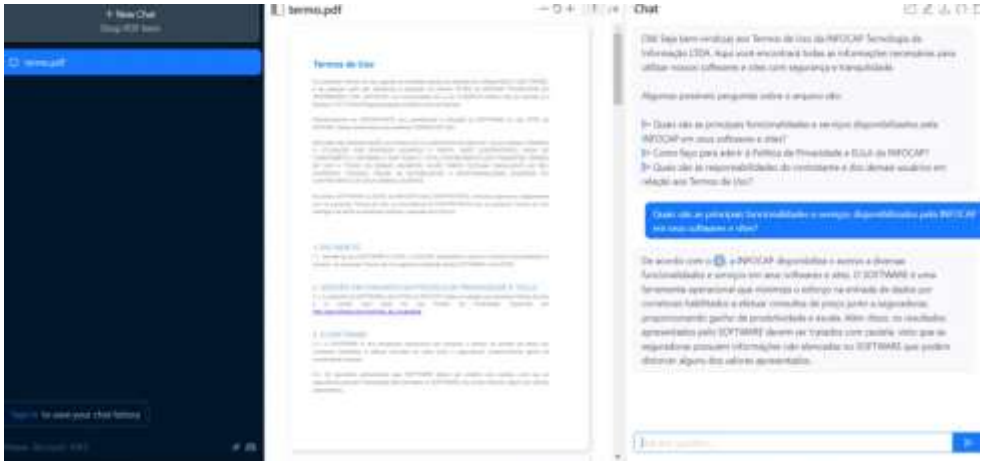


Nota. Adaptado de How to start using OpenAI API, por Ofir Ziv, 2023, Medium (https://medium.com/@ofirziv/how-to-start-using-openai-api-step-by-step-38ff1ee4b15b).

Además de inspiración de diseños de otras páginas como ChatPDF Figura 8.

Figura 8

interfaz ChatPDF

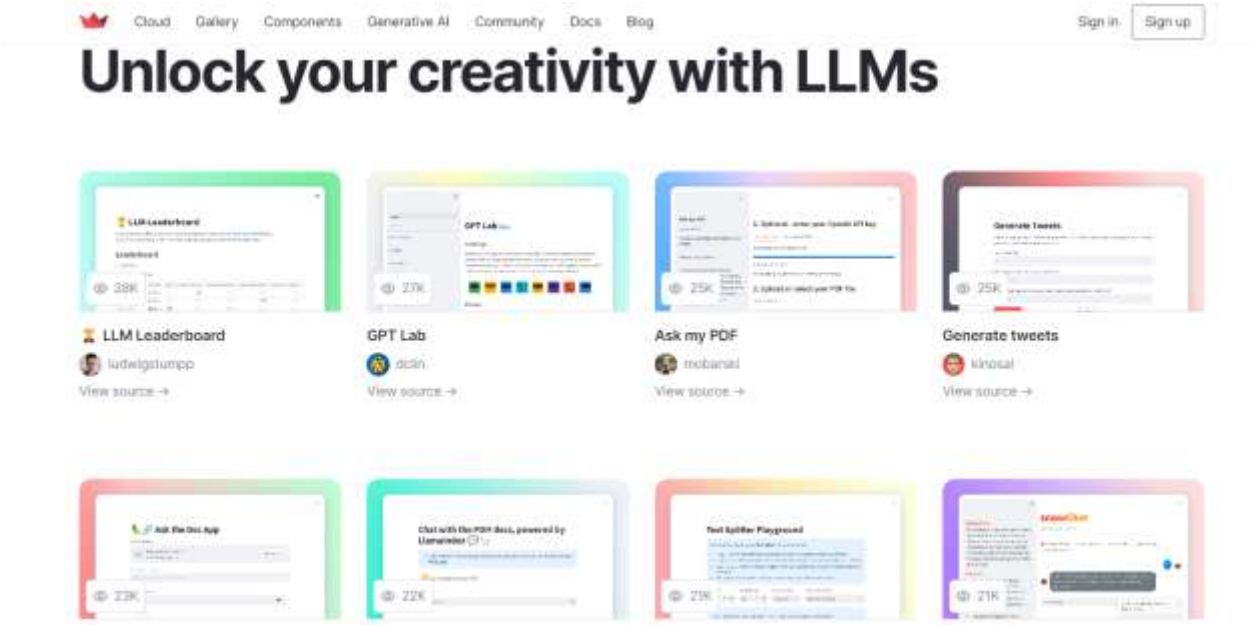


Nota. ChatPDF, Por Fabricio Calixto, 2023, Canaltech (https://canaltech.com.br/inteligencia-artificial/chatpdf-site-usa-chatgpt-para-transformar-qualquer-pdf-em-chatbot-252852/).

Plantillas básicas predeterminas de Streamlit (Figura 9 y Figura 10).

Figura 9

Plantillas Streamlit



Nota. Adaptado generative AI apps, por Streamlit, 2024, Streamlit (https://streamlit.io/generative-ai).

Figura 10

Plantillas Streamlit



Nota. Adaptado generative AI apps, por Streamlit, 2024, Streamlit (https://llm-examples.streamlit.app/).

2.4. Etapa de desarrollo

En esta etapa se realizó el seguimiento del desarrollo del programa, desde el fin del análisis, hasta completar todas las historias de usuario detalladas, esta etapa. Se va a distribuir en varias partes. Primeramente, una visión global del diseño, donde se explicará las tecnologías utilizadas y el motivo de su elección. Luego se detallará cada fase del desarrollo, llamada Sprint. En total se contará con cinco Sprint de cuatro semanas cada uno. Más adelante en el apartado de los Sprint se describirá y explicará con más detalle el funcionamiento y composición de éstos.

2.4.4. Herramientas de desarrollo

Para el diseño del programa se ha seleccionado una combinación de tecnologías y metodologías. La metodología a utilizar está ya predeterminada, por la naturaleza del proyecto, siendo imprescindible el empleo de la metodología Scrum. La combinación elegida para el diseño del programa es:

- Microsoft Visual Studio Code.
- Lenguaje Python.
- Mongo DB.
- Streamlit.
- CSS.
- HTML.
- Elasticsearch.

2.4.5. Desarrollo del Backend

En el desarrollo del backend, se inició con el **Sprint 1 Configuración Inicial y Desarrollo del Backend**. En la historia de usuario, pide que él software debe funcionar en

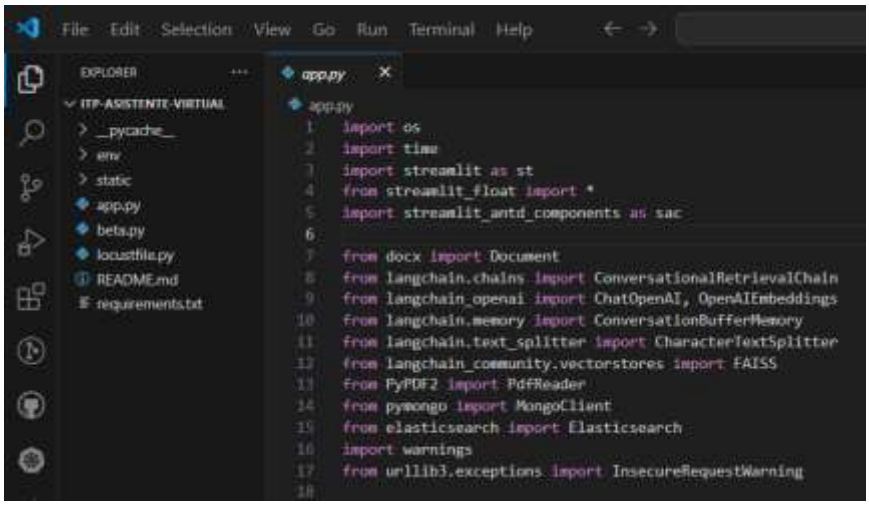
Windows, de ahí por esta razón, se ha considerado que las combinaciones, de estas herramientas son las más óptimas para su correcto funcionamiento.

La estructura del código es una arquitectura monolítica organizada y la configuración se realizó en base a los requerimientos que se plasman en los Sprint, esta etapa pondrá en ejecución los puntos establecidos anteriormente.

Se crea el proyecto en Visual Studio Code, se le asigna un nombre en este caso el proyecto lleva como nombre ITP asistente virtual.

Figura 11

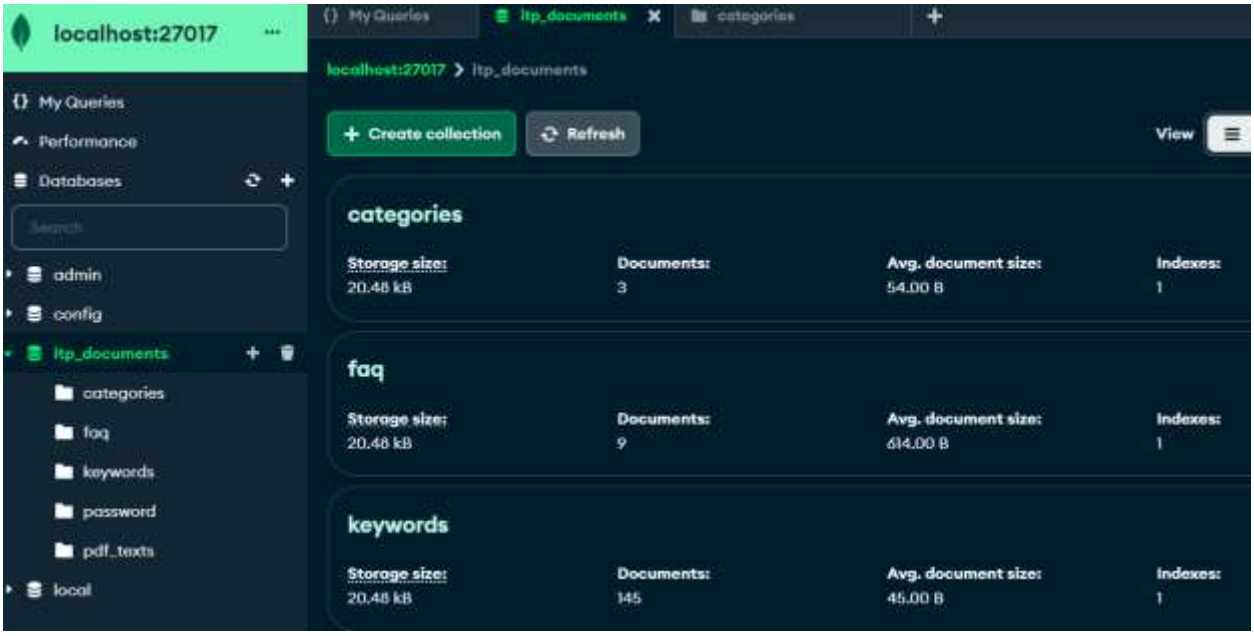
Creación del proyecto en Visual Code



Después, para la historia de usuario 3 (HU03), que se centra en la creación de la base de datos, se genera una base de datos en mongo db con el nombre itp_documents:

Figura 12

Creación y configuración de Mongo DB



Una vez creada la base de datos, denominada itp_documents, una base datos completamente no relacional. El siguiente procedimiento es estructurar el diseño de la base de datos. Para ello se determina la necesidad de la creación de las siguientes colecciones:

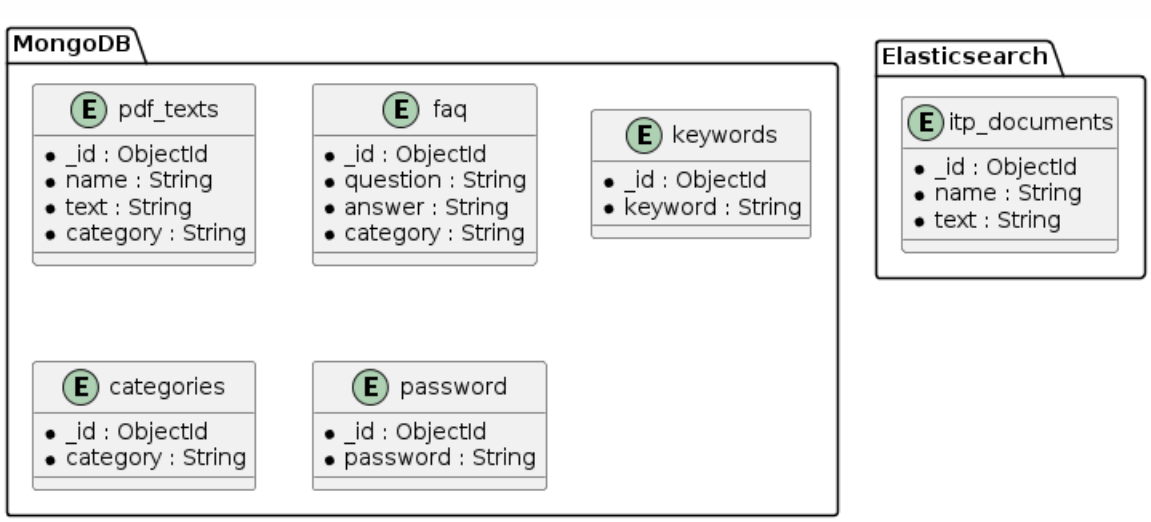
- **Categories:** Esta colección se encarga de almacenar las categorías de los documentos, predeterminadas por el administrador.
- **Faq:** Es la colección que se encarga de almacenar las preguntas frecuentes, predeterminadas por el administrador.
- **Keywords:** En esta colección se agregan algunas de las palabras, que se tendrán en cuenta para validar las consultas d ellos usuarios y evitar el uso indebido del programa.
- **Password:** Esta colección contiene la clave de acceso con la que el administrador, puede realizar cambios en el programa.

- **Pdf_texts:** En esta colección se almacenan los textos de los documentos o pdfs cargados al programa.

Teniendo en cuenta todos los datos anteriores, la estructura de la base de datos en Mongo db es la siguiente manera:

Figura 13

Estructura de la base de datos.



El siguiente proceso a realizar es configurar Mongo db y el buscador vectorial Elasticsearch.

Figura 14

Configuración Mongo db con el backend

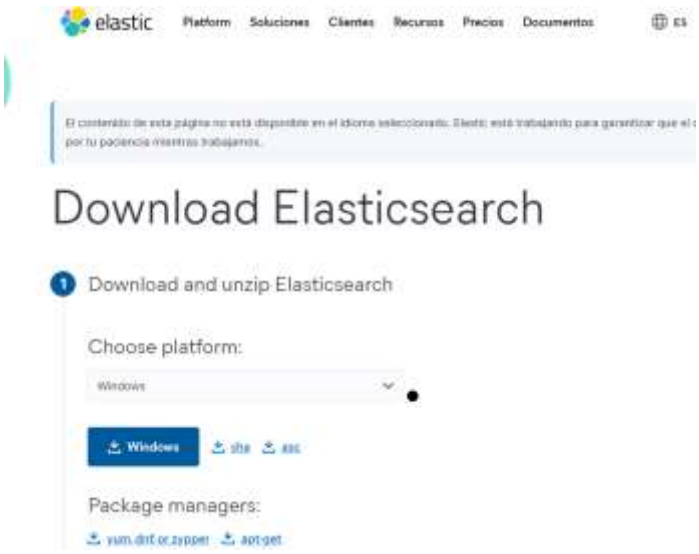
```
from pymongo import MongoClient

# Configuración de la conexión a MongoDB
client = MongoClient("mongodb://localhost:27017/")
db = client["itp_documents"]
collection = db["pdf_texts"]
collection_faq = db["faq"]
collection_keywords = db["keywords"]
collection_categories = db["categories"]
collection_password = db["password"]
```

En la Figura 14, se realiza la integración de la base de datos en el backend, con Python, se importan las librerías necesarias y se establece la conexión con el servidor local de Mongo db, para realizar los procesos de inserción, edición y eliminación de los datos.

Figura 15

Instalación de Elasticsearch



La instalación del buscador vectorial se realiza, ingresando al sitio oficial de Elasticsearch Figura 15, se descarga el archivo zip, se descomprime en un lugar de elección dentro del ordenador y se ejecuta el servidor de Elasticsearch abriendo un CMD en la raíz de la carpeta descomprimida y ejecutando el siguiente comando **bin\elasticsearch.bat.**

Figura 16

Configuración Elasticsearch con el backend

```
from elasticsearch import Elasticsearch

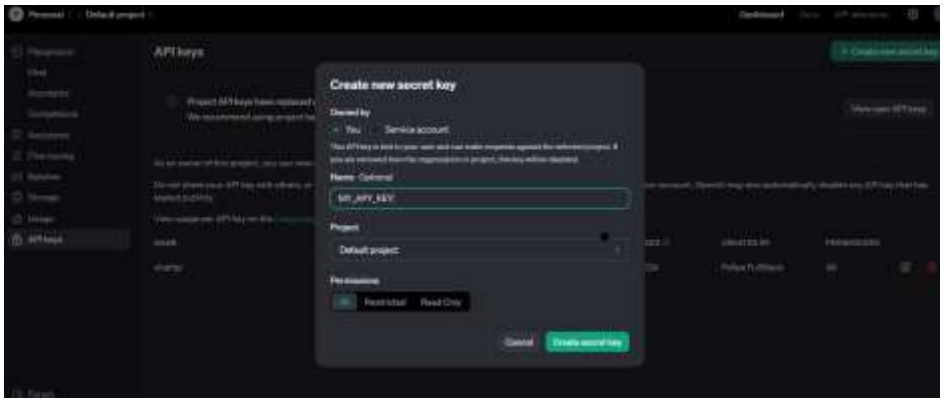
# Configuración de la conexión a Elasticsearch
es = Elasticsearch(
    ["https://localhost:9200"],
    basic_auth=("elastic", "blpR+66njAXsS3P2hP6w"),
    verify_certs=False
)
```

Una vez el servidor de Elasticsearch se encuentra en ejecución se realiza la configuración en el backend Figura 16, su función es analizar los textos procesados de los pdfs y vectorizarlos, con el fin de realizar la comparación de fragmentos en relación a las solicitudes realizada por el usuario y enviar solo los fragmentos relevantes a la API.

La integración de inteligencia artificial al programa es el siguiente proceso que se realizó siguiendo las indicaciones del **Sprint 3 Integración de APIs de IA y Procesamiento de Lenguaje Natural**, mediante la empresa de OpenAI, que brinda el servicio de prestación de modelos de IA mediante unas claves o en su defecto Api Keys.

Figura 17

Generación de Api Key



El proceso de obtención de la Apli Key, se puede realizar desde el sitio oficial de OpenAI Figura 17, es sencillo el proceso, hay que iniciar sesión e ingresar a la sección de Api Keys, se selecciona crear Secret Key, se le brinda un nombre al api y se le asigna los permisos que se requieren dependiendo el caso.

Figura 18

Integración de Api Key en el backend

```
import os
from langchain.chains import ConversationalRetrievalChain
from langchain_openai import ChatOpenAI, OpenAIEmbeddings


os.environ["OPENAI_API_KEY"] = "YOUR_API_KEY"  # OPENAI_API_KEY
```

El proceso siguiente es realizar la configuración para hacer uso de los servicios del api, se llaman las librerías, necesarias para el funcionamiento del programa como se observa en la Figura 18.

Figura 19

Integración del modelo de IA

```
def get_llm_chain(vectors):
    retriever = vectors.as_retriever()
    memory = ConversationBufferMemory(
        memory_key="chat_history",
        return_messages=True,
        output_key="answer"  # Especifica la clave de salida
    )
    llm_chain = ConversationalRetrievalChain.from_llm(
        llm=ChatOpenAI(model_name="gpt-3.5-turbo", temperature=0.7),
        retriever=retriever,
        memory=memory,
        return_source_documents=True  # Devuelve los documentos fuente
    )
    return llm_chain
```

La integración del modelo de inteligencia artificial, se realiza mediante una configuración, predeterminada en la documentación de OpenAI Figura 19, en este fragmento de código se configuro el modelo gpt-3.5-turbo, mediante el análisis de precio calidad que se representó en la Tabla1, además de configurar el modelo de IA, también permite manejar la temperatura del modelo, en pocas palabras el valor de la temperatura, entre más cerca este del 1 será más creativo y entre más cerca del 0 el programa reducirá el uso del lenguaje natural.

Detalles del Sprint 1

- **Duración:** 4 semanas
- **Estado:** Completado

Detalles del Sprint 3

- **Duración:** 4 semanas
- **Estado:** Completado

El funcionamiento completo del código se especifica en el manual de usuario que se presenta en los anexos (**Ver Anexo 5. Manual de Programador**).

2.4.6. Desarrollo del Frontend

En el desarrollo del Frontend comienza el **Sprint 2 Desarrollo del Frontend y Diseño de la UI** , se especifica que la interfaz debe ser intuitiva y accesible a la hora de interactuar con el sistema, algunas de las validaciones que se especifican, son que sea responsiva y funcionar en diferentes navegadores.

Lo más importante es que estén en comunicación entre el frontend y el backend para que las solicitudes del usuario sean procesadas correctamente.

Las tecnologías que se usaron para el desarrollo del Frontend fueron:

- **Streamlit:** Es una biblioteca de Python que facilita la creación de aplicaciones web para análisis de datos, que para este software en específico es ideal, además permite la creación de interfases interactivas.
- **HTML:** Es un estándar para crear y estructurar páginas web y aplicaciones web, en este caso, su función fue embeber el programa desarrollado en Streamlit y facilitar el proceso de implementación en la página del itp.
- **CSS:** Es un lenguaje utilizado para diseñar y personalizar la apariencia de las páginas web, en este caso se logró relacionar CSS con Streamlit y obtener modelos más personalizados.

Figura 20

Interfaz Principal del chatbot



En la Figura 20, se visualiza la pantalla de inicio del chatbot, esta es lo primero que se visualiza al iniciar el programa por default está configurado en el modo básico, este modo permite a los usuarios hacer preguntas sobre todos los pdfs subidos en la base de datos del software. En la parte izquierda se muestra una barra lateral con cuatro botones, cuyas funciones se explicarán de manera escalonada. En la parte superior central, hay dos botones que representan los modos de chat. A la derecha de la barra lateral, se encuentra el chat, que incluye un menú de categorías de preguntas. Finalmente, en la parte derecha del chat, se encuentran los contenidos de los Pdfs cargados en la base de datos del software.

Algunas de las funcionalidades que integra el software se visualizan a continuación:

Figura SEQ Figura * ARABIC 22
Categorías preguntas frecuentes (Estatuto 1)



Figura SEQ Figura * ARABIC 21
Categorías preguntas frecuentes (Estatuto 2)

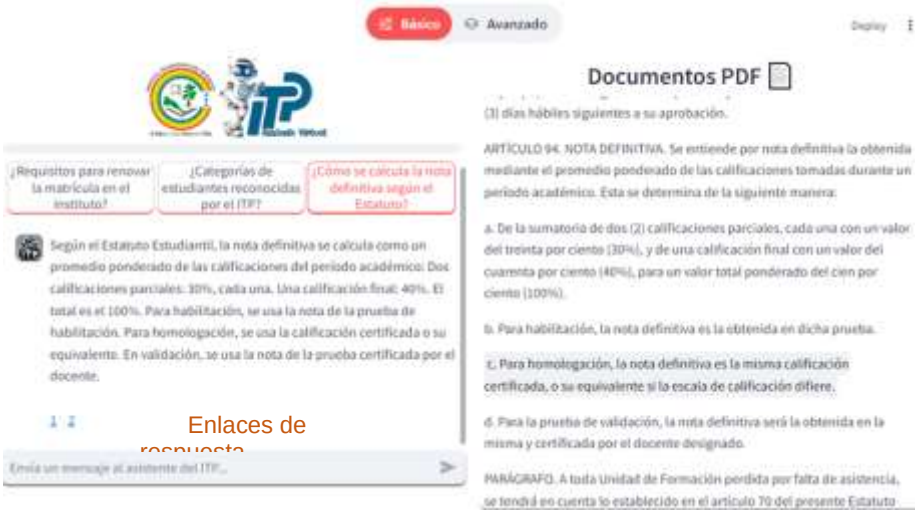


Los usuarios tendrán acceso a un menú de opciones de preguntas frecuentes como se observa en la (Figura 21 y 22). Los usuarios pueden seleccionar una opción y presionar el botón correspondiente para obtener una respuesta rápida. Este menú fue diseñado para ofrecer ideas base de preguntas, en caso de que los usuarios no sepan qué preguntar.

Al presionar una pregunta frecuente se obtiene una respuesta como se observa a continuación:

Figura 23

Interfaz principal, Realizar una solicitud



En la Figura 23 se puede observar dos números en la parte inferior del mensaje, los cuales actúan como enlaces o botones, que indican de dónde se obtuvo la información en los contenidos de los pdfs.

En la parte superior de la pantalla de inicio se encuentran dos botones uno que es el modelo básico del cual se mencionó anteriormente, al seleccionar el botón se obtiene la siguiente interfaz:

Figura 24

Interfaz principal, modo avanzado



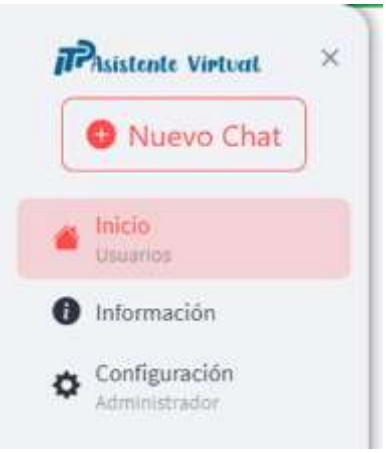
En este modo Figura 25, es necesario elegir una categoría antes de hacer una pregunta. Al seleccionar una categoría, se mostrarán las preguntas frecuentes y el PDF correspondiente. En resumen, el modo avanzado permite hacer preguntas a un solo PDF para obtener respuestas más precisas y detalladas.

El resto de las funciones son iguales a las del modo básico: se presiona un botón para obtener una respuesta, los enlaces o botones numerados indican la fuente de la información, y la entrada del chat está limitada a 200 caracteres.

En la barra lateral, tenemos cuatro botones: Nuevo chat, Inicio, Información y Configuración. Como se muestra en la interfaz a continuación:

Figura 25

Interfaz principal, barra lateral



Al presionar el botón "Nuevo chat" Figura 25, como su nombre lo indica, se iniciará una nueva conversación. Tener en cuenta que, al hacerlo, ya no se podrá ver los mensajes recibidos anteriormente.

El botón de inicio es el que esta seleccionado, por default puesto que, se encuentra la interfaz de la pantalla principal del software, a la que tienen acceso todos los usuarios y donde se encuentran las funciones explicadas anteriormente.

Al presionar el botón de informacion, se tendrá acceso a la informacion básica del chatbot, como se visualiza a continuación:

Figura 26

Interfaz Principal, Guía chatbot



Esta sección Figura 26, contiene detalles sobre cómo funciona cada modo de chat. Además, aquí se puede añadir información más detallada sobre qué es el software y cómo funciona programa.

Por último, tenemos el botón o sección de configuración como se visualiza a continuación:

Figura 27

Interfaz Secciones, configuración



Esta sección Figura 27, está reservada exclusivamente para el administrador, quien cuenta con una clave de acceso. Aquí se encuentran todas las configuraciones necesarias para añadir o eliminar opciones y/o documentos del chatbot.

Al ingresar la clave de acceso se tiene disponibilidad de las siguientes configuraciones que se visualizan a continuación:

Figura 28

Interfaz Secciones, Configuración Administrador



Detalles del Sprint 2

- **Duración:** 4 semanas
- **Estado:** Completado

El funcionamiento completo de la interfaz se especifica en el manual de usuario que se presenta en los anexos (**Ver Anexo 4. Manual de Usuario**).

2.5. Etapa de Implementación

Cómo se indicó al principio del proyecto, la metodología utilizada en la gestión del desarrollo del software es la metodología SRUM. Por este motivo el programa se va desarrollando en relación a las necesidades o requerimientos solicitados al principio en la etapa

de análisis en esta etapa se tiene en cuenta el **Sprint 5 Pruebas de Usabilidad y Preparación para Despliegue**.

En la siguiente tabla se presentan algunos aspectos técnicos (activos, versiones, paquetes) que fueron usados al momento del análisis, diseño y desarrollo del programa.

Tabla 4

Recursos del Programa

Lenguaje de Programación	**Python**
Software de diseño y Programación	Visual Studio Code (Ultima versión disponible)
	Figma
Frameworks	Streamlit
Base de Datos	MongoDB
Apis de IA	OpenAI (gpt-3.5-turbo)
Formato de Empaquetamiento	Entorno virtual de Python (virtualenv)
Requerimientos Computacionales Mínimos	• 6 – 8 GB RAM • 2,6 GHz Procesador • Windows 10

El entorno de desarrollo del chatbot se basa en Python, un lenguaje de programación orientado a objetos ampliamente utilizado en aplicaciones de inteligencia artificial. Para la gestión del backend y la creación de la interfaz de usuario, se utiliza el framework Streamlit, que facilita la construcción de aplicaciones web interactivas y rápidas. El manejo de datos se realiza mediante MongoDB, una base de datos NoSQL que ofrece flexibilidad y escalabilidad para almacenar información y registros de interacción.

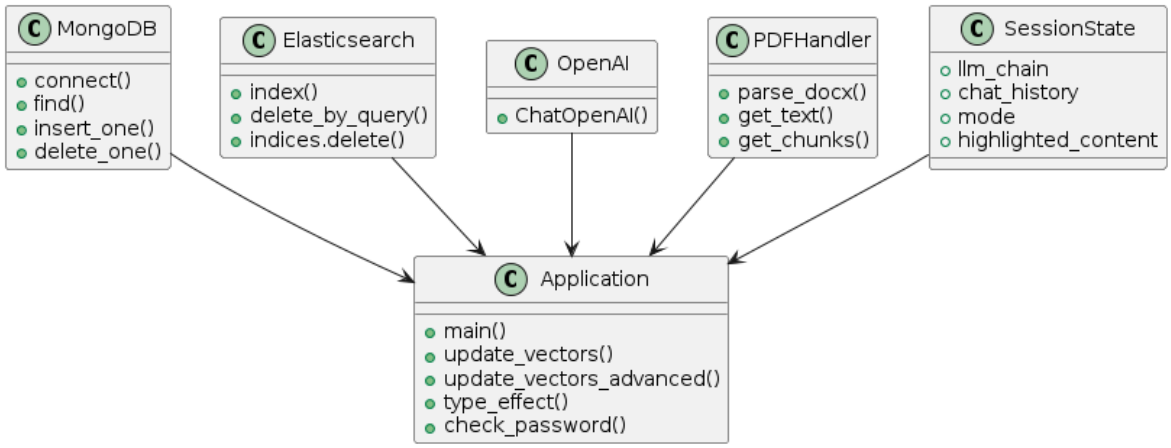
La inteligencia artificial del chatbot se potencia mediante la integración de la API de OpenAI (gpt-3.5-turbo), que permite el procesamiento y generación de respuestas contextuales utilizando técnicas avanzadas de procesamiento del lenguaje natural. Para asegurar la consistencia y facilidad de gestión del entorno de desarrollo, se utiliza un entorno virtual de Python (virtualenv), que permite aislar las dependencias necesarias y mantener un entorno de desarrollo limpio y replicable.

El software de edición y programación incluye Visual Studio Code, una herramienta potente y versátil, y Figma, que se utiliza para diseñar la interfaz de usuario. Estos componentes trabajan juntos para crear un sistema cohesivo y eficiente que interactúa con los usuarios de manera fluida y efectiva.

A continuación, se describe el comportamiento del sistema mediante un diagrama de clases que ilustra la interacción entre los componentes principales del chatbot Figura 29.

Figura 29

Diagrama de clases



La siguiente tabla contiene detalles de la aplicación para su publicación. El código fuente del repositorio es libre de uso y/o modificación.

Tabla 5

Especificaciones del programa

Especificaciones del Programa	
Nombre	ITP Asistente Virtual
Descripción	¿Necesitas información rápida y precisa sobre el Instituto Tecnológico del Putumayo? ¿Quieres resolver tus dudas sobre los servicios y estatutos institucionales? El Chatbot Asistente está aquí para ayudarte. Este chatbot interactivo utiliza inteligencia artificial para responder a tus consultas en tiempo real, mejorando la accesibilidad y eficiencia en la gestión de información.
Categoría	Educación
Sistema Operativo	Basado en Web
Desarrolladores	Yan Carlos Cuaran – Maicol Libardo Jossa
Repositorio en GitHub	https://github.com/yancarloscuaran/ChatbotITP
Enlace de la Pagina	https://asistentevirtual.itp.edu.co/
Clasificación de contenido	Todas las edades
Licencia	MIT License

A continuación, se describe el procedimiento que se llevó a cabo, para la implementación del chatbot en la infraestructura tecnológica del itp, la conexión se realizó vía remota mediante una VPN autorizada por las personas a cargo de los servidores del ITP.

Figura 30

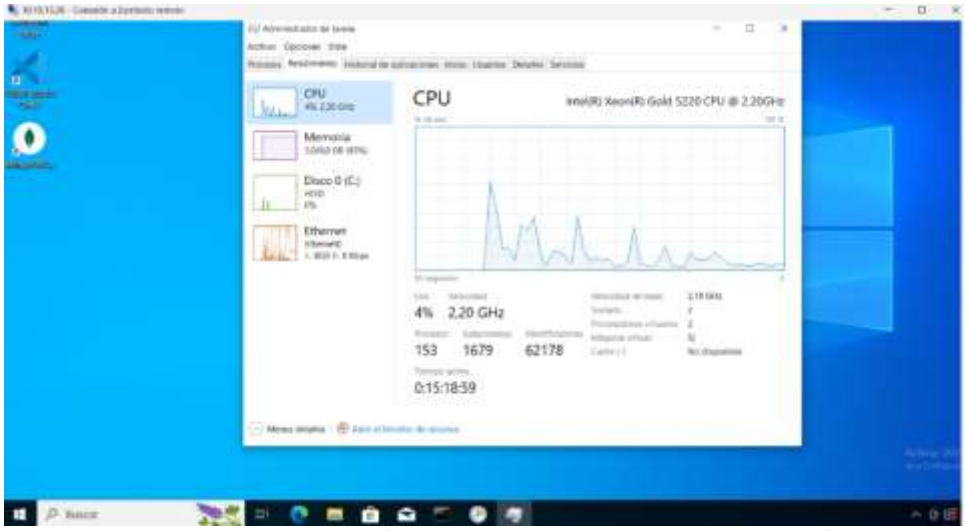
Configuración VPN



Una vez configurada la VPN, el siguiente paso, fue acceder a una maquina virtual designada, para realizar la configuración e instalación del chatbot, en la Figura 31 se visualizan algunas de las características de la, máquina virtual.

Figura 31

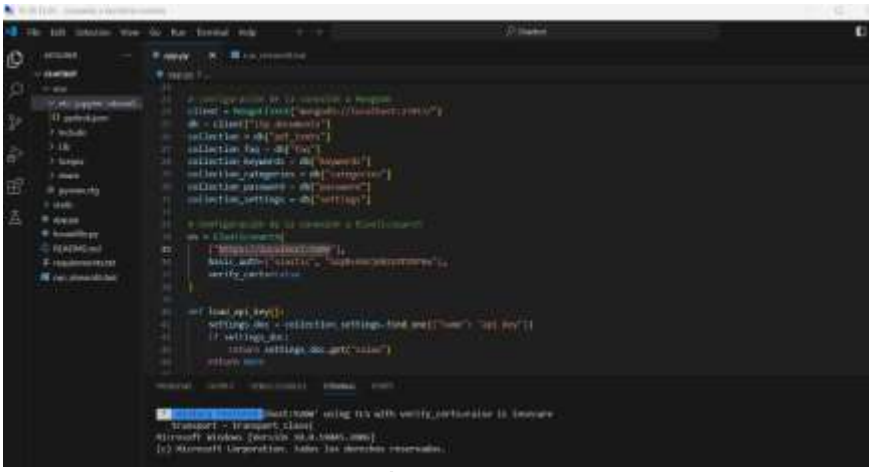
Características de la Máquina Virtual



En la Figura 32, se presenta el código del chatbot, alojado en la máquina virtual, con los requerimientos instalados.

Figura 32

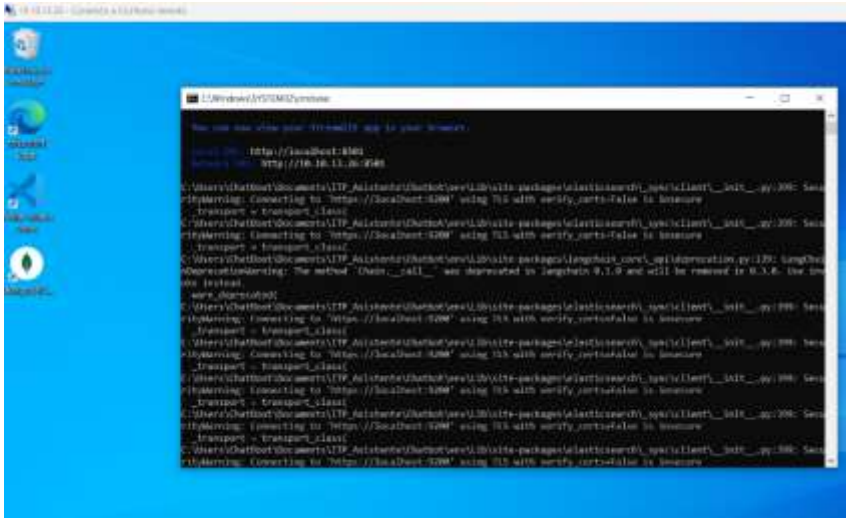
Alojamiento del Código



Una vez instalado el chatbot en la máquina virtual, el siguiente paso, es redireccionar el enlace o el puerto en donde se está ejecutando localmente el programa, a la página del ITP, este último proceso lo realizan los encargados de la página.

Figura 33

Ejecución del Chatbot en la Máquina Virtual



Detalles del Sprint 5

- **Duración:** 4 semanas
- **Estado:** Completado

2.6. Etapa de Evaluación

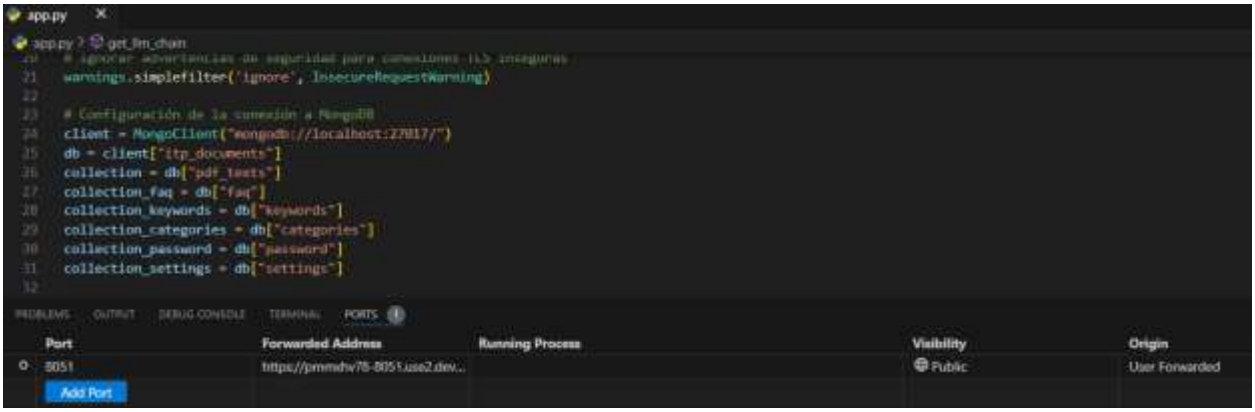
La etapa de evaluación se centró en medir la satisfacción de los usuarios y analizar el impacto del software en relación con los objetivos propuestos. Este proceso se dividió en varias fases para asegurar una evaluación completa y precisa del software en esta etapa se tuvo en cuenta el **Sprint 4 Pruebas y Optimización**.

El objetivo de la evaluación fue poner a prueba una version beta del software, con el fin de medir la satisfacción de los usuarios con el software y analizar el impacto del software en el cumplimiento de los objetivos propuestos.

El procedimiento que se llevó a cabo para la evaluación de eficiencia del software, fue que los estudiantes accedieron al software a través de un enlace generado en Visual Studio Code, configurando el ordenador como servidor para permitir el acceso a múltiples usuarios. Configuración Remote Tunnels

Figura

Configuración Remote Tunnels



Mediante el enlace generado los estudiantes navegaron por la aplicación sin supervisión directa, interactuando con los diferentes contenidos. Este enfoque permitió que los estudiantes exploraran libremente, identificaran errores y proporcionaran comentarios sobre el software.

Los participantes de la evaluación se llevó a cabo con estudiantes de 8vo y 10mo semestre de la carrera de Ingeniería de Sistemas en el Instituto Tecnológico del Putumayo.

Figura 34

Participantes 8vo semestre



Figura 35

Participantes 10mo semestre

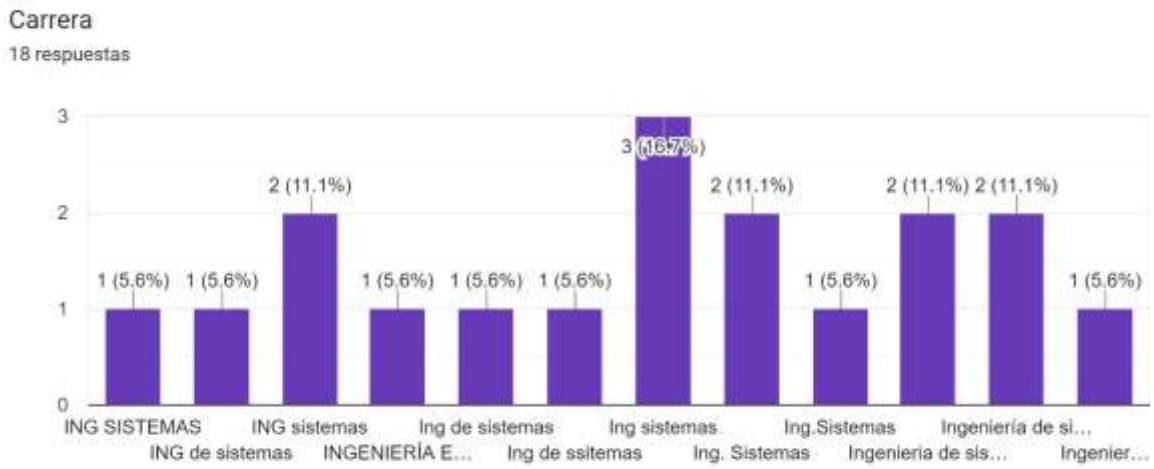


Una vez culminaron las pruebas del software, se realizó un formulario de usabilidad del chatbot, con el objetivo de recopilar datos sobre la experiencia de los estudiantes, se obtuvieron un total de 18 estudiantes, los cuales realizaron el cuestionario.

El formulario constaba de 15 preguntas diseñadas para evaluar la usabilidad, funcionalidad y eficacia del software.

Figura 36

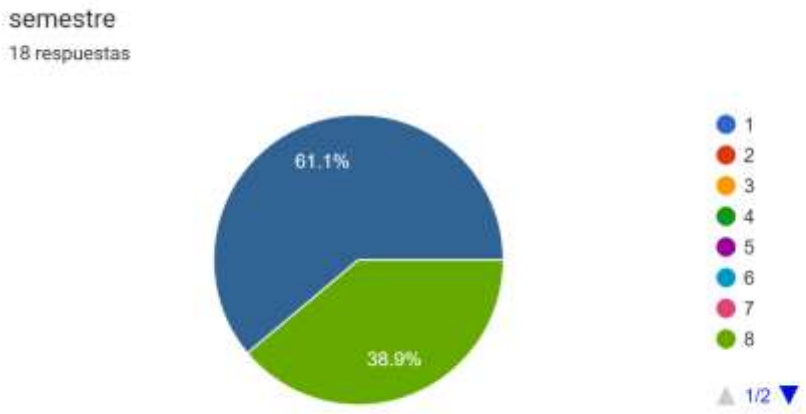
Pregunta - ¿Carrera a la que pertenece?



De acuerdo a la Figura 27, el 100% de los encuestados indicó que son de la carrera de Ingeniería de Sistemas.

Figura 37

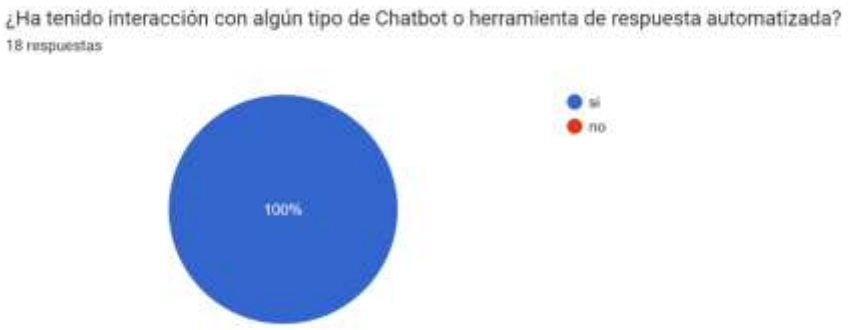
Pregunta - ¿Semestre al cual pertenece?



De acuerdo a la Figura 28, el 61.1% de estudiantes pertenecen a 10mo semestre y el 38.9% 8vo semestre.

Figura 38

Pregunta - ¿Ha tenido interacción con algún tipo de Chatbot o herramienta de respuesta automatizada?



De acuerdo a la Figura 29, el 100 % de los estudiantes, habían interactuado, alguna vez con un chatbot.

Figura 39

Pregunta - Si en la pregunta anterior su respuesta fue sí, por favor, cuéntenos como ha sido su experiencia con chatbots

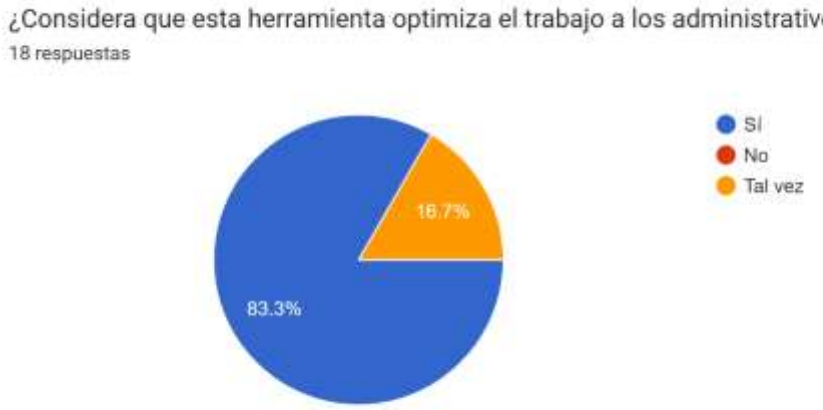


De acuerdo a la Figura 29, las preguntas 3 y 4 están relacionadas. Si los usuarios respondían afirmativamente a la pregunta 3, debían especificar cuáles chatbots habían

utilizado. Se concluyó que los usuarios interactuaron con chatbots basados en inteligencia artificial y que su experiencia general con estos fue positiva.

Figura 40

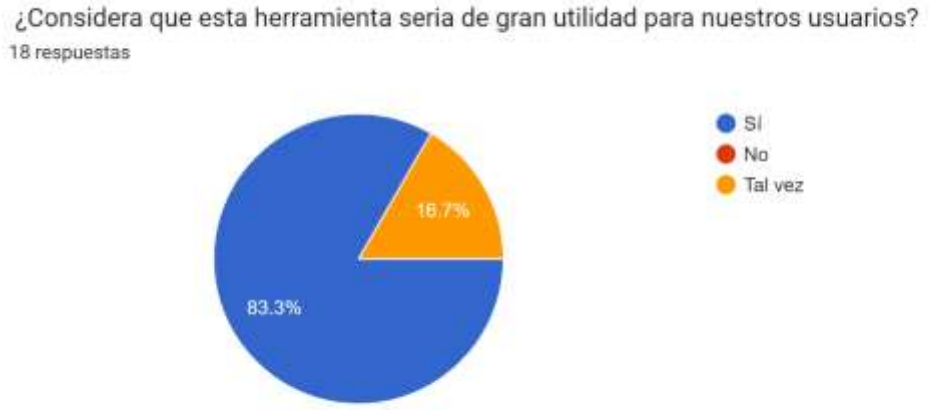
Pregunta - ¿Considera que esta herramienta optimiza el trabajo a los administrativos?



De acuerdo a la Figura 29, los estudiantes estuvieron de acuerdo en su gran mayoria que el chatbot, sería de gran ayuda, para los administrativos, puesto que reduciría un poco la carga que tienen.

Figura 41

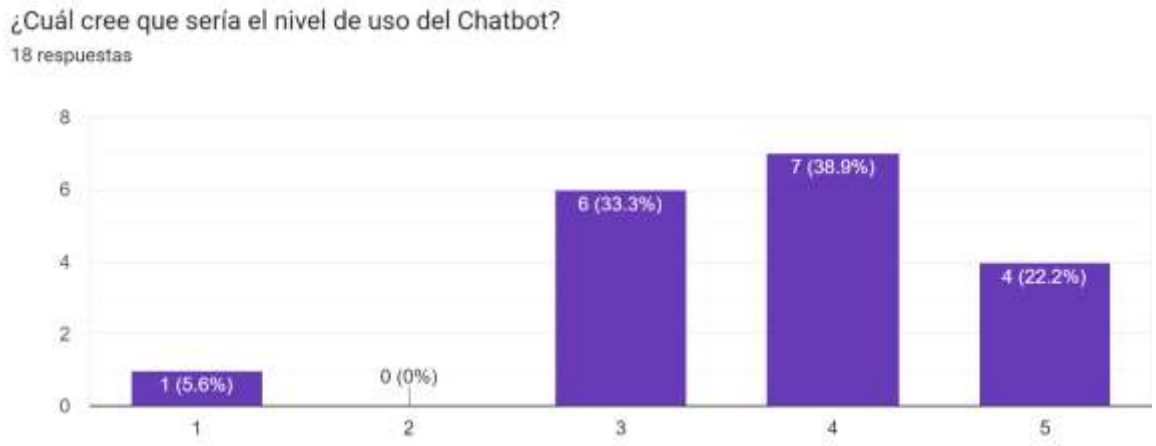
Pregunta - ¿Considera que esta herramienta sería de gran utilidad para nuestros usuarios??



De acuerdo a la Figura 29, la gran mayoría de estudiantes indicaron que esta herramienta sería útil para los usuarios. Este feedback sugiere que la implementación del software podría mejorar significativamente la eficiencia y la efectividad de la informacion.

Figura 42

Pregunta - ¿Cuál cree que sería el nivel de uso del Chatbot?



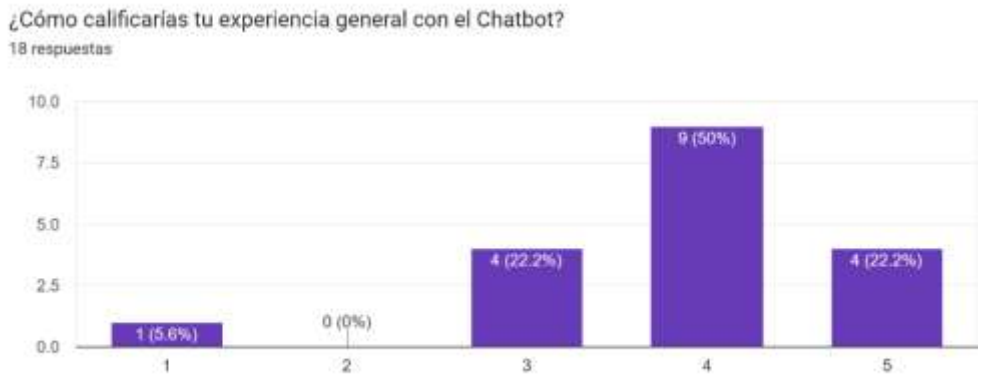
Nota. **1:** Nada frecuente, **5:** Muy frecuente

De acuerdo a la Figura 29, con las estadísticas obtenidas de esta pregunta, se puede concluir que los estudiantes creen que el nivel de uso del chatbot podría no ser muy frecuente. Aunque consideran que la herramienta es útil, perciben que su adopción y uso cotidiano podrían no ser tan altos como se esperaba. Esto sugiere la necesidad de promover más el uso del chatbot y quizás mejorar ciertos aspectos para incrementar su frecuencia de uso entre los estudiantes.

Figura 43

Pregunta - *¿Cuál cree que sería el nivel de uso del Chatbot?*

?



Nota. **1:** Nada frecuente, **5:** Muy frecuente

De acuerdo a la Figura 29, Con los resultados obtenidos en esta pregunta, se puede concluir que los usuarios reportaron una experiencia intermedia con el software. Es importante tener en cuenta que durante la evaluación se les proporcionó una versión beta del software. Por lo tanto, es razonable esperar que sus respuestas mejoren con la entrega de una versión final más pulida y completa. Esta mejora potencial destaca la importancia de continuar refinando el software basado en el feedback recibido para optimizar la experiencia del usuario.

Figura 44

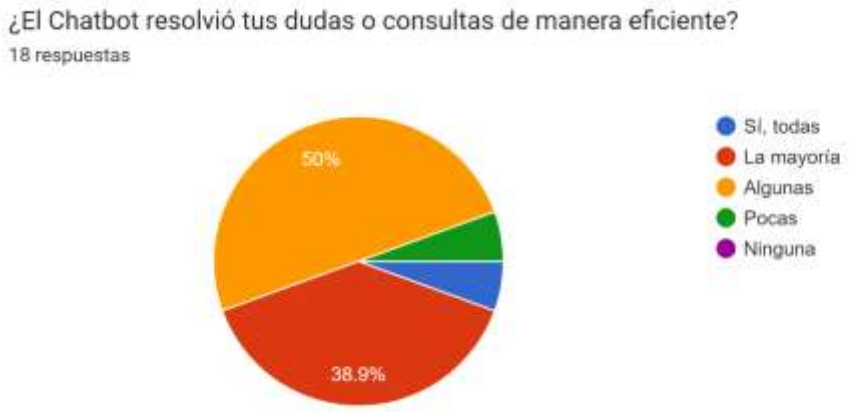
Pregunta - ¿Qué tan fácil fue para ti encontrar la información que buscabas utilizando el Chatbot?



De acuerdo a la Figura 29, los porcentajes proporcionados, concluyen que los estudiantes encuentran fácil de usar el software y no lo perciben como complejo. Este hallazgo sugiere que la interfaz y la usabilidad del software están diseñadas de manera efectiva para facilitar su uso por parte de los usuarios, lo cual es crucial para asegurar una experiencia positiva y productiva con la herramienta.

Figura 45

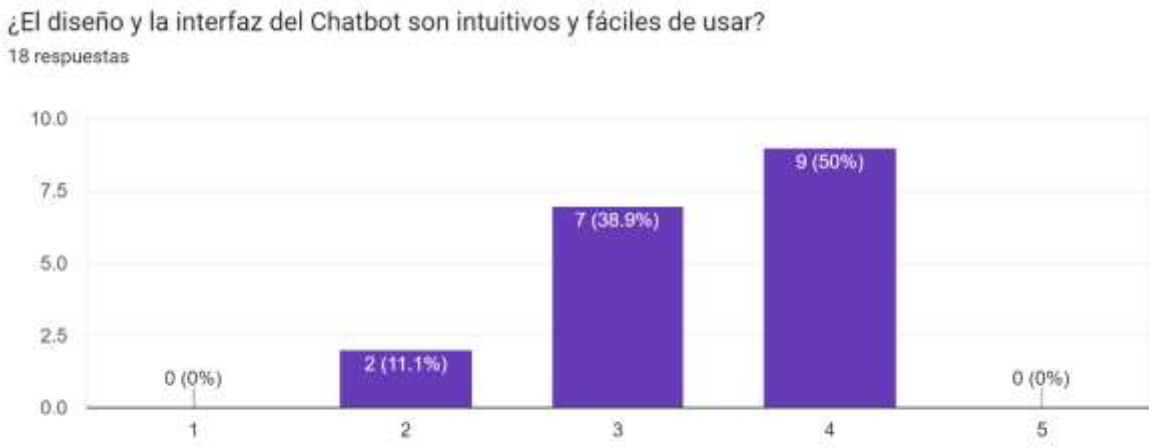
Pregunta - ¿El Chatbot resolvió tus dudas o consultas de manera eficiente?



De acuerdo a la Figura 29, el 50% de los estudiantes, que corresponden a 9 personas, indicaron que obtuvieron respuestas a algunas de las preguntas que hicieron. Además, el 38.9% (7 estudiantes) afirmaron haber recibido respuestas eficientes a la mayoría de sus preguntas. Los demás porcentajes podrían indicar que algunos usuarios podrían no estar familiarizados con la forma de hacer preguntas al software o que aprendieron a formular preguntas más efectivas para obtener respuestas eficientes.

Figura 46

Pregunta - ¿El diseño y la interfaz del Chatbot son intuitivos y fáciles de usar?

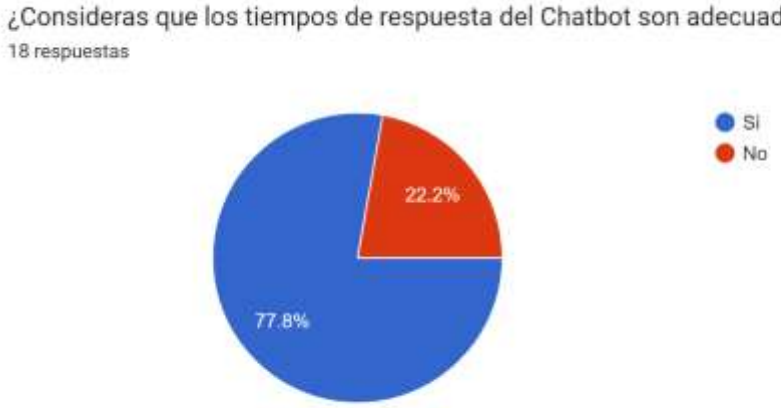


Nota. **1:** totalmente en desacuerdo**, 5:** totalmente de acuerdo

De acuerdo a la Figura 29, la gráfica indica que algunos usuarios perciben que el software no es intuitivo ni fácil de usar. Es importante tener en cuenta que esta percepción se basa en una versión beta del software que se presentó. Es probable que, con futuras iteraciones y mejoras en el diseño y la usabilidad, estas opiniones puedan cambiar favorablemente, mejorando la experiencia general de los usuarios.

Figura 47

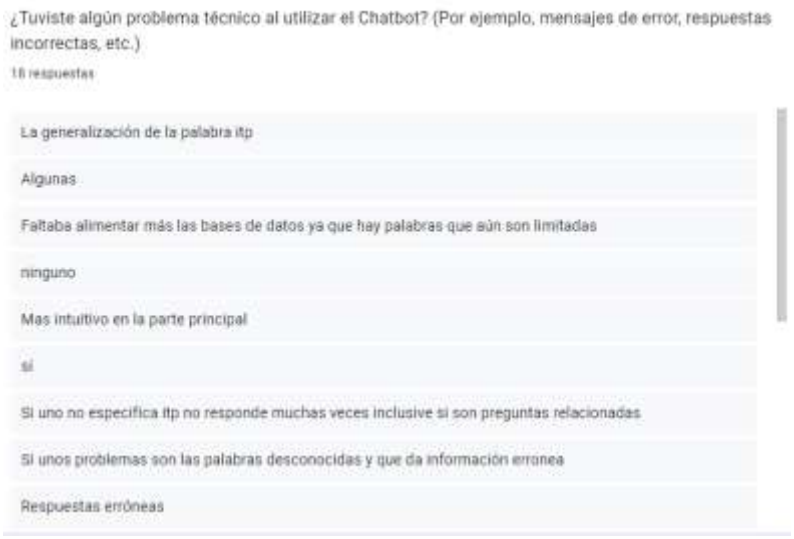
Pregunta - ¿Consideras que los tiempos de respuesta del Chatbot son adecuados?



De acuerdo a la Figura 29, El 77.8% de los estudiantes afirman que el tiempo de respuesta del software es adecuado. Es importante destacar que varios estudiantes estaban utilizando el software simultáneamente, y que un portátil estaba actuando como servidor durante las pruebas. A pesar de estas condiciones, la mayoría de los usuarios consideraron que el tiempo de respuesta cumplía con sus expectativas, lo cual sugiere un rendimiento satisfactorio del sistema bajo carga moderada.

Figura 48

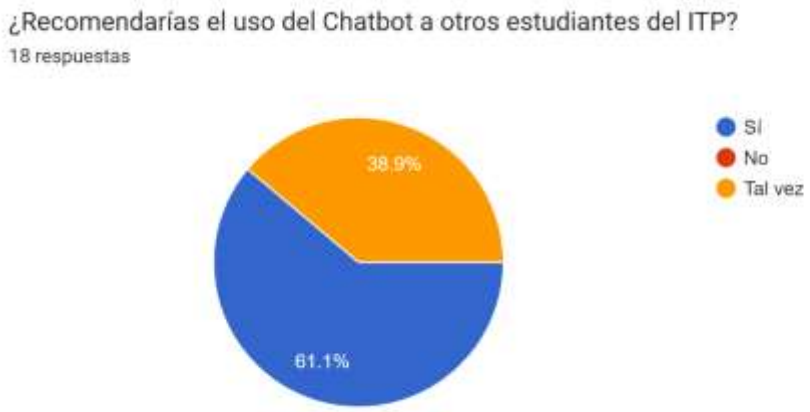
Pregunta - ¿Tuviste algún problema técnico al utilizar el Chatbot? (Por ejemplo, mensajes de error, respuestas incorrectas, etc.)



De acuerdo a la Figura 29, Se solicitó a los encuestados que detallaran los problemas que encontraron al utilizar el software, y se concluyó que la mayoría de los estudiantes no estaban satisfechos con el uso de palabras clave al enviar mensajes, dado que en las pruebas solo se consideraban algunas palabras específicas. Basándonos en esta retroalimentación, se amplió la base de datos para incluir un conjunto más amplio de palabras clave, lo que permitirá al software reconocer y responder a una variedad más amplia de consultas de los usuarios de manera efectiva.

Figura 49

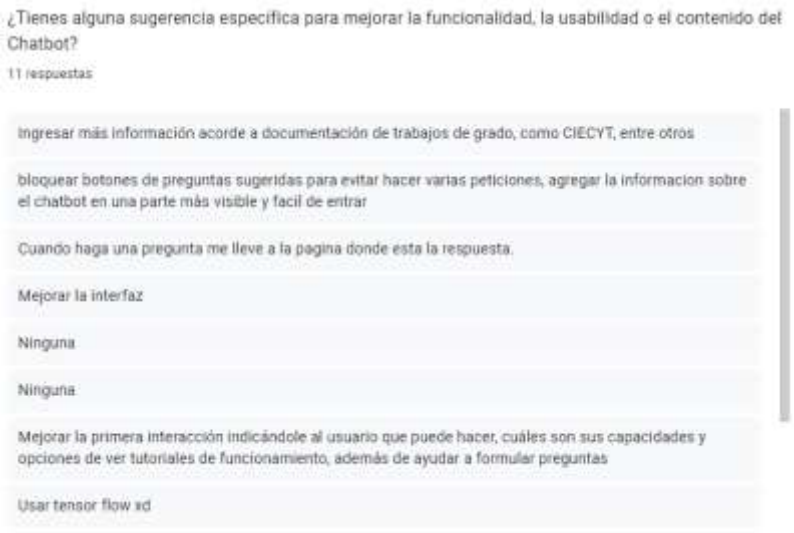
Pregunta - ¿Recomendarías el uso del Chatbot a otros estudiantes del ITP?



De acuerdo a la Figura 29, se puede concluir que los encuestados estarían dispuestos a recomendar el software a otros estudiantes. Esto sugiere que ven valor en esta herramienta y creen que podría ser útil para obtener información del Instituto de una manera efectiva y eficiente.

Figura 50

Pregunta - ¿Tienes alguna sugerencia específica para mejorar la funcionalidad, la usabilidad o el contenido del Chatbot?



De acuerdo a la Figura 29, se les pidió a los encuestados que sugirieran mejoras para el software, lo cual fue fundamental para la mejora de la versión final. Sus aportes y sugerencias jugaron un papel crucial en el proceso de desarrollo, permitiéndonos implementar mejoras significativas en el software.

Capítulo 4: Conclusiones y Trabajo Futuro

1. Conclusiones

¿El uso de un chatbot inteligente implementado en el Instituto Tecnológico del Putumayo (ITP) es de verdadero interés y utilidad para los estudiantes y el personal administrativo?

Durante la evaluación del chatbot, se evidenció que los participantes encontraron la herramienta valiosa y útil para obtener información rápida y precisa sobre los servicios y documentos institucionales (Estatutos). La accesibilidad y la capacidad del chatbot para responder a consultas en tiempo real demostraron ser beneficiosas, mejorando significativamente la eficiencia en la gestión de información.

Los participantes de la evaluación, que incluían estudiantes de 8vo y 10mo semestre de la carrera de Ingeniería de Sistemas, expresaron una alta satisfacción con la usabilidad y funcionalidad del chatbot. Los resultados del "Formulario de Usabilidad del Chatbot" indicaron que la mayoría de los usuarios encontraron la interfaz intuitiva y fácil de usar, y las respuestas proporcionadas por el chatbot fueron precisas y útiles.

Además, el análisis de los comentarios y opiniones detalladas de los usuarios permitió identificar áreas de mejora que fueron implementadas para ajustar el software según las necesidades y expectativas de los usuarios. Este proceso de retroalimentación continua es esencial para asegurar que la herramienta evolucione y mejore en el tiempo.

El uso del chatbot también contribuyó al cumplimiento de los objetivos educativos establecidos, proporcionando una herramienta tecnológica efectiva que facilita el acceso a la información del instituto y promueve la autonomía de los estudiantes. La integración de la API de OpenAI para el procesamiento del lenguaje natural, fue un factor clave en la capacidad del chatbot para manejar consultas complejas y proporcionar respuestas contextuales.

En base a lo anterior se puede afirmar, que el desarrollo e implementación del chatbot en el ITP ha demostrado ser de verdadero interés y utilidad para los estudiantes, el personal administrativo y la comunidad en general. La herramienta no solo mejora la eficiencia en la gestión de información, sino que también enriquece la experiencia educativa y administrativa, alineándose con las necesidades y objetivos del Instituto Tecnológico del Putumayo.

2. Trabajo Futuro

El desarrollo del chatbot ha demostrado ser una herramienta de utilidad para la comunidad del Instituto Tecnológico del Putumayo (ITP), pero hay muchas oportunidades para mejorar y expandir sus funcionalidades a futuro. Estas mejoras no solo reducirán costos, sino que también aumentarán la eficacia y la accesibilidad del chatbot. A continuación, se detallan algunas de las áreas en las que se podría trabajar:

- Integración de Modelos de IA Open Source: Actualmente, el chatbot utiliza la API de OpenAI, que conlleva costos asociados. Una solución a futuro es integrar modelos de inteligencia artificial locales y de código abierto, como LLaMA 2, GPT-4all, o Phi-3, los cuales pueden ser implementados y optimizados para trabajar sin necesidad de una conexión constante a servicios de terceros. Esta integración podría reducir significativamente los costos operativos a largo plazo.
- Optimización del Uso de API: Desarrollar estrategias para optimizar el uso de las API externas, limitando las llamadas innecesarias y aprovechando al máximo las respuestas en caché, también puede ayudar a reducir los costos.

- Implementación de Árboles Binarios: La incorporación de estructuras de datos como árboles binarios puede mejorar la eficiencia del procesamiento de las consultas del chatbot. Esto puede facilitar una búsqueda y recuperación de información más rápida y estructurada.
- Interfaz de Usuario Mejorada: Continuar mejorando la interfaz de usuario para que sea más intuitiva y fácil de usar. Esto incluye la implementación de accesibilidad para usuarios con discapacidades.
- Migración a Infraestructura Escalable: Considerar la migración del chatbot a una infraestructura más escalable, como servicios en la nube, que puedan manejar un mayor número de usuarios simultáneos y proporcionar redundancia y alta disponibilidad.

Detalles del Sprint 4

- **Duración:** 4 semanas
- **Estado:** Completado

Capítulo 5: Gestión del Proyecto

3. Presupuesto

La realización del trabajo implica el uso de una serie de recursos humanos, materiales, económicos etc. mostrados en la siguiente tabla.

Tabla 6

Recursos y Presupuesto

RUBROS	DESCRIPCIÓN	JUSTIFICACIÓN	CANTIDAD	VALOR UNITARIO	VALOR TOTAL	CONTRAPARTIDA		ENTIDAD FINANCIADORA	TOTAL
						ESPECIE	DINERO		
EQUIPOS Y SOFTWARE	Computador portátil. Api key de OpenAI. Máquina virtual ITP	Desarrollo del software. Integración de IA al chatbot. Alojamiento del chatbot en el ITP			4’000.000 0 0				’500.000
PERSONAL	Desarrollador Diseñador	Back end Front tend			0 0				
VIAJES									
SALIDAS DE CAMPO	Visita al ITP	Evaluación del software			0				
MATERIALES Y SUMINISTROS									
SERVICIOS TÉCNICOS									
PUBLICACIONES Y PATENTES									
MATERIAL									

BIBLIOGRÁFICO									
CONSTRUCCIONES									
MANTENIMIENTO									
ADMINISTRACIÓN									
TOTAL						**4’500.000**			

4. Planificación del Proyecto

La siguiente tabla muestra la división de tareas por cada objetivo propuesto y la duración estimada de cada una de ellas.

Tabla 7

Cronograma de actividades

° Obj.	**Actividad**	**AÑO 2023 -2024**										
		go.	**ep.**	**ct.**	**ov.**	**ic.**	**ne.**	**eb.**	**ar.**	**br.**	**ay.**	**un.**
	Búsqueda de la información											
	Distribución de la información											
	Análisis de la información											
	Documentar hallazgos y puntos clave											
	Definir los requisitos funcionales y no funcionales del chatbot.											
	Diseñar la arquitectura general del chatbot y su interfaz de usuario.											
	Establecer las capacidades de manejo de consultas y la lógica de respuesta.											
	Crear prototipos de las interfaces de usuario para las interacciones del chatbot.											

	Realizar pruebas exhaustivas en el entorno de prueba, corrigiendo errores y optimizando el rendimiento del chatbot.											
	Recolectar datos de prueba y ajustar los modelos de procesamiento de lenguaje natural para mejorar la precisión											
	Realizar pruebas en un entorno semi realista para simular interacciones reales.											
	Recopilar retroalimentación de los usuarios y realizar ajustes en la funcionalidad del chatbot.											
	Realizar los ajustes finales en base a la retroalimentación de los usuarios.											
	Generar documentación detallada, incluyendo manuales de usuario											

Referencias

Abeliuk, A., & Gutiérrez, C. (2021). *Inteligencia Artificial El primer programa de IA*.

Álvarez, O., Alvarado, K., & Sánchez, K. (2018). El valor de la Inteligencia Artificial en el Servicio al Cliente EciBot. *LACCEI International Multi-Conference for Engineering, Education, and Technology: «innovation in Education and Inclusion»*, *July*, 19-21. http://www.laccei.org/LACCEI2018-Lima/work_in_progress/WP301.pdf

Anjali Chauhan. (2019). A Review on Various Aspects of MongoDb Databases. *International Journal of Engineering Research & Technology (IJERT)*. https://d1wqtxts1xzle7.cloudfront.net/60661268/a-review-on-various-aspects-of-mongodb-databases-IJERTV8IS05003120190921-10051-a8lu8n-libre.pdf?1569063684=&response-content-disposition=inline%3B+filename%3DIJERT_A_Review_on_Various_Aspects_of_Mon.pdf&Expires=1718068016&Signature=A7JSPqDJhB9NtWtBpmxYhpx5Y5pa1h2KP3UEgz4ivWqCb~95PhsiJG-LLjVvkORaXBuADBQh~tbVT2tK6X316Zi-PKtmRWCcGh4ERqXf0vkTotLULii~cADyNteRG7~hvrEIJZrMDrr1WmOBGxMqhaqAzXObE1dVnUKANyvs7akV8uXvi-viW2~ao4y~SyRsUEzmTXg9G64EVOA0wUctbbNEqD7PxB6r~GAhR-MBug1bzOVzUNUpgRFMqPNJ4epNdYnqAygRe1AeBgJk1aqYcHfjRetzm4OxkhRvQx4uDfz3I1E3BvlcOoWrNOW3g6hX5qcTkxxHNIXfG~Qk-c3BFg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020). A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3386723.3387897

Aydin, Ö., & Karaarslan, E. (2022). *OpenAI ChatGPT Generated Literature Review: Digital Twin in Healthcare*. https://ssrn.com/abstract=4308687

Cesar, J., Becerra, A., Elena, C., & Vanegas, D. (2018). Propuesta de un método para desarrollar Sistemas de Información Geográfica a partir de la metodología de desarrollo ágil - SCRUM. *Cuaderno activa*, *10*, 29-41. https://doi.org/10.53995/20278101.490

Estupiñán Ricardo, J., Yelandi Leyva Vázquez, M., Javier Peñafiel Palacios, A., El Assafiri Ojeda, Y., Ricardo, E., Vázquez, L., Palacios, P., & Assafiri Ojeda, E. (2021). Inteligencia artificial y propiedad intelectual. *Universidad y Sociedad*, *13*(S3), 362-368. https://rus.ucf.edu.cu/index.php/rus/article/view/2490

Garibay, F. (2020). Diseño E Implementación De Un Asistente Virtual (Chatbot) Para Ofrecer Atención a Los Clientes De Una Aerolínea Mexicana Por Medio De Sus Canales Conversacionales. *Infotec Posgrados*, 63. https://infotec.repositorioinstitucional.mx/jspui/handle/1027/402

Henriquez Miranda, C., Rios Perez, J. D., & Sanchez-Torres, G. (2024). Toward the improvement of teaching in object-oriented programming: the integration of intelligent chatbot assistance and professor Alex's implementation. *REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA)*, *1*(43), 134-143. https://doi.org/10.24054/rcta.v1i43.2803

Laguia, D., Hallar, K., Sofía, O., González, L., & Gesto, E. (2022). Especificación de requisitos de un sistema IoT con UML. *Informes Científicos Técnicos - UNPA*, *14*(2), 200-215. https://doi.org/10.22305/ict-unpa.v14.n2.887

Monroy, C. R. (2022). El Lenguaje Python Y Su Potencial En El Desarrollo De Software De Inteligencia Artificial. *Masferrer Investiga: Revista Científica de la Universidad Salvadoreña Alberto Masferrer*, *12*(1), 18-42.

Morales González, B., & Morales González, B. (2022). Diseño instruccional según el modelo ADDIE en la formación inicial docente. *Apertura (Guadalajara, Jal.)*, *14*(1), 80-95. https://doi.org/10.32870/AP.V14N1.2160

Moral-Sánchez, S. N., Rey, F. J. R., & Cebrián-De-la-serna, M. (2023). Analysis of artificial intelligence chatbots and satisfaction for learning in mathematics education. *International Journal of Educational Research and Innovation*, *2023*(20). https://doi.org/10.46661/ijeri.8196

Múnera Torres, M. T., Salazar Álvarez, L. M., & Osorio Osorio, A. S. (2022). Estudio inicial de un chatbot para estudiantes de la modalidad virtual de la Escuela Interamericana de Bibliotecología. *Investigación bibliotecológica*, *36*(90), 13-30. https://doi.org/10.22201/IIBI.24488321XE.2022.90.58452

Musthafa, S. A., & Kumar, P. S. (2023). Evaluation of Agrometeorology Using Bootstrap Aggregation with Streamlit. *International Conference on Recent Trends in Data Science and its Applications*. https://www.riverpublishers.com/pdf/ebook/chapter/RP_9788770040723C84.pdf

Ocaña-Fernandez, Y., Valenzuela-Fernandez, L., & Garro-Aburto, L. (2019). Inteligencia artificial y sus implicaciones en la educación superior. *Propósitos y Representaciones*, *7*(2), 536-568. https://doi.org/10.20511/PYR2019.V7N2.274

Peña-Torres, J. A., Giraldo-Alegría, S., Arango-Pastrana, C. A., & Bucheli, V. A. (2021). Un chatbot para asistir a las necesidades de información en tiempos de COVID-19. *Ingeniería Y Competitividad*, *24*(1), 1-17. https://doi.org/10.25100/iyc.v24i1.11004

Pokhrel, S., Ganesan, S., Akther, T., & Karunarathne, L. (2024). Building Customized Chatbots for Document Summarization and Question Answering using Large Language Models using a Framework with OpenAI, Lang chain, and Streamlit. *Journal of Information Technology and Digital World*, *6*(1), 70-86. https://doi.org/10.36548/JITDW.2024.1.006

Romero, M., Casadevante, C., & Montoro, H. (2020). Cómo construir un psicólogo-chatbot. *Papeles del Psicólogo*, *41*(1), 27-34. https://doi.org/10.23923/PAP.PSICOL2020.2920

Shah, N., Willick, D., & Mago, V. (2022). A framework for social media data analytics using Elasticsearch and Kibana. *Wireless Networks*. https://link.springer.com/article/10.1007/s11276-018-01896-2

Vidal Ledo, M. J., González, A. M., & Valdés Santiago, D. (2019). Inteligencia artificial en la docencia médica. *Scielo*. http://scielo.sld.cu/scielo.php?pid=S0864-21412019000300014&script=sci_arttext

APÉNDICES

5. Apéndice A. Autorización de Implementación – En el ITP



MOCOA,12-06-2024

Señor(a):
John Andrés Ceron
Vicerrector Administrativo

Cordial saludo:

Mediante este oficio, nosotros como estudiantes activos de ingeniería de sistemas y aspirantes al título profesional, nos dirigimos a usted, para solicitar su autorización de poder hacer uso de la infraestructura tecnológica del itp, con el fin de implementar un chatbot inteligente especializado en el procesamiento de documentos, que se obtuvo como resultado de una investigación de tesis que fue asesorada por el Doctor Jhoni Ricardo Cerón Chaves.

Este trabajo de investigación tiene la intención de mejorar la accesibilidad a la información pública del ITP contenida en documentos como los estatutos. Asimismo, busca reducir la carga del personal administrativo al atender algunas de las consultas de los estudiantes que actualmente se realizan de manera presencial. Además, pretendemos aportar al ámbito de la innovación y modernización, haciendo uso de tecnologías avanzadas como la inteligencia artificial.

Le agradecemos por la atención que pueda brindarle a esta solicitud.

Atentamente:

ESTUDIANTES		
NOMBRE Y APELLIDOS	**No. de Cédula**	
Yan Carlos Cuaran Imbacuan	**C.C. 1006789060**	Yan Carlos Cuaran Imbacuan
Maicol Libardo Jossa Campaña	**C.C. 1006816052**	Maicol Jossa

Firma vicerrector administrativo

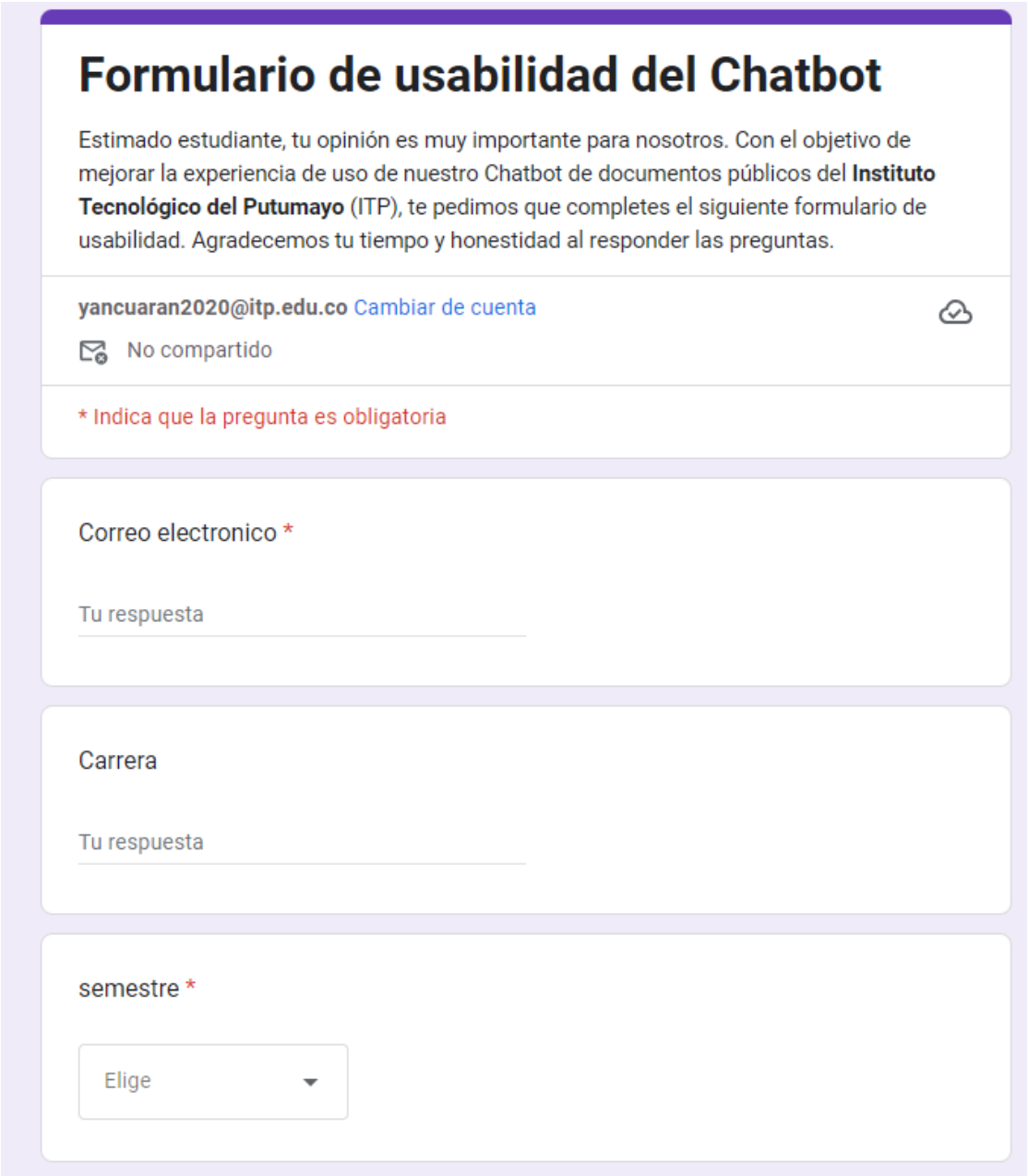
Con copia al Ing. Fernando Guzmán.

Nit 800.247.540-1 Sede Mocoa: "Alto Libre" Barrio Luis Carlos Galan Telefonos: 4296105 - 4200822 – 4201206
Subsede Sibundoy: Vía al Canal C – Granja Versalles Teléfono: 310 243 4685
Email: atencionalusuario@itp.edu.co
www.itp.edu.co

ANEXOS

6. Anexo 1. Formulario de Usabilidad del Chatbot

Datos de los participantes de la evolución del chatbot.



Formulario de usabilidad del Chatbot

Estimado estudiante, tu opinión es muy importante para nosotros. Con el objetivo de mejorar la experiencia de uso de nuestro Chatbot de documentos públicos del **Instituto Tecnológico del Putumayo** (ITP), te pedimos que completes el siguiente formulario de usabilidad. Agradecemos tu tiempo y honestidad al responder las preguntas.

yancuaran2020@itp.edu.co Cambiar de cuenta

No compartido

* Indica que la pregunta es obligatoria

Correo electronico *

Tu respuesta

Carrera

Tu respuesta

semestre *

Elige

Preguntas sobre el uso de herramientas, semejantes.

¿Ha tenido interacción con algún tipo de Chatbot o herramienta de respuesta automatizada? *

- si
- no

Si en la pregunta anterior su respuesta fue sí, por favor, cuéntenos como ha sido su experiencia con chatbots

Tu respuesta

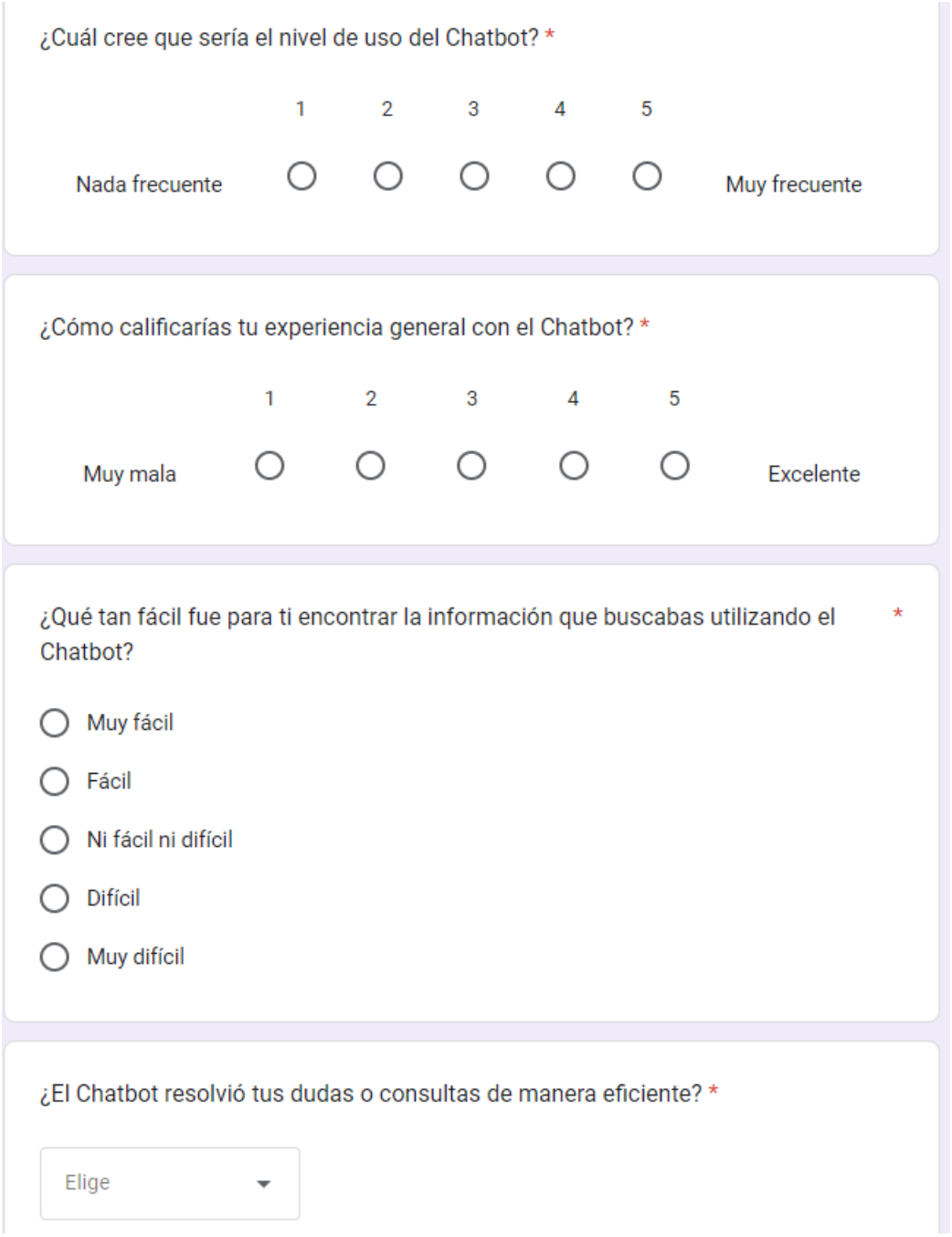
¿Considera que esta herramienta optimiza el trabajo a los administrativos? *

- Sí
- No
- Tal vez

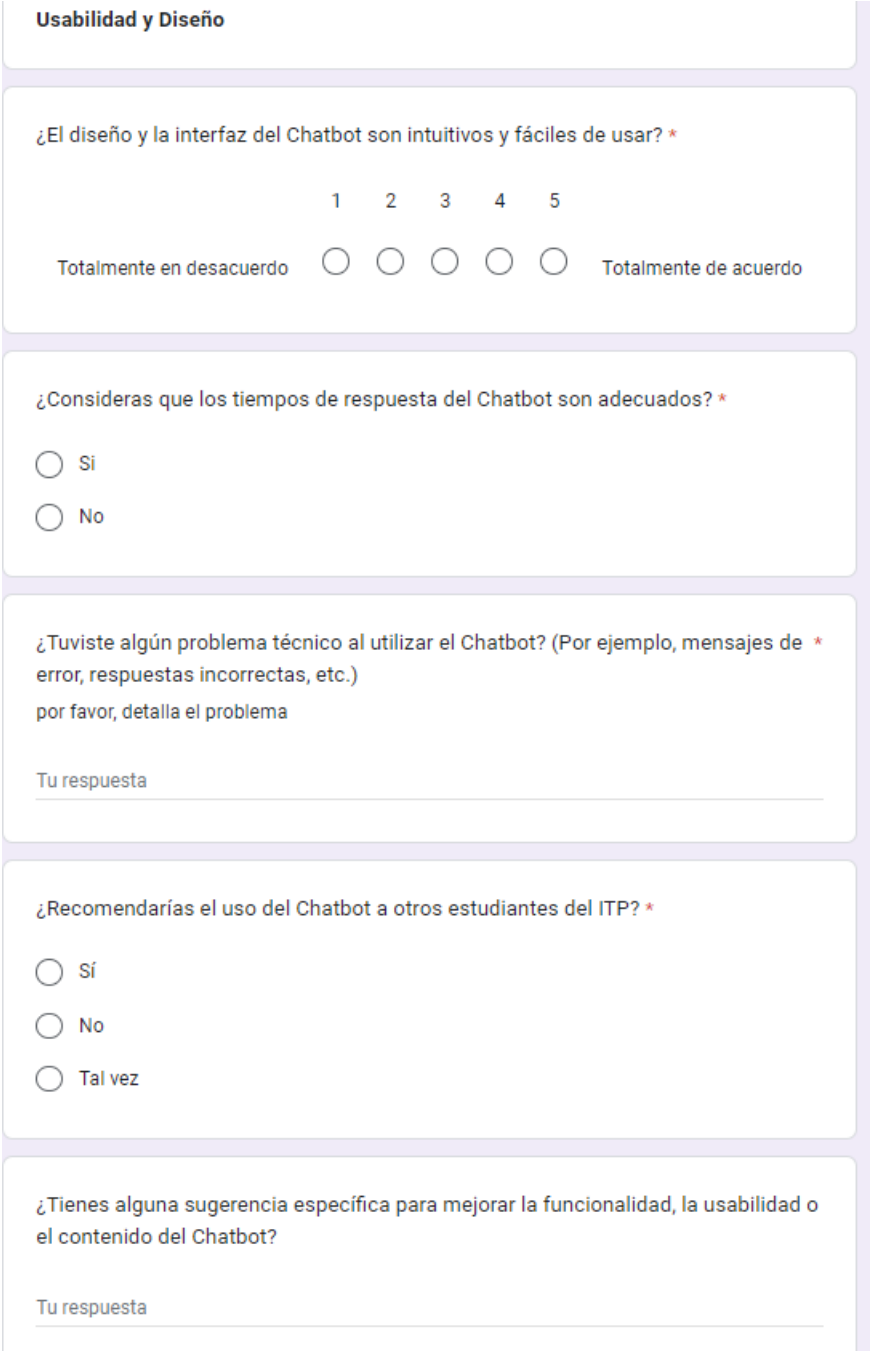
¿Considera que esta herramienta seria de gran utilidad para nuestros usuarios? *

- Sí
- No
- Tal vez

Preguntas sobre la experiencia, que obtuvieron los participantes durante las pruebas del software.



Sección de usabilidad y diseño, en esta sección se realizaron unas preguntas que involucran, características muy importantes sobre el software como, la interfaz del chatbot, tiempos de respuesta, problemas presentes durante la ejecución y sugerencias para el mejoramiento del software.



Usabilidad y Diseño

¿El diseño y la interfaz del Chatbot son intuitivos y fáciles de usar? *

	1	2	3	4	5	
Totalmente en desacuerdo	○	○	○	○	○	Totalmente de acuerdo

¿Consideras que los tiempos de respuesta del Chatbot son adecuados? *

- ○ Si
- ○ No

¿Tuviste algún problema técnico al utilizar el Chatbot? (Por ejemplo, mensajes de error, respuestas incorrectas, etc.) *

por favor, detalla el problema

Tu respuesta

¿Recomendarías el uso del Chatbot a otros estudiantes del ITP? *

- ○ Sí
- ○ No
- ○ Tal vez

¿Tienes alguna sugerencia específica para mejorar la funcionalidad, la usabilidad o el contenido del Chatbot?

Tu respuesta

7. Anexo 2. Resultados Revisión Sistemática

PRESENTACIÓN DE LA PROPUESTA

Título de la propuesta

Desarrollo e Implementación de un Chatbot con Inteligencia artificial, en el Instituto Tecnológico del Putumayo, para la extracción y respuesta automática a partir de documentos.

Objetivos

Objetivo General

Diseñar, desarrollar e implementar un chatbot con inteligencia artificial que facilite el acceso a la información relevante del instituto tecnológico, mediante un sistema de preguntas de manera escrita y verbal, con el fin de mejorar la experiencia de la comunidad académica y fomentar la autonomía.

Objetivos Específicos

- Realizar una revisión de literatura sobre la implementación de chatbots con inteligencia artificial, específicamente en el manejo de documentos, con el fin de comprender las mejores prácticas, desafíos y beneficios asociados a su uso.
- Diseñar un chatbot que sea capaz de manejar consultas de forma precisa y eficiente.
- Desarrollar e implementar el chatbot en un entorno de prueba, asegurando su correcta funcionalidad y compatibilidad con los sistemas actuales del instituto tecnológico del putumayo.
- Evaluar la eficiencia del chatbot en la provisión de información relevante a la comunidad educativa, y cómo facilita su acceso a la misma.

Hipótesis

La implementación de un chatbot en el entorno institucional facilitará el acceso a la información relevante, proporcionando respuestas rápidas y precisas a las consultas de la comunidad

educativa, y, en consecuencia, mejorando la experiencia de quienes buscan información sobre el Instituto Tecnológico del Putumayo.

Estado de la propuesta: Aprobada.

Objetivo general de la revisión Sistemática:

Investigar y analizar la literatura sobre la eficiencia y eficacia de chatbots con inteligencia artificial, con un enfoque especifico en el manejo de informacion y documentos.

Objetivos específicos de la revisión Sistemática:

- Identificar estudios, artículos y documentos que describan o evalúen la Implementación y desempeño de chatbots con inteligencia artificial, específicamente en lo que respecta al manejo de informacion y documentos.
- Analizar las mejores prácticas, desafíos y beneficios asociados con la implementación de chatbots, para extraer recomendaciones que puedan aplicarse a la página principal del instituto tecnológico del putumayo.

Criterios de inclusión:

- Estudios publicados en revistas científicas.
- Trabajos centrados en chatbots que utilizan algún tipo de inteligencia artificial.
- Publicaciones que incluyan evaluaciones de desempeño, eficiencia, o eficacia de chatbots en el manejo de información.
- Estudios publicados en los últimos 5 años para asegurar la relevancia tecnológica.
- Trabajos en idiomas inglés y español para facilitar la interpretación de resultados.

Criterios de exclusión:

- Estudios publicados hace más de 5 años.
- Trabajos en idiomas diferentes al inglés y al español.
- Documentos que traten el tema de inteligencia artificial sin especificar su relación o aplicación en chatbots.
- Publicaciones que no tengan referencias bibliográficas.

Fuentes de Búsqueda:

- Bases de datos académicas (PubMed, IEEE, Scopus, Web of Science)
- Google Scholar
- Repositorios institucionales o bases de datos de tesis
- Bibliotecas digitales específicas en el campo de la informática o la inteligencia artificial.

Estrategia de Búsqueda:

Para lograr esto se es necesario realizar una combinación de palabras clave, a continuación, algunas de las combinaciones que podríamos utilizar:

Básicas:

- "Chatbots" AND "Manejo de Información" AND "Desempeño" AND "Inteligencia Artificial"

Documentos encontrados: 77

- "Beneficios" AND "chatbots" AND "Implementación" AND "Manejo de Informacion" AND "Inteligencia Artificial"

Documentos encontrados: 103

- "Mejores prácticas" AND "chatbots" AND "Manejo de información" AND "Inteligencia Artificial"

Documentos encontrados: 23

Mas especificas (Artículos):

- "Implementación" AND "Chatbots" AND "Inteligencia Artificial" AND "Eficacia"

Documentos encontrados: 24

- "Best Practices" AND "chatbots" AND "Information Management" AND "Artificial Intelligence"

Documentos encontrados: 123

PREGUNTAS DE INVESTIGACIÓN

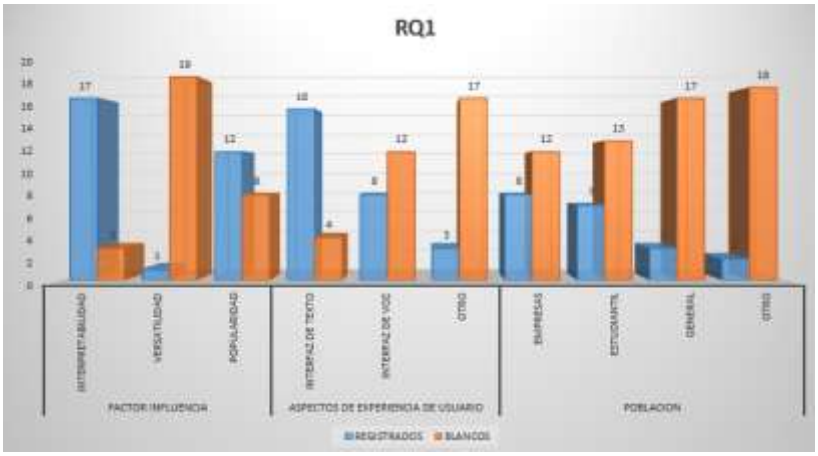
- ¿Cómo los chatbots con inteligencia artificial han contribuido a mejorar el acceso a la información y la experiencia del usuario?
- ¿Qué algoritmos se han destacado en la literatura para la creación de los chatbots con inteligencia artificial y han demostrado ser los más eficaces?
- ¿Qué metodologías y herramientas se han utilizado con éxito para implementar chatbots con inteligencia artificial en sistemas ya existentes, en diferentes campos?
- ¿En qué campos, aparte de la educación, se han implementado chatbots con inteligencia artificial con éxito?
- ¿Qué modelos avanzados de procesamiento de lenguaje natural en chatbots, han demostrado ser los más eficientes y eficaces en su uso?

RESULTADOS DE LA REVISIÓN SISTEMÁTICA

En esta parte se describen los hallazgos encontrados según cada pregunta de investigación, se muestran los resultados obtenidos de la codificación realizada, considerando las categorías establecidas y subcategorías definidas en la sección de planificación de la revisión con respecto a cada subpregunta de investigación.

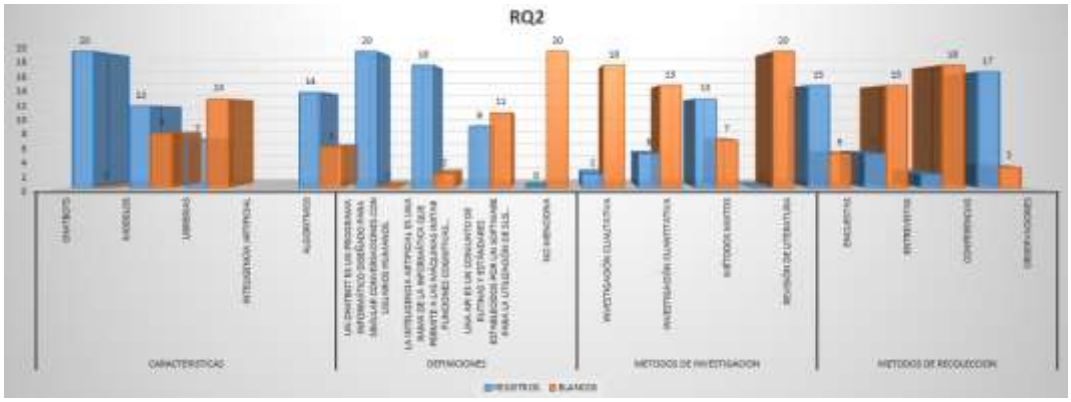
Se realizo la revisión de literatura a 20 artículos relacionados al tema principal de investigación, los resultados fueron los siguientes:

RQ1: ¿Cómo los chatbots con inteligencia artificial han contribuido a mejorar el acceso a la información y la experiencia del usuario?



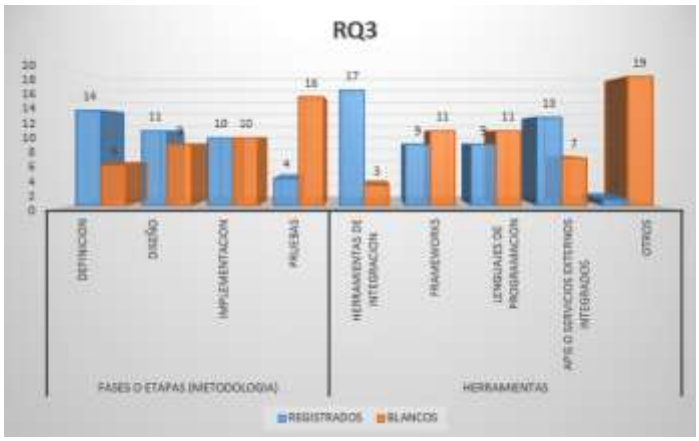
Los chatbots con inteligencia artificial han mejorado significativamente el acceso a la información y la experiencia del usuario, siendo la interpretabilidad el factor más valorado, con un 85% de relevancia en las fuentes analizadas, lo que sugiere que la claridad con la que los chatbots presentan la información es precisa. Además, su popularidad, indicada por el 60% de las menciones, refleja la efectividad percibida de los chatbots en la práctica. Las interfaces de texto son especialmente significativas, siendo preferidas en el 80% de los casos, lo que denota su importancia en la interacción eficiente con los usuarios. Las empresas y la población estudiantil son identificadas como los principales beneficiarios, con un 40% y un 35% respectivamente, lo que indica la utilidad de los chatbots en entornos empresariales y educativos. Por tanto, podemos afirmar que la eficacia, la claridad y la capacidad de adaptación de los chatbots son clave para enriquecer el acceso a la información y la experiencia del usuario.

RQ2: ¿Qué algoritmos se han destacado en la literatura para la creación de los chatbots con inteligencia artificial y han demostrado ser los más eficaces?



La literatura indica que los modelos y algoritmos son fundamentales en la creación de chatbots con inteligencia artificial eficaces, con una mención en el 100% y 70% de los registros respectivamente. Específicamente, los algoritmos basados en reglas, aprendizaje automático y redes neuronales son esenciales, siendo los basados en aprendizaje automático los más utilizados. La inteligencia artificial es un campo destacado en todas las fuentes revisadas, enfatizado en el 70% de ellas, subrayando su importancia para simular conversaciones humanas de manera convincente. Aunque las librerías son menos mencionadas, con un 35% de relevancia, siguen siendo componentes críticos en la construcción de chatbots. Los métodos de investigación cualitativa y las revisiones de literatura contribuyen en un 65% y 25% respectivamente, mostrando que son enfoques significativos para evaluar la eficacia de los chatbots. Finalmente, las observaciones, con una relevancia del 85%, resaltan como el método de recolección más prominente, demostrando que la evaluación directa es clave para determinar la eficacia de los chatbots en entornos reales. Esto corrobora la idea de que la combinación adecuada de modelos, algoritmos y librerías, junto con una evaluación meticulosa, es esencial para optimizar la eficiencia y eficacia de los chatbots con inteligencia artificial.

RQ3: ¿Qué metodologías y herramientas se han utilizado con éxito para implementar chatbots con inteligencia artificial en sistemas ya existentes, en diferentes campos?



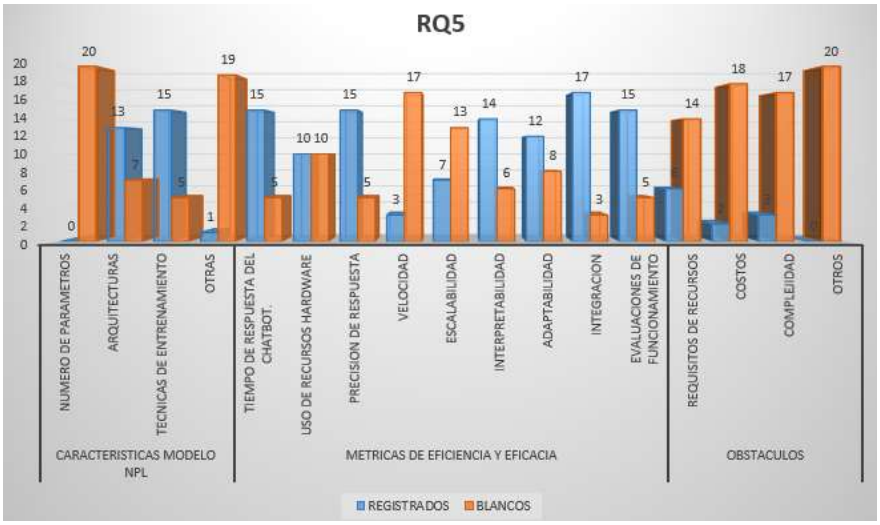
Las metodologías ágiles y las guiadas por pruebas son destacadas en la literatura para la implementación exitosa de chatbots con inteligencia artificial en diferentes campos, con la metodología ágil mencionada en la mayoría de los artículos revisados. La fase de pruebas es especialmente crítica, evidenciada en el 85% de las fuentes, asegurando que el chatbot funcione como se espera. Herramientas como Microsoft Bot Framework y Dialogflow, siendo este último más utilizado, facilitan la implementación eficaz de chatbots. Las fases de definición y diseño también son importantes, con una mención en el 70% y 55% respectivamente, para establecer objetivos claros y planificar adecuadamente. Las herramientas de integración y las APIs o servicios externos, mencionados en el 65% de las fuentes, son fundamentales para la adaptabilidad y funcionalidad en distintos entornos. Estas metodologías y herramientas han demostrado su eficacia en la creación y optimización de chatbots, facilitando su implementación en una amplia gama de aplicaciones.

RQ4: ¿En qué campos, aparte de la educación, se han implementado chatbots con inteligencia artificial con éxito?



Los chatbots con inteligencia artificial han logrado inclusiones significativas en diversos campos más allá de la educación, evidenciando éxito en campos como la salud con un 45%, las finanzas 35% y el transporte 35% que constituyen las aplicaciones e chatbots en estos campos. Los chatbots facilitan tareas que van desde guiar a pacientes, resolver dudas financieras, hasta ayudar con la planificación de viajes. Destacan especialmente en servicios de atención al cliente, donde han sido adoptados en un 75% de los casos, mejorando notablemente cómo las empresas interactúan con sus usuarios. Aunque menos frecuentes en roles como tutoría virtual 20% y soporte técnico 30%, su implementación es efectiva y promete expansión y adaptabilidad. La ausencia de registros que indican inviabilidad destaca la efectividad de los chatbots lo que evidencia su consolidación como herramientas indispensables en múltiples campos.

RQ5: ¿Qué modelos avanzados de procesamiento de lenguaje natural en chatbots, han demostrado ser los más eficientes y eficaces en su uso?



Los avances en los chatbots con procesamiento de lenguaje natural han demostrado que tanto las arquitecturas de los modelos como las técnicas de entrenamiento son fundamentales para su éxito, destacando en el 65% y el 75% de los registros analizados, respectivamente. Las métricas de eficiencia como el tiempo de respuesta y la precisión de la misma son críticas, mencionadas cada una en el 75% de los estudios, mientras que el uso eficiente de los recursos hardware se resalta en el 50% de ellos. La adaptabilidad y la integración son también altamente valoradas, con una mención en el 60% y el 85% respectivamente, enfatizando la importancia de la flexibilidad y la capacidad de los chatbots para integrarse con otros sistemas. A pesar de los desafíos presentados por los requisitos de recursos y la complejidad, señalados en el 30% y el 15% de los registros, respectivamente, modelos prominentes como GPT de OpenAI, BERT de Google Cloud Natural Language API y los servicios cognitivos de Azure han sido frecuentemente citados como ejemplos eficaces en la aplicación de chatbots. Estos modelos, a menudo accesibles a través de APIs, facilitan la implementación de soluciones de NLP potentes y versátiles. La literatura sugiere que una combinación cuidadosa de una arquitectura sólida, técnicas de entrenamiento optimizadas y una consideración meticulosa de las métricas de rendimiento es lo que conduce a chatbots altamente eficientes y eficaces en entornos reales.

DISCUSIÓN

La inteligencia artificial, de la mano con los chatbots han revolucionado el acceso a la información y la experiencia del usuario, siendo su capacidad para brindar información de manera clara e interpretable un factor crucial. Este aspecto, con un 85% de relevancia en las fuentes analizadas, destaca la importancia de la claridad en la comunicación de los chatbots. Su popularidad y efectividad se reflejan en su uso extendido, especialmente en interfaces de texto, preferidas en el 80% de los casos. Esto subraya su rol esencial en la interacción con los usuarios en contextos variados, como los empresariales y educativos. Por otro lado, la literatura resalta la importancia de algoritmos basados en reglas y aprendizaje automático para la creación de chatbots efectivos, con una mención significativa en los artículos analizados. Estos algoritmos, junto con la inteligencia artificial, son fundamentales para simular conversaciones humanas de manera convincente, lo cual es crucial en la implementación de chatbots en campos variados como la salud, las finanzas y el transporte, donde su adaptabilidad y funcionalidad se han convertido en herramientas muy útiles.

La metodología ágil y las pruebas guiadas son esenciales en la implementación exitosa de chatbots en sistemas existentes, facilitando su adaptación en una amplia gama de aplicaciones. Herramientas como Microsoft Bot Framework y Dialogflow han sido destacadas por su eficacia en este proceso, subrayando la importancia de las fases de definición y diseño. Además, la literatura sugiere que los modelos avanzados de procesamiento de lenguaje natural, como GPT de OpenAI y BERT de Google, son altamente eficientes y efectivos, lo que se refleja en su amplia aplicación en entornos reales. Estos modelos destacan por su capacidad para manejar de manera eficiente el tiempo de respuesta y la precisión, siendo fundamentales en la integración de chatbots con otros sistemas. Así, queda claro que la combinación de tecnologías avanzadas y una cuidadosa planificación son clave para optimizar la eficiencia y eficacia de los chatbots en el panorama actual de la inteligencia artificial, además brinda una visión más

amplia del uso que se les pueden llegar a dar a los chatbots, por su fácil adaptabilidad en diferentes situaciones y entornos.

OBSERVACIONES

La revisión sistemática se realizó en la asignatura Electiva Complementaría II (Seminario de Investigacion), fue de gran ayuda para el avancé de nuestra investigación, en el sentido que fue de guía que permitió, realizar el proceso correcto que se debe llevar a cabo, cuando se realiza una investigación. A demás todo lo aprendido se debió al docente a cargo de la clase, el docente Jairo Quintero, el cual aplico una metodología de enseñanza, que hizo que la clase fuera de gran interés.

Artículos estudiados:

https://docs.google.com/spreadsheets/d/14pNo_a8h1SPuAUQ6q9ingB1mOjg6DIkW/edit?usp=sharing&ouid=108176994911711108451&rtpof=true&sd=true

8. Anexo 3. Investigacion modelos LLM open Source y limitaciones

En el desarrollo del chatbot para el Instituto Tecnológico del Putumayo (ITP), se exploraron diversas opciones para integrar inteligencia artificial al chatbot, en este sentido se investigó sobre los modelos de lenguaje de gran tamaño (LLM) que permitan un procesamiento del lenguaje natural eficiente y efectivo. En esta parte de la investigación se centró en evaluar modelos de IA open-source debido a su potencial para reducir costos operativos y brindar flexibilidad en la implementación.

El objetivo principal de esta evaluación fue identificar modelos que pudieran ofrecer respuestas contextuales precisas y manejar múltiples usuarios simultáneamente. Se realizó una investigación sobre la informacion y documentación existente y se llevaron a cabo pruebas prácticas con varios modelos de LLM open-source para determinar su viabilidad en el contexto del proyecto. Los modelos evaluados incluyeron Llama2, GPT-4all, Phi3, Mistral y Gemma.

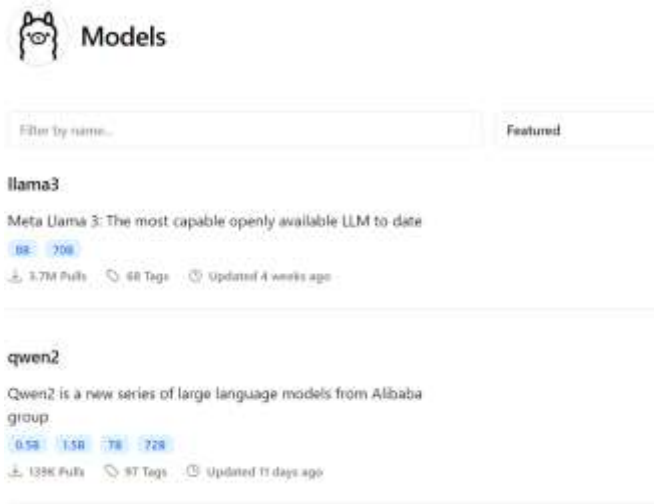
A lo largo de esta investigación, se identificaron tanto las capacidades como las limitaciones de cada modelo, permitiendo una comparación exhaustiva y fundamentada. A continuación, se presentan los detalles de cada modelo, incluyendo sus características clave y las limitaciones encontradas durante las pruebas.

Modelos Evaluados

Para hacer uso de algunos modelos que se presentaran a continuación, solo es necesario descargar Ollama AI esta es una herramienta que permite ejecutar modelos de lenguaje a gran escala (LLMs) directamente en la maquina local.



Dirección de la librería de descarga de los LLMs open source: https://ollama.com/library



Llama2

Descripción: Llama2 es un modelo de lenguaje desarrollado por Meta AI, diseñado para ser eficiente y eficaz en tareas de procesamiento del lenguaje natural.

Características:

- Open-source.
- Optimizado para varios tipos de tareas de NLP.

Limitaciones Encontradas:

- Requiere hardware de alto rendimiento.
- Escalabilidad limitada para múltiples usuarios simultáneamente.

GPT-4all

Dirección del sitio: https://gpt4all.io/index.html

Descripción: GPT-4all es GPT4All es un ecosistema para entrenar e implementar modelos de lenguaje grandes potentes y personalizados que se ejecutan localmente en CPU de consumo.

Características:

- Open-source.
- Integración con diversos modelos LLMs o modelos de IA.

Limitaciones Encontradas:

- Consume una gran cantidad de recursos computacionales.
- Desafíos en la gestión de múltiples usuarios debido a su alta demanda de memoria y CPU.

Phi3

Descripción: Phi3 es un modelo desarrollado por la comunidad open-source, con enfoque en la eficiencia y la precisión en NLP.

Características:

- Modelo de código abierto.
- Enfoque en la reducción de costos operativos.

Limitaciones Encontradas:

- Limitada capacidad para manejar grandes volúmenes de consultas simultáneas.
- Requiere optimización significativa para su integración en aplicaciones de alta demanda.

Mistral

Descripción: Mistral es un modelo de lenguaje de código abierto orientado a la eficiencia y la escalabilidad en entornos de producción.

Características:

- Open-source.
- Diseño para alta eficiencia y bajo consumo de recursos.

Limitaciones Encontradas:

- Escalabilidad limitada para aplicaciones con muchos usuarios simultáneos.
- Dificultades en la optimización para reducir la latencia en respuestas.

Gemma

Descripción: Gemma es un modelo de NLP open-source diseñado para ser ligero y rápido, con aplicaciones en diversos campos de NLP.

Características:

- Open-source.
- Enfoque en la rapidez y eficiencia.

Limitaciones Encontradas:

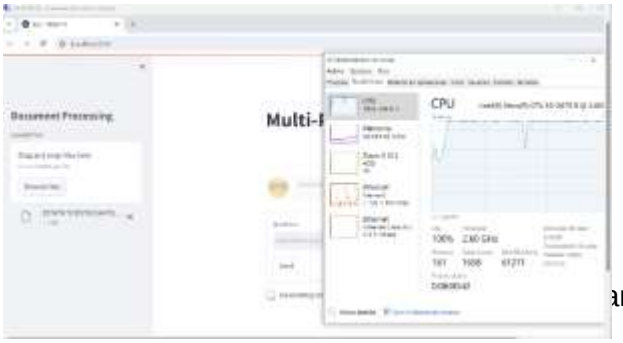
- Recursos computacionales necesarios pueden ser altos para entornos con múltiples usuarios.
- Limitada capacidad para manejar consultas complejas en tiempo real.

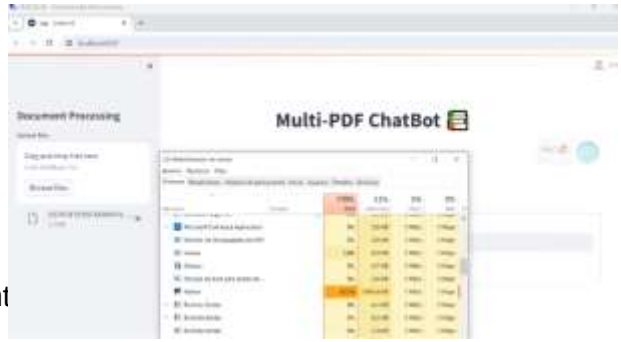
Análisis Comparativo

Modelo	Recursos Requeridos	Escalabilidad	Facilidad de Uso	Costo de Implementación
Llama 2	Alto	Moderada	Alta	Bajo
GPT-4all	Muy Alto	Baja	Alta	Bajo
Phi3	Moderado	Baja	Media	Bajo
Mistral	Alto	Baja	Media	Bajo
Gemma	Moderado	Baja	Alta	Bajo

Evidencia grafica de ejecución

Estas fueron algunas pruebas que se realizaron en una máquina virtual, con características computacionales, medianamente responsables, Procesador (8 núcleos), RAM (20 GB), este es un reflejo de los resultados que se obtenían a la hora de la ejecución de los modelos anteriores.





ant

computacionales, lo que limita su viabilidad en un entorno con múltiples usuarios simultáneos. La escalabilidad y la eficiencia en la gestión de múltiples usuarios son las principales barreras para la implementación de modelos LLM open-source en el proyecto del chatbot, al menos en estos momentos por lo que es una tecnología muy reciente.

Recomendación

Dada la necesidad de soporte para múltiples usuarios y la eficiencia en tiempo real, se optó por continuar con la API de OpenAI, que ofrece una solución probada y escalable para el proyecto del chatbot, en el momento que se realizó esta investigación es la mejor opción para este proyecto.

9. Anexo 4. Manual de usuario.

Este manual de usuario está diseñado para guiarte en el uso del Chatbot con IA desarrollado para el Instituto Tecnológico del Putumayo. Nuestro Chatbot ha sido creado con el propósito de mejorar la gestión de la información, facilitando el acceso a documentos y recursos de manera rápida y autónoma, mejorando así la experiencia de la comunidad estudiantil y el personal académico.

El Chatbot está diseñado para mantener diálogos coherentes y contextuales con los usuarios, atendiendo solicitudes de información y servicios de manera eficiente. Con su implementación, esperamos reducir la carga sobre el personal administrativo y proporcionar acceso a la documentación más allá de los horarios tradicionales de atención.

En este manual, encontrarás una descripción detallada de las funciones del Chatbot, cómo interactuar con él, y las mejores prácticas para aprovechar al máximo sus capacidades. A continuación, se presenta el contenido que abarca desde cómo iniciar una conversación hasta la resolución de problemas comunes.

Objetivo

El objetivo de este **manual de usuario** es proporcionar una guía completa y detallada para el uso del Chatbot desarrollado para el Instituto Tecnológico del Putumayo. Este Chatbot ha sido diseñado para mejorar la gestión de la información dentro de la institución, facilitando el acceso a documentos y recursos de manera rápida y autónoma. Mediante este manual, se busca

asegurar que los usuarios puedan interactuar de manera eficiente y efectiva con el Chatbot, aprovechando al máximo sus capacidades para mejorar la experiencia de la comunidad estudiantil y el personal académico. Además, se pretende reducir la carga sobre el personal administrativo al ofrecer acceso continuo a la información y servicios, incluso fuera del horario de atención tradicional.

Empezar a usar el software asistente virtual del ITP

1. **Inicio de la pagina**

La pantalla principal, como se muestra en la Figura 1, presenta múltiples funciones.



Figura 1: Ventana principal del programa

En la parte izquierda se muestra una barra lateral con cuatro botones, cuyas funciones se explicarán de manera ordenada. En la parte superior central, hay dos botones que representan los modos de chat. A la derecha de la barra lateral, se encuentra el chat, que incluye un menú de categorías de preguntas. Finalmente, en la parte derecha del chat, se encuentran los PDFs cargados en el software.

2. Menú superior central, modos de chat.

2.1. Modo básico.

Opciones del modo básico, Figura 2.



Figura 2: Ventana del modo básico

En este modo, los usuarios podrán hacer preguntas sobre todos los PDFs subidos en el software. También tendrán acceso a un menú de opciones de preguntas frecuentes para cada PDF cargado (Figura 3 y 4). Los usuarios pueden seleccionar una opción y presionar el botón correspondiente para obtener una respuesta rápida. Este menú fue diseñado para ofrecer ideas base de preguntas, en caso de que los usuarios no sepan qué preguntar.

Figura 3: preguntas base del estatuto estudiantil

Figura 4: preguntas base del estatuto general



ner una respuesta, se pueden observar dos números en la parte inferior del mensaje, los cuales actúan como enlaces o botones que indican de dónde se obtuvo la información en los Pdfs (Figura 5).

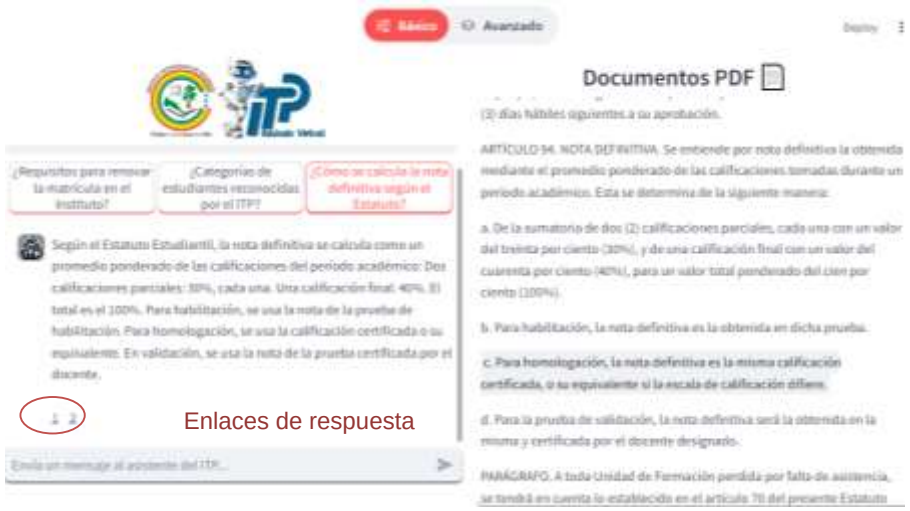


Figura 5: respuesta y enlaces de información

Al escribir una pregunta, tener en cuenta que los caracteres están limitados a 200. (Figura 6)

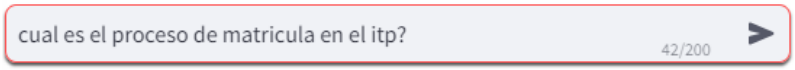


Figura 6: entrada de chat limitada

2.2. Modo avanzado.

En este modo avanzado contamos con la misma interfaz (Figura 7)



Figura 7: interfaz del modo avanzado

En este modo, es necesario elegir una categoría antes de hacer una pregunta. Al seleccionar una categoría, se mostrarán las preguntas frecuentes y el PDF correspondiente. En resumen, el modo avanzado permite hacer preguntas a un solo PDF para obtener respuestas más precisas y detalladas.

El resto de las funciones son iguales a las del modo anterior: se presiona un botón para obtener una respuesta, los enlaces o botones numerados indican la fuente de la información, y la entrada del chat está limitada a 200 caracteres.

3. **Barra latera, funciones.**

En la barra lateral, tenemos cuatro botones: Nuevo chat, Inicio, Información y Configuración, como se muestra en la interfaz (Figura 8).

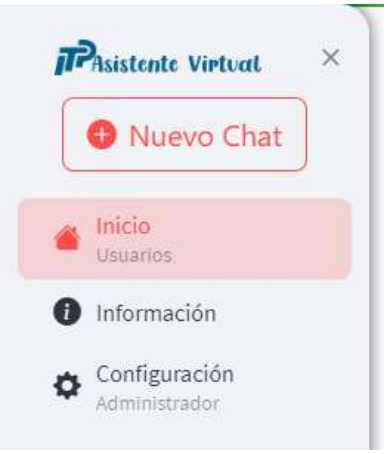


Figura 8: Barra lateral

3.1. Nuevo chat

Al presionar el botón "Nuevo chat", como su nombre lo indica, se iniciará una nueva conversación. Tener en cuenta que al hacerlo, ya no se podrá ver los mensajes recibidos anteriormente.

3.2.Inicio

Esta es la página principal del software, a la que tienen acceso todos los usuarios y donde se encuentran las funciones explicadas anteriormente.

4. Información

En el botón de Información, se encuentra una página a la que todos los usuarios pueden acceder. Esta página contiene detalles sobre cómo funciona cada modo de chat. Además, aquí se puede añadir información más detallada sobre qué es el software y cómo funciona (Figura 9).



Figura 9: Pagina de Información básica de los modos de chat.

5. Configuración

Esta página está reservada exclusivamente para el administrador, quien cuenta con una clave de acceso. Aquí se encuentran todas las configuraciones necesarias para añadir o eliminar opciones y/o documentos del chat (Figura 10).



Figura 10: página de configuración solo para administradores.

Una vez que haya ingresado como administrador, podrá ver la interfaz (Figura 11)



Figura 11: interfaz accediendo como administrador.

En este apartado gráfico, puede administrar las categorías, los PDFs, las preguntas frecuentes, las palabras clave y modificar la contraseña de acceso.

5.1. Administrar categorías

En esta categoría podemos añadir categorías en función de los PDFs cargados en el software. Esto significa que, para subir un PDF, primero debemos crear su categoría correspondiente para que el usuario pueda hacer preguntas sobre la información subida (Figura 12). También podemos eliminar categorías creadas previamente, simplemente seleccionamos la categoría y presionamos el botón eliminar (Figura 13).



Figura 12: Añadir categorías.

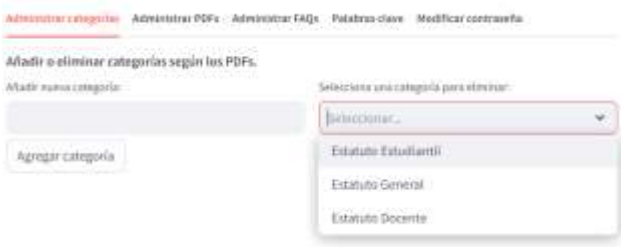


Figura 13: Eliminar categorías.

5.2. Administrar PDFs

Aquí podemos añadir o eliminar los PDFs subidos (Figura 14).

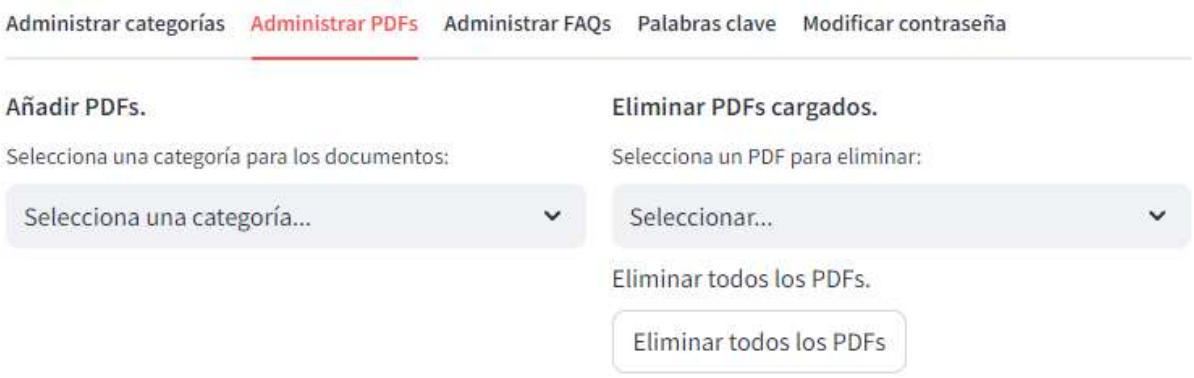


Figura 14: Subir o eliminar PDFs.

Para **agregar** archivos PDF, simplemente selecciona la categoría deseada y cargar el archivo PDF correspondiente (Figura 15 y 16).

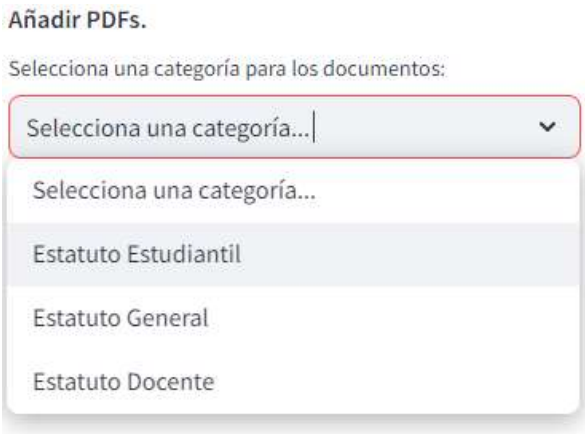


Figura 15: Seleccionar categoría.

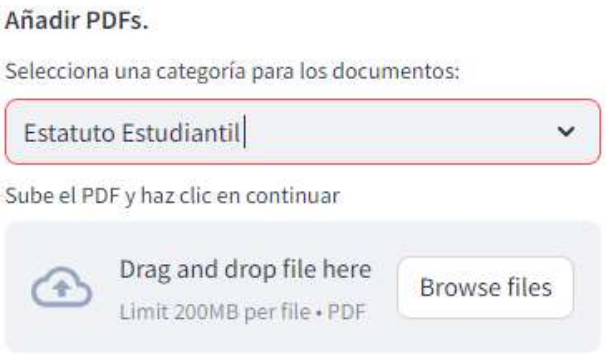


Figura 16: Seleccionar PDF a cargar.

Para **eliminar** archivos PDF, tienes dos opciones: la primera es eliminar los archivos uno por uno seleccionando el nombre del PDF y haciendo clic en el botón de eliminar (Figura 17 y 18); la segunda opción es eliminar todos los PDFs de una vez utilizando el botón que aparece en pantalla (Figura 18).

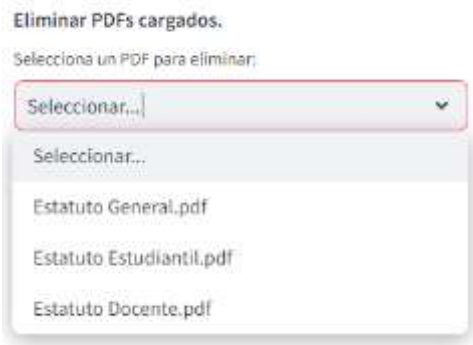


Figura 17: Seleccionar PDF a eliminar.

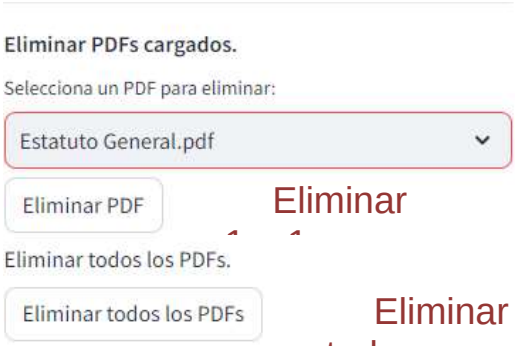


Figura 18: botones para eliminar.

5.3. Administrar preguntas frecuentes

Administrar las preguntas frecuentes tenemos las opciones de agregar, editar o eliminar (Figura 19)

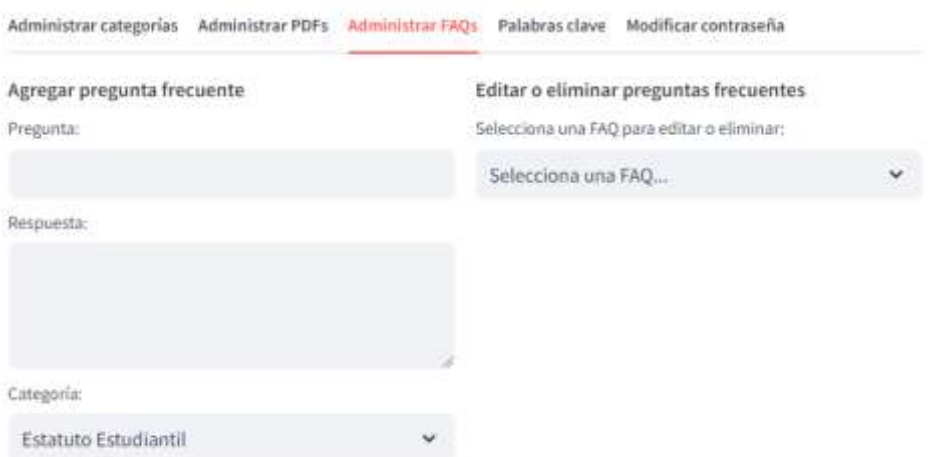


Figura 19: Administrar preguntas frecuentes.

Agregar preguntas frecuentes, simplemente introduce la pregunta, su respuesta y selecciona la categoría a la que pertenecerá (Figura 20).

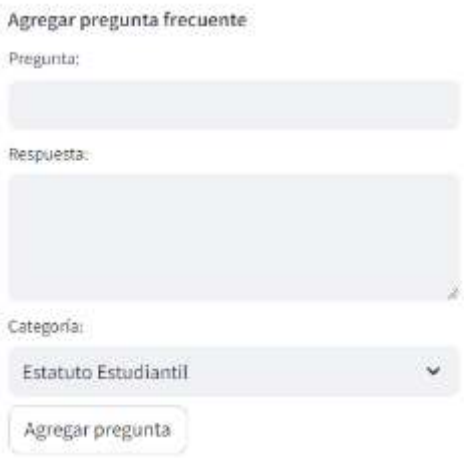


Figura 20: Agregar pregunta frecuente.

Editar las preguntas, selecciona la pregunta que deseas modificar (Figura 21). Aparecerán los campos para cambiar el texto necesario, como la pregunta, la respuesta y la categoría. Una vez realizados los cambios, simplemente presiona el botón 'Guardar cambios' (Figura 22).

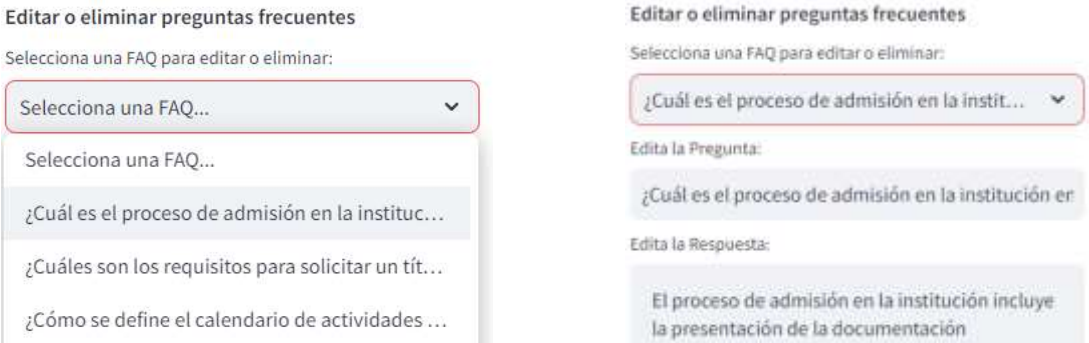


Figura 21: Elegir pregunta frecuente.

Figura 22: Editar pregunta frecuente.

Cuando decidimos **eliminar** las preguntas, disponemos de dos alternativas. Una opción consiste en eliminar individualmente cada pregunta seleccionando el botón correspondiente junto a cada una, mientras que la otra alternativa es eliminar todas las preguntas de una sola vez (Figura 23).

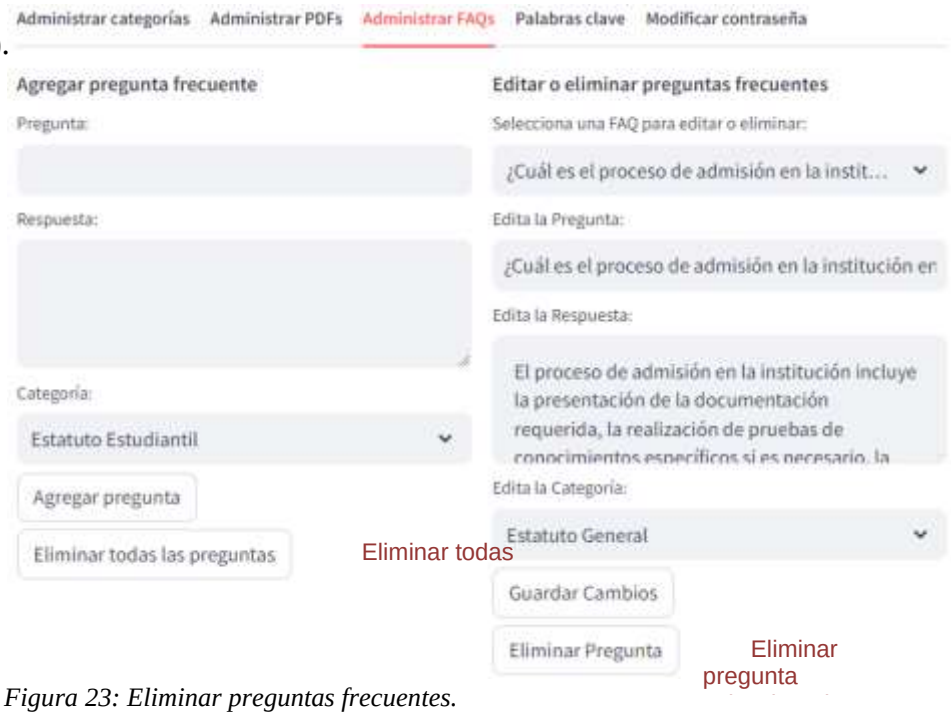


Figura 23: Eliminar preguntas frecuentes.

5.4. Administrar palabras clave

Las palabras clave son necesarias para que, cuando un usuario envíe un mensaje, este incluya una de ellas para que el mensaje pueda ser enviado. Si un usuario envía un mensaje sin

una palabra clave, recibirá una advertencia que dice 'la pregunta no está relacionada con el instituto', estas pueden ser agregadas y eliminadas.

Para agregar las palabras solo hay que escribirlas separadas por comas, al momento de eliminarlas solo seleccionar una palabra (Figura 24).



Figura 24: Palabras clave, agregar o eliminar.

5.5. Modificar contraseña

Cambiar la contraseña consiste simplemente en ingresar la nueva contraseña para acceder a la sección de configuración (Figura 25).



Figura 25: Modificar contraseña.

5.6. Bloquear secciones

Después de completar todas las configuraciones, hay un botón disponible para bloquear las secciones de los botones (Figura 26).

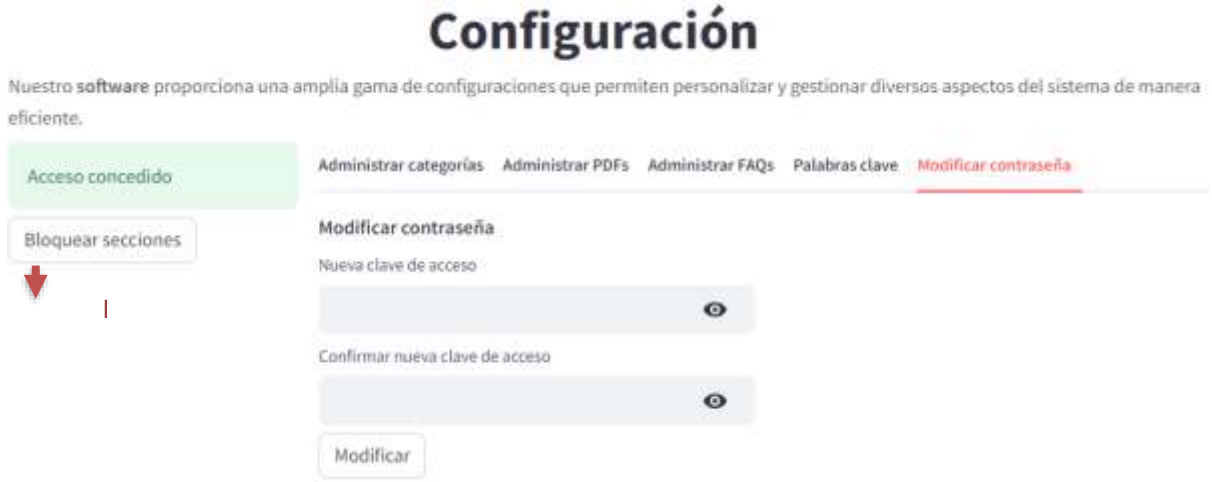


Figura 25: Botón de bloquear secciones de configuración.



Menú superior central
Elegir modo

1. Básico
2. Avanzado

Menú lateral

3. Nuevo chat
4. Inicio
5. Información
6. Configuración

Chat

7. Logo
8. Categorías de preguntas

9. PDFs

10. Anexo 5. Manual de Programador

El propósito de este manual del programador es proporcionar al lector una comprensión detallada de los códigos fuente utilizados en el desarrollo del software. A través de este documento, se busca explicar de manera clara y concisa cada segmento del código, así como la lógica de programación empleada en la creación del software.

Este manual está diseñado para que el usuario pueda modificar según sus necesidades los valores y parámetros de las funciones presentadas. De este modo, se facilita la personalización y ajuste del asistente virtual del ITP, permitiendo a los programadores realizar cambios específicos y optimizaciones de acuerdo con sus requerimientos.

Herramientas de desarrollo

Para la creación de este software se utilizó las siguientes herramientas.

Sistema Operativo	Lenguaje de Programación	Entorno de Desarrollo	Base de Datos	Motor de Búsqueda	Frontend
Windows 10 y/o 11	Python V3.12.4	Visual Studio Code	Mongo DB	Elasticsearch	Streamlit, HTML, CSS

Tabla 1. Tecnologías de desarrollo

Base de datos

MongoDB es un sistema de gestión de bases de datos (DBMS) no relacional de código abierto, diseñado para almacenar grandes volúmenes de datos. A diferencia de las bases de datos relacionales tradicionales, MongoDB no se organiza en tablas y columnas, sino en documentos y colecciones.

Las colecciones en MongoDB son agrupaciones de documentos y son equivalentes a las tablas en una base de datos relacional. Esta arquitectura basada en documentos y colecciones permite a MongoDB gestionar eficientemente grandes volúmenes de datos, proporcionando una alta escalabilidad y rendimiento.

Creación y configuración de Mongo DB

Una vez creada la base de datos, denominada **itp_documents**, la cual es completamente no relacional, el siguiente paso es estructurar su diseño (Figura 1).

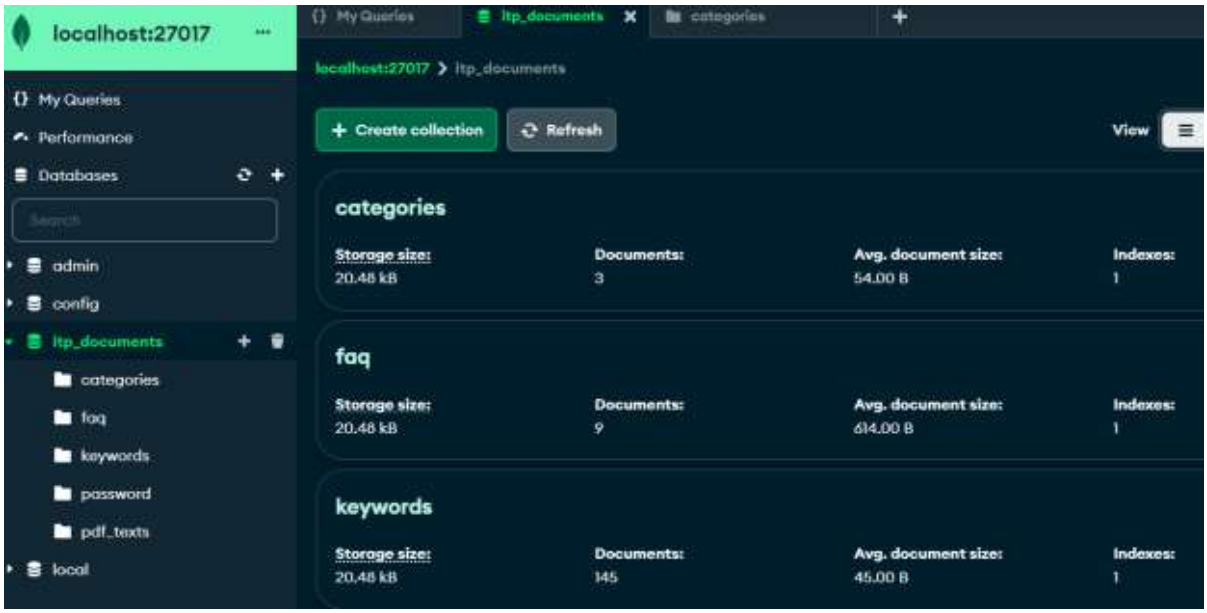


Figura 1. Mongo DB, colecciones

Para ello se determina la necesidad de la creación de las siguientes colecciones:

- **Categories**: Esta colección se encarga de almacenar las categorías de los documentos, predeterminadas por el administrador.
- **Faq**: Es la colección que se encarga de almacenar las preguntas frecuentes, predeterminadas por el administrador.

- **Keywords**: En esta colección se agregan algunas de las palabras, que se tendrán en cuenta para validar las consultas d ellos usuarios y evitar el uso indebido del programa.
- **Password**: Esta colección contiene la clave de acceso con la que el administrador, puede realizar cambios en el programa.
- **Pdf_texts**: En esta colección se almacenan los textos de los documentos o pdfs cargados al programa.

Teniendo en cuenta todos los datos anteriores, la estructura de la base de datos en Mongo db es la siguiente manera (Figura 2).

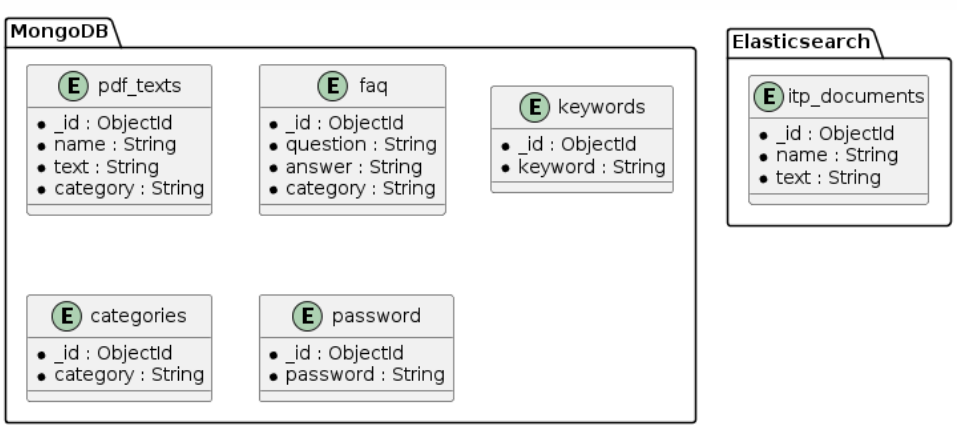


Figura 2. Estructura de DB.

Buscador vectorial

La instalación de **Elasticsearch** se lleva a cabo accediendo al sitio oficial de Elasticsearch (Figura 3). Una vez allí, se descarga el archivo zip correspondiente y se procede a descomprimirlo en una ubicación seleccionada en el ordenador. Para ejecutar el servidor de Elasticsearch, se abre una ventana de línea de comandos (CMD) en la raíz de la carpeta descomprimida y se utiliza el siguiente comando: **bin\elasticsearch.bat**.

Este proceso inicializa Elasticsearch y pone en funcionamiento el servidor para empezar a utilizar esta potente herramienta de búsqueda y análisis de datos.

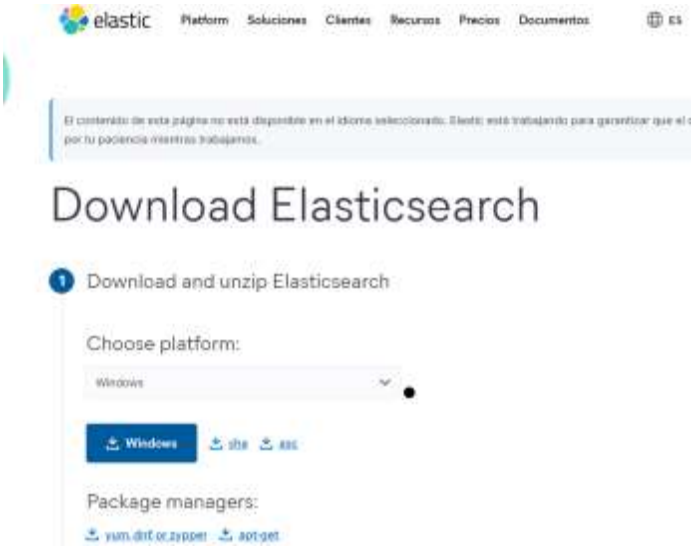


Figura 3. Elasticsearch, descarga.

Crear proyecto

Se crea el proyecto en Visual Studio Code, se le asigna un nombre en este caso el proyecto lleva como nombre **app** (Figura 4).

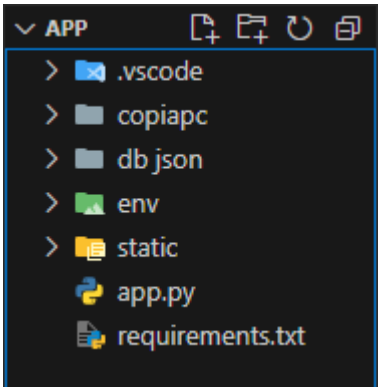


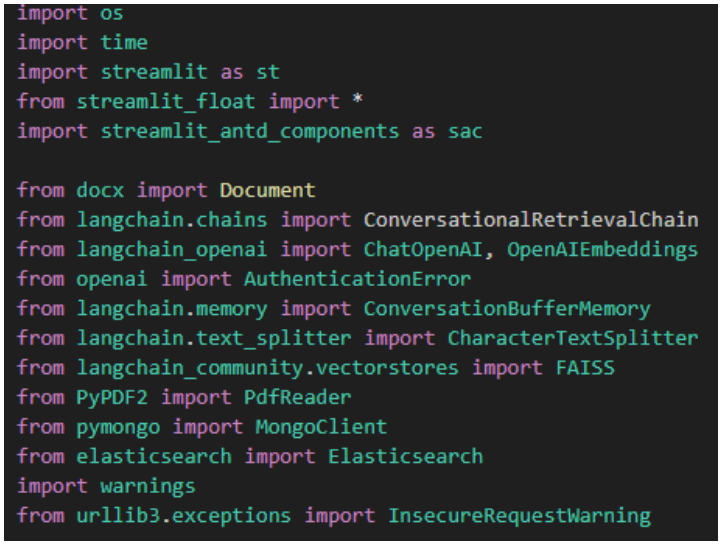
Figura 4. Proyecto en VSCode.

Para ejecutar el software, primero se instala un entorno virtual utilizando el siguiente comando: **python -m venv env**

Después de activar el entorno virtual, se puede comenzar a trabajar en el archivo ‘**app.py**’.

Funciones y Métodos

El primer paso en este archivo '**app.py'**, es importar las librerías necesarias para el desarrollo del software (Ilustración 1).



```
import os
import time
import streamlit as st
from streamlit_float import *
import streamlit_antd_components as sac

from docx import Document
from langchain.chains import ConversationalRetrievalChain
from langchain_openai import ChatOpenAI, OpenAIEmbeddings
from openai import AuthenticationError
from langchain.memory import ConversationBufferMemory
from langchain.text_splitter import CharacterTextSplitter
from langchain_community.vectorstores import FAISS
from PyPDF2 import PdfReader
from pymongo import MongoClient
from elasticsearch import Elasticsearch
import warnings
from urllib3.exceptions import InsecureRequestWarning
```

Ilustración 1. Librerías

Librerías de Sistema y Utilidad

os

- **Descripción**: Proporciona funciones para interactuar con el sistema operativo.
- **Uso común**: Manipular archivos y directorios, obtener información del sistema, ejecutar comandos del sistema.
- **Ejemplo**: os.listdir('.') lista todos los archivos y directorios en el directorio actual.

time

- **Descripción**: Proporciona funciones relacionadas con el tiempo.
- **Uso común**: Medir intervalos de tiempo, obtener la hora actual, retrasar la ejecución del programa.
- **Ejemplo**: time.sleep(5) pausa la ejecución del programa durante 5 segundos.

Librerías para Desarrollo de Aplicaciones Web

streamlit as st

- **Descripción**: Biblioteca para crear aplicaciones web interactivas y visualizaciones de datos de manera rápida y sencilla.
- **Uso común:** Crear dashboards interactivos y aplicaciones de análisis de datos.
- **Ejemplo**: st.write("Hola Mundo") muestra "Hola Mundo" en una aplicación Streamlit.

streamlit_antd_components as sac

- **Descripción**: Proporciona componentes adicionales de UI basados en Ant Design para Streamlit.
- **Uso común**: Mejorar la interfaz de usuario de una aplicación Streamlit con componentes avanzados.
- **Ejemplo**: sac.Button("Click me") crea un botón estilizado con Ant Design.

Librerías para Procesamiento de Documentos

docx

- **Descripción**: Permite crear, modificar y leer archivos de Microsoft Word (.docx).
- **Uso común**: Generar informes, procesar documentos de Word.
- **Ejemplo**: doc = Document(); doc.add_paragraph("Hola Mundo"); doc.save("demo.docx") crea un archivo de Word con un párrafo que dice "Hola Mundo".

PyPDF2

- **Descripción**: Biblioteca para trabajar con archivos PDF.
- **Uso común**: Leer, escribir, combinar y dividir archivos PDF.
- **Ejemplo**: reader = PdfReader("document.pdf"); page = reader.pages[0]; text = page.extract_text() extrae el texto de la primera página de un PDF.

Librerías para Procesamiento de Lenguaje Natural (NLP)

langchain.chains.ConversationalRetrievalChain

- **Descripción**: Parte de la biblioteca LangChain, utilizada para construir cadenas conversacionales con capacidades de recuperación de información.
- **Uso común**: Implementar chatbots que pueden buscar y recuperar información de grandes conjuntos de datos.
- **Ejemplo**: Crear un chatbot que responde preguntas basadas en documentos.

langchain_openai.ChatOpenAI

- **Descripción**: Interfaz para interactuar con modelos de chat de OpenAI a través de LangChain.
- **Uso común**: Integrar modelos de lenguaje de OpenAI en aplicaciones de NLP.
- **Ejemplo**: chatbot = ChatOpenAI(api_key="tu_api_key").

langchain_openai.OpenAIEmbeddings

- **Descripción**: Genera embeddings (representaciones numéricas) de textos utilizando modelos de OpenAI.
- **Uso común**: Embedding de textos para búsqueda y recuperación de información.
- **Ejemplo**: embeddings = OpenAIEmbeddings(api_key="tu_api_key"); vector = embeddings.embed("texto").

langchain.memory.ConversationBufferMemory

- **Descripción**: Maneja el estado de la conversación en aplicaciones de chat.
- **Uso común**: Almacenar y recuperar el historial de la conversación.
- **Ejemplo**: memory = ConversationBufferMemory().

langchain.text_splitter.CharacterTextSplitter

- **Descripción**: Divide textos en fragmentos más pequeños.
- **Uso común**: Preparar textos largos para su procesamiento en trozos manejables.
- **Ejemplo**: splitter = CharacterTextSplitter(); fragments = splitter.split(texto_largo).

langchain_community.vectorstores.FAISS

- **Descripción**: Implementación de FAISS (Facebook AI Similarity Search) para almacenar y buscar embeddings.
- **Uso común**: Indexación y búsqueda de textos basados en sus embeddings.
- **Ejemplo**: faiss_store = FAISS().

Librerías para Bases de Datos

pymongo.MongoClient

- **Descripción**: Cliente para interactuar con bases de datos MongoDB.

- **Uso común**: Conectar, consultar y manipular datos en MongoDB.
- **Ejemplo**: client = MongoClient("mongodb://localhost:27017/"); db = client["mi_bd"].

elasticsearch.Elasticsearch

- **Descripción**: Cliente para interactuar con servidores de Elasticsearch.
- **Uso común**: Indexar, buscar y analizar grandes volúmenes de datos en Elasticsearch.
- **Ejemplo**: es = Elasticsearch(); es.index(index="mi_indice", body=doc).

Otras Utilidades

warnings

- **Descripción**: Proporciona funciones para manejar advertencias en el código.
- **Uso común**: Emitir, suprimir y registrar advertencias.
- **Ejemplo**: warnings.warn("Esta es una advertencia").

urllib3.exceptions.InsecureRequestWarning

- **Descripción**: Clase de excepción utilizada para advertir sobre solicitudes HTTP inseguras.
- **Uso común**: Suprimir advertencias de solicitudes inseguras.
- **Ejemplo**: warnings.filterwarnings("ignore", category=InsecureRequestWarning).

Configuración a MongoDB

Línea 19. Ignora ciertas advertencias de seguridad relacionadas con las solicitudes

Después, se debe establecer una conexión con un servidor de MongoDB. A continuación, se selecciona una base de datos específica dentro de ese servidor. Posteriormente, se obtienen referencias a varias colecciones dentro de dicha base de datos, preparando así el entorno para realizar operaciones posteriores con esos datos (Ilustración 2).

```
warnings.simplefilter('ignore', InsecureRequestWarning)

client = MongoClient("mongodb://localhost:27017/")
db = client["itp_documents"]
collection = db["pdf_texts"]
collection_faq = db["faq"]
collection_keywords = db["keywords"]
collection_categories = db["categories"]
collection_password = db["password"]
collection_settings = db["settings"]
```

Ilustración 2. Conexión a MongoDB

Conexión a Elasticsearch

El código establece una conexión con un servidor de Elasticsearch, una potente herramienta de búsqueda y análisis que permite almacenar, buscar y analizar grandes volúmenes de datos de manera rápida y en tiempo real (Ilustración 3).

```
es = Elasticsearch(
    ["https://localhost:9200"],
    basic_auth=("elastic", "blpR+66njAXsS3P2hP6w"),
    verify_certs=False
)
```

Ilustración 3. Conexión a MongoDB

Funciones

Este fragmento de código define varias funciones que se utilizan para cargar una clave API desde una base de datos, extraer texto de documentos PDF, y dividir texto en fragmentos manejables (Ilustración 4).



```
def load_api_key():
    settings_doc = collection_settings.find_one({"name": "api_key"})
    if settings_doc:
        return settings_doc.get("value")
    return None

def parse_docx(data):
    document = Document(docx=data)
    content = ""
    for para in document.paragraphs:
        content += para.text
    return content

def get_text(doc):
    doc_text = ""
    if ".pdf" in doc.name:
        pdf_reader = PdfReader(doc)
        for each_page in pdf_reader.pages:
            doc_text += each_page.extract_text()
        doc_text += "\n"
    elif ".docx" in doc.name:
        doc_text += parse_docx(data=doc)
    return doc_text

def get_chunks(data):
    text_splitter = CharacterTextSplitter(
        separator="\n", chunk_size=1500, chunk_overlap=200, length_function=len
    )
    text_chunks = text_splitter.split_text(data)
    return text_chunks
```

Ilustración 4. Funciones

- **load_api_key**: Carga una clave API desde MongoDB.
- **parse_docx**: Extrae y concatena texto de un archivo DOCX.
- **get_text**: Extrae texto de documentos PDF y DOCX.
- **get_chunks**: Divide el texto en fragmentos manejables.

Define varias funciones para almacenar, indexar y cargar textos de documentos PDF en una base de datos MongoDB y un servidor Elasticsearch (Ilustración 5).



```python
def store_pdf_text_in_db(pdf_name, text, category):
    if collection.find_one({"name": pdf_name}):
        print(f"Documento {pdf_name} ya existe en la base de datos.")
        return
    document = {
        "name": pdf_name,
        "text": text,
        "category": category
    }
    collection.insert_one(document)

def index_pdf_text_in_elasticsearch(pdf_name, text):
    document = {
        "name": pdf_name,
        "text": text
    }
    es.index(index="itp_documents", body=document)

def load_texts_from_db():
    texts = []
    for document in collection.find():
        texts.append(document["text"])
    return texts

def load_all_texts_from_db():
    texts = []
    for document in collection.find():
        texts.append(f"Documento: {document['name']}\n{document['text']}\n{'-'*80}\n")
    return "\n".join(texts)
```

Ilustración 5. Funciones

- **store_pdf_text_in_db**: Almacena el texto de un documento PDF en MongoDB, evitando duplicados.
- **index_pdf_text_in_elasticsearch**: Indexa el texto de un documento PDF en Elasticsearch para facilitar la búsqueda.
- **load_texts_from_db**: Carga todos los textos de los documentos PDF desde MongoDB en una lista.
- **load_all_texts_from_db**: Carga todos los textos de los documentos PDF desde MongoDB, incluyendo nombres y separadores, y los devuelve como una cadena única.

Funciones para crear vectores de texto utilizando embeddings, configurar una cadena de modelos de lenguaje (LLM chain) para la recuperación conversacional y actualizar estos vectores en el estado de la sesión de Streamlit (Ilustración 6).



```
def get_vector(chunks):
    return FAISS.from_texts(texts=chunks, embedding=OpenAIEmbeddings())

def get_llm_chain(vectors):
    retriever = vectors.as_retriever()
    memory = ConversationBufferMemory(
        memory_key="chat_history",
        return_messages=True,
        output_key="answer"
    )
    llm_chain = ConversationalRetrievalChain.from_llm(
        llm=ChatOpenAI(model_name="gpt-3.5-turbo", temperature=0.7),
        retriever=retriever,
        memory=memory,
        return_source_documents=True
    )
    return llm_chain

def update_vectors():
    texts = load_texts_from_db()
    if texts:
        doc_chunks = []
        for text in texts:
            doc_chunks.extend(get_chunks(text))
        vectors = get_vector(doc_chunks)
        st.session_state.llm_chain = get_llm_chain(vectors)
        st.session_state.doc_len = len(texts)
```

Ilustración 6. Funciones

- **get_vector**: Crea un índice de vectores de texto a partir de fragmentos utilizando embeddings de OpenAI y FAISS.
- **get_llm_chain**: Configura una cadena de modelos de lenguaje para la recuperación conversacional utilizando un modelo de lenguaje y un índice de vectores.
- **update_vectors**: Actualiza los vectores de texto y la cadena de modelos de lenguaje en el estado de la sesión de Streamlit, cargando los textos desde la base de datos y procesándolos.

Funciones adicionales para actualizar vectores de texto, mostrar texto con un efecto de escritura, y gestionar palabras clave en una base de datos MongoDB (Ilustración 7).

```python
def update_vectors_advanced(pdf_text):
    doc_chunks = get_chunks(pdf_text)
    vectors = get_vector(doc_chunks)
    st.session_state.llm_chain = get_llm_chain(vectors)
    st.session_state.doc_len = 1

def type_effect(text, placeholder):
    full_response = ""
    for chunk in text.split():
        full_response += chunk + " "
        time.sleep(0.05)
        placeholder.markdown(full_response + "▌", unsafe_allow_html=True)
    placeholder.markdown(full_response, unsafe_allow_html=True)

def add_keyword(keyword):
    if not collection_keywords.find_one({"keyword": keyword}):
        collection_keywords.insert_one({"keyword": keyword})

def add_keywords(keywords):
    keywords = [keyword.strip() for keyword in keywords.split(",")]
    for keyword in keywords:
        if not collection_keywords.find_one({"keyword": keyword}):
            collection_keywords.insert_one({"keyword": keyword})

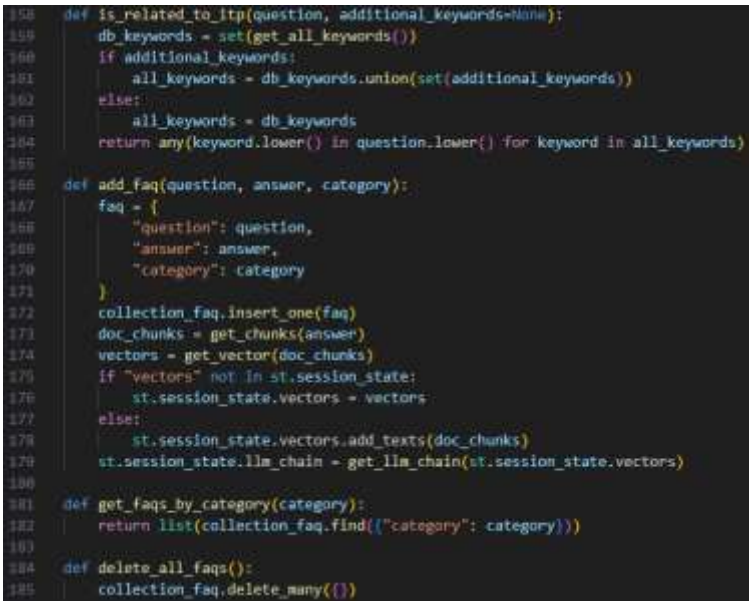
def delete_keyword(keyword):
    collection_keywords.delete_one({"keyword": keyword})

def get_all_keywords():
    keywords = collection_keywords.find()
    return [keyword["keyword"] for keyword in keywords]
```

Ilustración 7. Funciones

- **update_vectors_advanced**: Actualiza los vectores de texto en el estado de la sesión de Streamlit utilizando un texto PDF.
- **type_effect**: Muestra texto con un efecto de escritura en una interfaz de usuario.
- **add_keyword**: Añade una palabra clave a la base de datos si no existe.
- **add_keywords**: Añade múltiples palabras clave a la base de datos.
- **delete_keyword**: Elimina una palabra clave de la base de datos.
- **get_all_keywords**: Obtiene todas las palabras clave de la base de datos.

Este fragmento de código define varias funciones para determinar la relevancia de una pregunta en relación con palabras clave de ITP, añadir preguntas frecuentes (FAQs) a una base de datos, obtener FAQs por categoría y eliminar todas las FAQs (Ilustración 8).



```
def is_related_to_itp(question, additional_keywords=None):
    db_keywords = set(get_all_keywords())
    if additional_keywords:
        all_keywords = db_keywords.union(set(additional_keywords))
    else:
        all_keywords = db_keywords
    return any(keyword.lower() in question.lower() for keyword in all_keywords)

def add_faq(question, answer, category):
    faq = {
        "question": question,
        "answer": answer,
        "category": category
    }
    collection_faq.insert_one(faq)
    doc_chunks = get_chunks(answer)
    vectors = get_vector(doc_chunks)
    if "vectors" not in st.session_state:
        st.session_state.vectors = vectors
    else:
        st.session_state.vectors.add_texts(doc_chunks)
    st.session_state.llm_chain = get_llm_chain(st.session_state.vectors)

def get_faqs_by_category(category):
    return list(collection_faq.find({"category": category}))

def delete_all_faqs():
    collection_faq.delete_many({})
```

Ilustración 8. Funciones

- **is_related_to_itp**: Determina si una pregunta está relacionada con palabras clave específicas.
- **add_faq**: Añade una FAQ a la base de datos y actualiza los vectores de texto en el estado de la sesión de Streamlit.
- **get_faqs_by_category**: Obtiene todas las FAQs de una categoría específica.
- **delete_all_faqs**: Elimina todas las FAQs de la base de datos.

Define funciones para añadir y obtener categorías, resaltar textos con palabras destacadas, y verificar una contraseña almacenada en una base de datos MongoDB (Ilustración 9).

- **add_category**: Añade una nueva categoría a la base de datos si no existe ya.
- **get_all_categories**: Obtiene todas las categorías almacenadas en la base de datos.
- **highlight_texts**: Resalta palabras o frases específicas en un texto utilizando etiquetas HTML.
- **check_password**: Verifica una contraseña ingresada por el usuario contra una contraseña almacenada en la base de datos y actualiza el estado de la sesión en Streamlit.



```
def add_category(category):
    if not collection_categories.find_one({"category": category}):
        collection_categories.insert_one({"category": category})

def get_all_categories():
    categories = collection_categories.find()
    return [category["category"] for category in categories]

def highlight_texts(text, highlights):
    for idx, highlight in enumerate(highlights):
        if highlight in text:
            text = text.replace(highlight, f"<mark id='highlight-{idx}'>{highlight}</mark>")
    return text

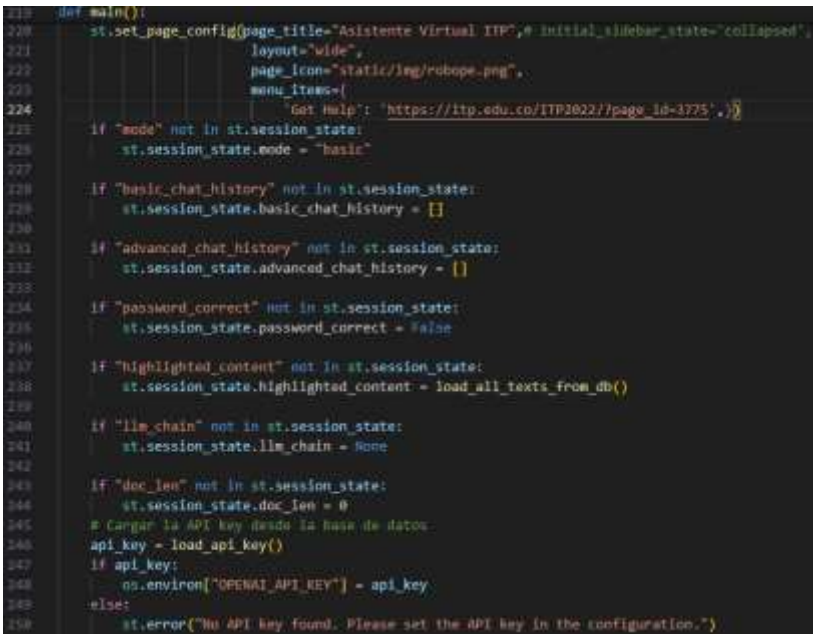
def check_password():
    if "password_correct" not in st.session_state:
        st.session_state.password_correct = False
    password_input = st.text_input("**Clave de Acceso**", type="password")
    if st.button("Ingresar"):
        stored_password_doc = collection_password.find_one()
        if stored_password_doc:
            stored_password = stored_password_doc.get("password")
            if password_input == stored_password:
                st.session_state.password_correct = True
                st.success("Acceso concedido")
            else:
                st.session_state.password_correct = False
                st.error("Clave incorrecta")
        else:
            st.error("No se ha configurado ninguna contraseña.")
```

Ilustración 9. Funciones

Función ‘main()’

El código define una función ‘main()’ para configurar y gestionar el estado de una aplicación de Streamlit que actúa como un asistente virtual. Esta función se encarga de establecer la configuración inicial de la página, gestionar el estado de la sesión y cargar claves API necesarias (Ilustración 10).

- **Configuración de la Página**: Define el título, el diseño y el icono de la página, además de añadir un enlace de ayuda al menú.
- **Inicialización del Estado de la Sesión**: Establece valores iniciales para varios elementos del estado de la sesión, asegurando que todas las variables necesarias estén definidas.
- **Carga de la API Key**: Carga la clave API desde la base de datos y la establece como una variable de entorno, mostrando un mensaje de error si no se encuentra la clave.



```
def main():
    st.set_page_config(page_title="Asistente Virtual ITP",# initial_sidebar_state='collapsed',
                       layout="wide",
                       page_icon="static/img/robope.png",
                       menu_items={
                           'Get Help': 'https://itp.edu.co/ITP2022/?page_id=3775',})
    if "mode" not in st.session_state:
        st.session_state.mode = "basic"

    if "basic_chat_history" not in st.session_state:
        st.session_state.basic_chat_history = []

    if "advanced_chat_history" not in st.session_state:
        st.session_state.advanced_chat_history = []

    if "password_correct" not in st.session_state:
        st.session_state.password_correct = False

    if "highlighted_content" not in st.session_state:
        st.session_state.highlighted_content = load_all_texts_from_db()

    if "llm_chain" not in st.session_state:
        st.session_state.llm_chain = None

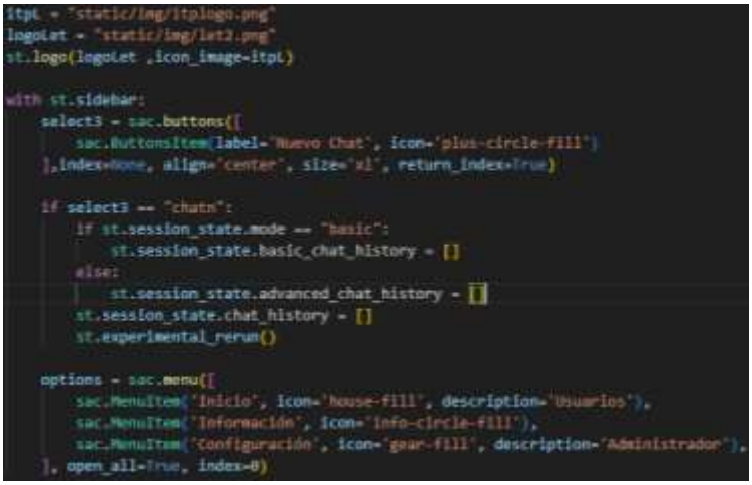
    if "doc_len" not in st.session_state:
        st.session_state.doc_len = 0
    # Cargar la API key desde la base de datos
    api_key = load_api_key()
    if api_key:
        os.environ["OPENAI_API_KEY"] = api_key
    else:
        st.error("No API key found. Please set the API key in the configuration.")
```

Ilustración 10. Main() configuración inicial de la página

Sidebar

Define la interfaz gráfica del asistente virtual utilizando Streamlit y componentes de streamlit_antd_components (Ilustración 11).

- **Logos y Encabezado**: Muestra los logos de la aplicación en el encabezado.
- **Botón 'Nuevo Chat'**: Añade un botón en la barra lateral para iniciar un nuevo chat, reiniciando el historial de chat cuando se selecciona.
- **Menú de Navegación**: Crea un menú de navegación en la barra lateral con opciones de 'Inicio', 'Información' y 'Configuración'.



```
itpL = "static/img/itplogo.png"
logoLet = "static/img/let2.png"
st.logo(logoLet ,icon_image=itpL)

with st.sidebar:
    select3 = sac.buttons([
        sac.ButtonsItem(label='Nuevo Chat', icon='plus-circle-fill')
    ],index=None, align='center', size='xl', return_index=True)

    if select3 == "chatn":
        if st.session_state.mode == "basic":
            st.session_state.basic_chat_history = []
        else:
            st.session_state.advanced_chat_history = []
        st.session_state.chat_history = []
        st.experimental_rerun()

    options = sac.menu([
        sac.MenuItem('Inicio', icon='house-fill', description='Usuarios'),
        sac.MenuItem('Información', icon='info-circle-fill'),
        sac.MenuItem('Configuración', icon='gear-fill', description='Administrador'),
    ], open_all=True, index=0)
```

Ilustración 11. Interfaz gráfica del asistente virtual

Configuración de administrador

Este código implementa una interfaz de configuración para un sistema de asistente virtual basado en Streamlit.

Carga de Estilos CSS: Se carga un archivo CSS (config.css) para personalizar el aspecto visual de la página utilizando st.markdown.

Autenticación por Contraseña: Se verifica y gestiona el acceso mediante una contraseña guardada en una base de datos (collection_password). Si la contraseña ingresada coincide con la almacenada, se permite el acceso a las secciones configuradas.

Configuración de Tabs: Utiliza st.tabs para organizar diferentes secciones de configuración:

Administrar Categorías: Permite agregar nuevas categorías y eliminar las existentes.

Administrar PDFs: Permite cargar nuevos PDFs, eliminar PDFs existentes y limpiar todos los PDFs cargados.

Administrar FAQs: Facilita la gestión de preguntas frecuentes, incluyendo la adición, edición y eliminación de las mismas.

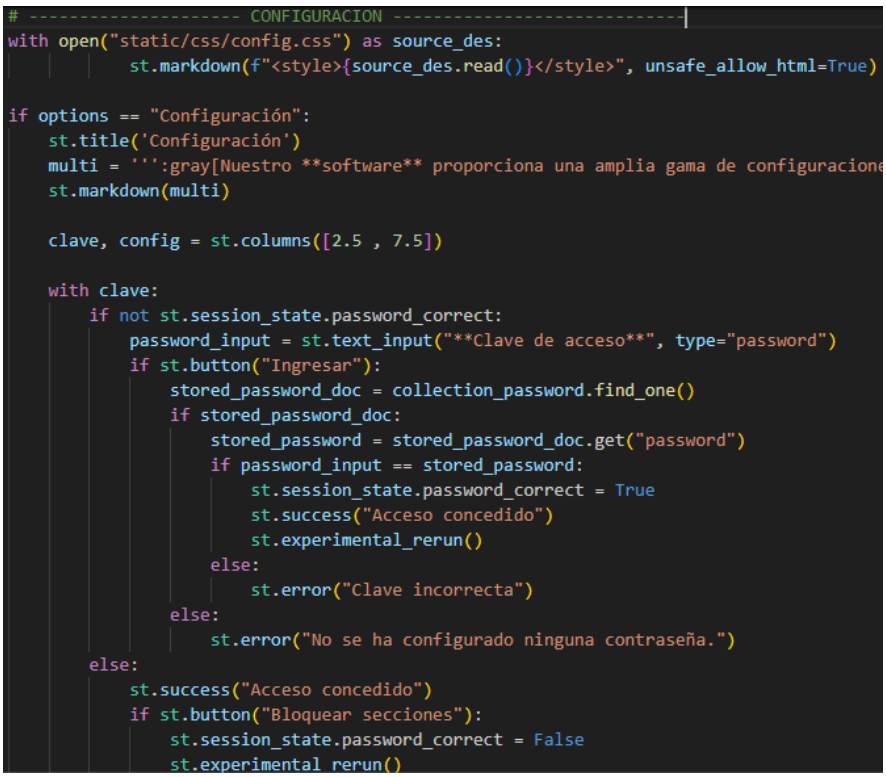
Palabras Clave: Permite agregar y eliminar palabras clave.

Modificar Contraseña: Permite cambiar la contraseña de acceso al sistema.

Modificar API Key: Permite cambiar la API Key utilizada para servicios externos, como OpenAI.

Funcionalidades Adicionales: Se incluyen mensajes de éxito (st.success) y errores (st.error), así como la recarga experimental (st.experimental_rerun) para actualizar la interfaz cuando se realizan cambios significativos.

Interacción con Bases de Datos: Utiliza operaciones de inserción, actualización y eliminación (insert_one, update_one, delete_one, delete_many) sobre colecciones MongoDB (collection_categories, collection, collection_faq, collection_keywords, collection_password, collection_settings) para gestionar categorías, PDFs, preguntas frecuentes, palabras clave, contraseñas y configuraciones de API.



```
# -------------------- CONFIGURACION ----------------------------
with open("static/css/config.css") as source_des:
            st.markdown(f"<style>{source_des.read()}</style>", unsafe_allow_html=True)

if options == "Configuración":
    st.title('Configuración')
    multi = '''gray[Nuestro **software** proporciona una amplia gama de configuracione
    st.markdown(multi)

    clave, config = st.columns([2.5 , 7.5])

    with clave:
        if not st.session_state.password_correct:
            password_input = st.text_input("**Clave de acceso**", type="password")
            if st.button("Ingresar"):
                stored_password_doc = collection_password.find_one()
                if stored_password_doc:
                    stored_password = stored_password_doc.get("password")
                    if password_input == stored_password:
                        st.session_state.password_correct = True
                        st.success("Acceso concedido")
                        st.experimental_rerun()
                    else:
                        st.error("Clave incorrecta")
                else:
                    st.error("No se ha configurado ninguna contraseña.")
        else:
            st.success("Acceso concedido")
            if st.button("Bloquear secciones"):
                st.session_state.password_correct = False
                st.experimental_rerun()
```

Ilustración 12. Código de configuraciones de administrador

Menú, básico y avanzado

Carga de Estilos CSS: Utiliza st.markdown para cargar estilos personalizados desde un archivo CSS (main.css) en la sección "Inicio" de la aplicación.

Interfaz de Usuario Principal:

- Usa sac.segmented para mostrar dos opciones de modo: "Básico" y "Avanzado", permitiendo al usuario seleccionar entre ellos.
- Divide la pantalla en dos columnas principales: "chat" y "pdf".

Sección de Chat:

- Muestra una imagen (robot3.png) y un contenedor (chat_container) para el historial de chat.
- Permite al usuario enviar mensajes al asistente usando st.chat_input.
-

En el modo "Básico":

- Proporciona una lista desplegable para seleccionar una categoría de preguntas frecuentes (faq_type).
- Carga preguntas frecuentes relacionadas con la categoría seleccionada y muestra botones para cada pregunta.
- Al hacer clic en una pregunta, el sistema consulta una cadena de modelos de lenguaje (llm_chain) para obtener respuestas y destaca el contenido relevante en los documentos asociados.

En el modo "Avanzado":

- Muestra documentos PDF cargados en una lista desplegable según la categoría seleccionada.
- Permite seleccionar una categoría de preguntas frecuentes y muestra preguntas con respuestas similares.
-

Cambiar Modo de Chat: Cuando se cambia el modo de "Básico" a "Avanzado" o viceversa (selected_mode), actualiza el historial de chat y el contenido visible en función del modo seleccionado.

Sección de PDF:

- Muestra el contenido de los documentos PDF en una caja con desplazamiento vertical (pdf).
- Utiliza st.markdown para mostrar el contenido del PDF resaltado (highlighted_content).

Manejo de Errores: Captura errores potenciales, como problemas de autenticación (AuthenticationError) al interactuar con servicios externos, y muestra mensajes de error correspondientes.

```python
#  ---------------------------------- menu ----------------------------------------------
if options == "Inicio":
    with open("static/css/main.css") as source_des:
            st.markdown(f"<style>{source_des.read()}</style>", unsafe_allow_html=True)

    selected_mode = sac.segmented(
        items=[
            sac.SegmentedItem(label='Básico', icon='sliders'),
            sac.SegmentedItem(label='Avanzado', icon='layers'),
        ], align='center', radius='xl',color='red', divider=False
    )
    chat, pdf = st.columns([5.1 , 4.9])

    with chat:
        st.image("static/img/robot3.png")
        if st.session_state.llm_chain is None:
            with st.spinner("Cargando documentos desde la base de datos..."):
                try:
                    update_vectors()
                except AuthenticationError:
                    st.error("Error de autenticación. Verifica tu API key.")
        chat_container = st.container(height=200)
        user_input = st.chat_input("Envía un mensaje al asistente del ITP...", max_chars=200)

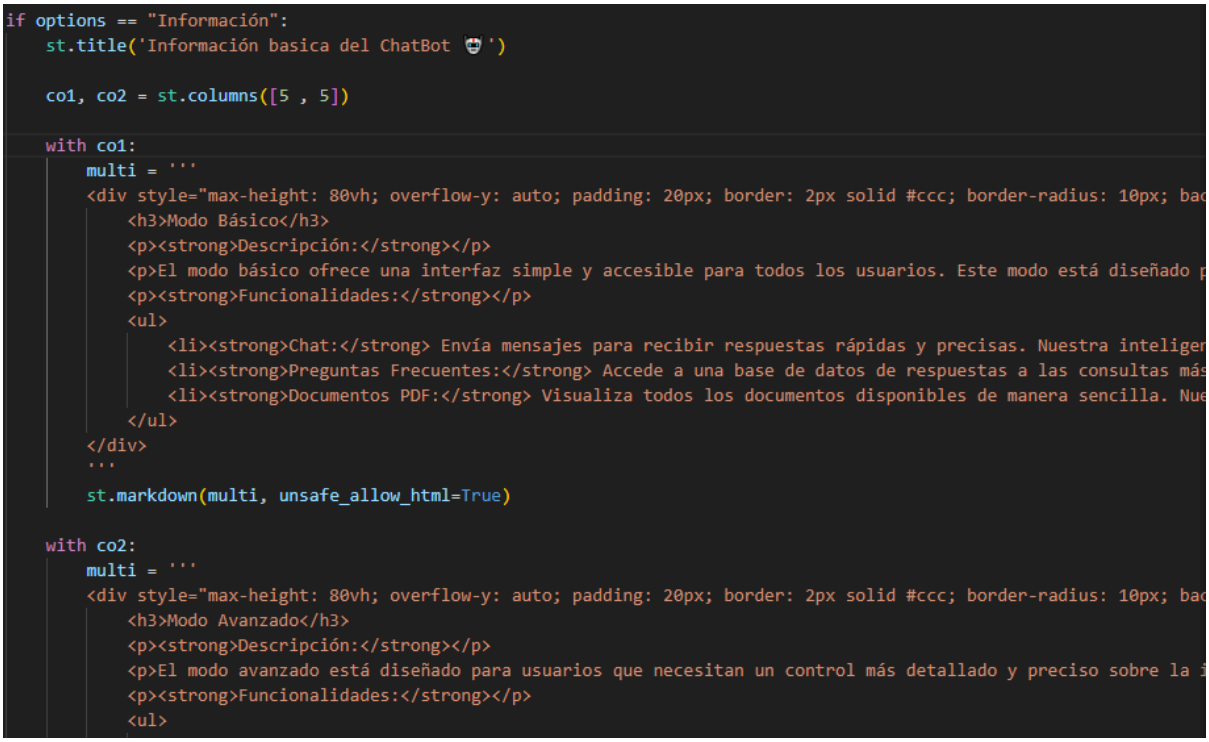
        #  ---------------------------------- basico ----------------------------------
        if st.session_state.mode == "basic":
            st.session_state.chat_history = st.session_state.basic_chat_history

            if not user_input:
                categories = get_all_categories()
```

Ilustración 13. Código de modos, básico y avanzado

Información

Este código proporciona información clara y estructurada sobre los modos de funcionamiento del chatbot en la aplicación, destacando las diferencias y las ventajas de usar el modo básico versus el modo avanzado.



```
if options == "Información":
    st.title('Información basica del ChatBot 🤖')

    co1, co2 = st.columns([5 , 5])

    with co1:
        multi = '''
        <div style="max-height: 80vh; overflow-y: auto; padding: 20px; border: 2px solid #ccc; border-radius: 10px; bac
            <h3>Modo Básico</h3>
            <p><strong>Descripción:</strong></p>
            <p>El modo básico ofrece una interfaz simple y accesible para todos los usuarios. Este modo está diseñado p
            <p><strong>Funcionalidades:</strong></p>
            <ul>
                <li><strong>Chat:</strong> Envía mensajes para recibir respuestas rápidas y precisas. Nuestra inteliger
                <li><strong>Preguntas Frecuentes:</strong> Accede a una base de datos de respuestas a las consultas más
                <li><strong>Documentos PDF:</strong> Visualiza todos los documentos disponibles de manera sencilla. Nue
            </ul>
        </div>
        '''
        st.markdown(multi, unsafe_allow_html=True)

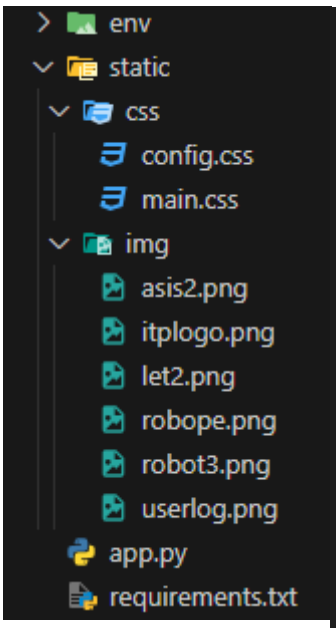
    with co2:
        multi = '''
        <div style="max-height: 80vh; overflow-y: auto; padding: 20px; border: 2px solid #ccc; border-radius: 10px; bac
            <h3>Modo Avanzado</h3>
            <p><strong>Descripción:</strong></p>
            <p>El modo avanzado está diseñado para usuarios que necesitan un control más detallado y preciso sobre la i
            <p><strong>Funcionalidades:</strong></p>
            <ul>
```

Ilustración 14. Información

CSS MEDIA-QUERIES

El código CSS está diseñado para personalizar la apariencia y el diseño de la interfaz, con especial atención a elementos como imágenes, encabezados, barras laterales, botones y áreas específicas, tales como el chat y los documentos PDF.

Se emplean dos archivos CSS diferentes: uno destinado a la página principal y otro específico para la página de configuraciones del administrador. Esto permite una gestión más organizada y eficiente del estilo visual en las distintas secciones de la aplicación.




```css
img {
    display: block;
    margin-left: auto;
    margin-right: auto;
    width: 60%;
}

.nombr{
    text-align: center;
}

h3{
    text-align: center;
}

/* barra superior */
[data-testid="stHeader"]{
    height:0px;
    background: rgba(255, 255, 255, 0);
}
/* logo */
[data-testid="stLogo"]{
    margin:auto;
}
/* espacio de chat */
[data-testid="stVerticalBlock"]{
    gap: 0.4rem;
}

/* ------------ input pregunta -------------- */
[data-testid="stChatInput"]{
    filter: drop-shadow(0px 2px 1px #000000a6);
```