

Pregrado

Carrera: Desarrollo de Software

Asignatura (UIC): Gestión de proyectos Informáticos

Trabajo de titulación previo a la obtención del

Título en: Tecnólogo Superior en Desarrollo
de Software

Tema: (AsIster) Desarrollo de un Chatbot integrando
un motor de inteligencia artificial para el asesoramiento
personalizado alojado en una aplicación web”

Autor/es: Guillermo Alejandro Alonso Pérez

Tutor: Mg. Yngrid Josefina Melo Quintana

Fecha: Octubre 2024



Autor:

Alonso Pérez Guillermo Alejandro



Título a obtener: Tecnología Superior en Desarrollo de Software

Matriz: Sangolquí -Ecuador

Correo electrónico:

Dirigido por:

Ing. Yngrid Josefina Melo Q. Mg.



Título: Ingeniero de Sistemas/master en Computación aplicada

Matriz: Sangolquí -Ecuador

Correo electrónico: yngrid.melo@ister.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

©2024 Tecnológico Universitario Rumi-

ñahui SANGOLQUÍ – ECUADOR

Alonso Pérez Guillermo Alejandro

(AsIster) Desarrollo de un Chatbot integrando un motor de inteligencia artificial para el asesoramiento personalizado alojado en una aplicación web”



CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-DES-2024-ISTER-6-6.2

Sangolquí, 14 de octubre del 2024

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

Por medio de la presente, yo, Guillermo Alejandro Alonso Pérez declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado “(AsIster) Desarrollo de un Chatbot integrando un motor de inteligencia artificial para el asesoramiento personalizado alojado en una aplicación web”, de la Tecnología Superior en Desarrollo de Software; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,



Guillermo Alejandro Alonso Pérez
C.I.: 171999050-7

FORMULARIO PARA ENTREGA DE PROYECTOS EN BIBLIOTECA INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-DES-2024-ISTER-1

CARRERA:
TECNOLOGIA SUPERIOR EN DESARROLLO DE SOFTWARE

AUTOR /ES:
GUILLERMO ALEJANDRO ALONSO PÉREZ

TUTOR:
YNGRID JOSEFINA MELO QUINTANA

CONTACTO ESTUDIANTE:
0983114905

CORREO ELECTRÓNICO:
Memomusic11.66@gmail.com

TEMA:
(ASISTER) DESARROLLO DE UN CHATBOT INTEGRANDO UN MOTOR DE
INTELIGENCIA ARTIFICIAL PARA EL
ASESORAMIENTO PERSONALIZADO ALOJADO EN UNA APLICACIÓN WEB

OPCIÓN DE TITULACIÓN:
UNIDAD DE INTEGRACIÓN CURRICULAR

RESUMEN EN ESPAÑOL:

El trabajo presenta el desarrollo de AsIster, un chatbot diseñado para brindar asistencia académica en el Instituto Tecnológico Superior Rumiñahui (ISTER). A través de inteligencia artificial y la API de OpenAI, AsIster responde preguntas sobre procesos académicos, ofertas educativas, mallas curriculares y costos de matrícula. La aplicación web que lo integra se ha desarrollado utilizando Laravel en el backend y Vue.js en el frontend, garantizando una interfaz moderna y funcional. El proyecto ha sido gestionado bajo metodologías ágiles, combinando Kanban y Extreme Programming para asegurar eficiencia y adaptabilidad durante el desarrollo del software. Con esta herramienta, se busca optimizar y modernizar la atención a estudiantes y usuarios en el ISTER, facilitando el acceso a información académica relevante de manera automatizada y eficiente.

PALABRAS CLAVE:

chatbot, inteligencia artificial, Laravel, Vue.js

ABSTRACT:

This project presents the development of a chatbot named AsIster, designed to assist users in obtaining academic information at Instituto Tecnológico Superior Rumiñahui (ISTER). By leveraging artificial intelligence and OpenAI's API, the chatbot responds to inquiries about academic processes, educational offerings, curriculum structures, and tuition costs. The system is integrated into a web application built with Laravel for the backend and Vue.js for the frontend. Agile methodologies such as Kanban and Extreme Programming were applied for project management and software development. The project aims to modernize and optimize user support within ISTER's academic environment.

PALABRAS CLAVE:

Chatbot, artificial intelligence, OpenAI, Laravel, Vue.js

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-DES-2024-ISTER-2
Sangolquí, 14 de octubre del 2024

Sres.-

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

A través del presente me permito aceptar la publicación del trabajo de titulación de la Unidad de Integración Curricular en el repositorio digital “DsPace” del estudiante: GUILLERMO ALEJANDRO ALONSO PÉREZ, con C.I.: 171999050-7 alumno de la Carrera DESARROLLO DE SOFTWARE.

Atentamente,



Firma del Estudiante
C.I.: 171999050-7

SÓLO PARA USO DEL ISTER

Han sido revisadas las similitudes del trabajo en el software “TURNITING” y cuenta con un porcentaje de; motivo por el cual, el Proyecto Técnico de Titulación es publicable. (EL PORCENTAJE DE SIMILITUD DEBE SER MÁXIMO DE 15%)

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

Dedicatoria:

Este proyecto es la culminación de un viaje personal de descubrimiento y aprendizaje. La música siempre ha sido una parte esencial de mi vida, y ahora, expandir mis habilidades más allá de lo conocido, en el mundo de la programación y la creación de proyectos, ha sido una promesa cumplida de mí para mí. Estos nuevos caminos en la tecnología, la investigación y la creatividad se han convertido en pilares fundamentales en mi desarrollo personal.

Dedico este proyecto a la banda, no a la banda musical, sino a la *banda* en su sentido más amplio: a la familia, los amigos, los músicos y todos los que me han acompañado en este viaje. A Alejandra Sauer, Carlos Alonso, Mariana Pérez, Tulo, Chino, Nathi, Sebas, Monchos, Monchitos, Monchita, Santiago, Jero, Pijo,Manu, Maribel—gracias por estar siempre ahí, por el apoyo, la inspiración y por compartir esta travesía conmigo.



Agradecimientos:

En mi vida personal, he emprendido muchos viajes a través del conocimiento, lo que me ha permitido explorar libremente las metodologías aplicadas en cada rama del saber. Estos viajes han sido fundamentales para forjar una visión propia sobre la experiencia que es la vida. Por ello, considero justo y necesario ofrecer un agradecimiento sincero a las personas que han sido parte de esta travesía.

Primero, agradezco a la familia Alonso Pérez, mi familia, por su constante certeza en mis capacidades, y por brindarme la confianza, el entendimiento y el apoyo necesarios para un desarrollo genuino y enfocado en la vivencia.

Mi agradecimiento también va a la familia Sauer Andrade, por el respaldo brindado desde el inicio, a lo largo y al final de esta experiencia de aprendizaje.

Un agradecimiento especial a Alejandra Sauer, quien ha sido mi compañera incondicional en mi desarrollo como estudiante y como persona, convenciéndome de mis propias convicciones y capacidades personales.

Finalmente, expreso mi gratitud a los Ingenieros involucrados directamente en mi formación en el Ister: Yngrid Melo, Julio Gancino, Danny Páez y Jorge Chapaca, por brindarme conocimientos útiles, prácticos y esenciales para entender las diversas tecnologías.



Resumen:

El presente trabajo describe el desarrollo de un chatbot llamado AsIster, cuyo objetivo es asistir a los usuarios en la obtención de información académica en el Instituto Tecnológico Superior Rumiñahui (ISTER). Utilizando inteligencia artificial y la API de OpenAI, el chatbot responde a consultas relacionadas con procesos académicos, ofertas educativas, mallas curriculares y costos de matrícula. El sistema está integrado en una aplicación web desarrollada con Laravel para el backend y Vue.js para el frontend. Además, se han aplicado metodologías ágiles como Kanban y Extreme Programming para la gestión y desarrollo del software. El proyecto busca modernizar y optimizar la atención al usuario dentro del entorno académico del ISTER.

Palabras claves: ASISTER, chatbot, inteligencia artificial, Laravel, Vue.js, gestión académica, ISTER, automatización, tecnología educativa.



Abstract:

This project presents the development of a chatbot named AsIster, designed to assist users in obtaining academic information at Instituto Tecnológico Superior Rumiñahui (ISTER). By leveraging artificial intelligence and OpenAI's API, the chatbot responds to inquiries about academic processes, educational offerings, curriculum structures, and tuition costs. The system is integrated into a web application built with Laravel for the backend and Vue.js for the frontend. Agile methodologies such as Kanban and Extreme Programming were applied for project management and software development. The project aims to modernize and optimize user support within ISTER's academic environment.

Keywords: Chatbot, artificial intelligence, OpenAI, Laravel, Vue.js, academic management, automation, web technologies, ISTER.



Contenido

Dedicatoria:.....	2
Agradecimientos:.....	4
Resumen:.....	5
Abstract:.....	6
1. El Problema de Investigación.....	11
1.1. Antecedentes del problema.....	11
1.2. Planteamiento del Problema. (Formulación problemática).....	13
1.3. Objetivos.....	14
1.3.1. General.....	14
1.3.2. Específicos.....	14
1.4. Justificación. (Legal, social, técnica, académica).....	15
1.5. Alcance.....	17
1.5.1. Resultados esperados.....	17
2. Marco Teórico.....	18
2.1. Fundamentaciones.....	18
2.1.1. Fundamentación Legal para Chatbots en Ecuador.....	18
2. Código de la Producción, Comercio e Inversiones.....	19
2.1.2. Fundamentación Teórica.....	19
2.2. Chatbot.....	21



2.2.1.	Definiciones.....	21
2.2.1.	Chatbots y sus tipos	22
2.2.2.	Aplicaciones	24
2.2.3.	Ventajas y desventajas	25
2.3.	Inteligencias artificiales y chatbots.....	27
2.3.1.	Conceptos Fundamentales de Inteligencia Artificial.....	27
2.3.1.	Motores de Inteligencia Artificial para Chatbots.	27
2.4.	Plataformas y Herramientas de Desarrollo para Chatbots	28
2.4.1.	Plataformas de Desarrollo de Chatbots integrales.....	28
2.4.2.	Herramientas de Desarrollo.....	28
2.4.3.	Comparación de Características y Capacidades	31
2.5.	Uso de Lenguajes en la Creación de un Sitio Web.....	32
2.5.1.	HTML, CSS y JavaScript	32
2.5.2.	Frameworks.....	33
2.5.3.	Utilidad de los Frameworks en Proyectos Web	34
2.5.4.	Ejemplos de Frameworks en Desarrollo Web	35
2.6.	Arquitectura de Software para Aplicaciones Web.....	36
2.6.1.	Microservicios vs. Monolítico.....	36
2.6.2.	MVC (Modelo-Vista-Controlador).....	37
2.6.3.	Experiencia de Usuario (UX) y Diseño de Interfaz de Usuario (UI)	37



2.6.4.	Seguridad en Aplicaciones Web.....	39
2.7.	APIs en el Desarrollo de Chatbots.....	40
2.7.1.	Concepto y Utilidad de las APIs.....	40
2.7.2.	API de OpenAI y GPT	41
3.	Marco Metodológico.....	43
3.1.	Metodología de Investigación.....	43
3.1.1.	Tipo de Investigación.....	43
3.1.2.	Técnicas de recolección de información	43
	Instrumentos de recolección	43
	Tratamiento y análisis de la información	43
3.2.	Metodología de Desarrollo	44
3.2.1.	Kanban para la Gestión Documental	44
3.2.2.	Extreme Programming (XP) para el Desarrollo de Software.....	44
3.3.	Fases del Proyecto.....	45
3.3.1.	Fase de Investigación y Planificación.....	45
3.3.2.	Fase de Diseño.....	45
3.3.3.	Fase de Desarrollo	46
3.3.4.	Fase de Pruebas y Validación.....	47
3.4.	Gestión de Riesgos	47
3.5.	Paradigma de la Investigación	48



3.6.	Enfoque de la Investigación.....	48
3.7.	Población	48
4.	Resultados y discusión	49
4.1.	la Metodología Utilizada	49
	Organización del Trabajo con Kanban.....	49
	Desarrollo de Software con XP	50
4.1.1.	Historias de usuarios	51
4.1.2.	Casos de Uso	56
4.1.3.	Diagrama de Casos de uso	65
4.1.4.	Planificación de las Iteraciones	66
4.1.5.	Ejecución de las iteraciones	67
4.1.5.1.	Construcción de la Iteración 1	67
4.1.5.2.	Construcción de Iteración 2.....	78
	80
4.1.5.3.	Construcción de la Iteración 3.....	81
5.	Conclusiones.....	85
6.	Recomendaciones	86
7.	Anexos.....	87
8.	Bibliografía.....	90



1. El Problema de Investigación.

1.1. Antecedentes del problema.

La atención al cliente es imprescindible para las entidades, empresas y emprendimientos que aspiren mantener y mejorar continuamente su calidad de servicio. Dentro de este contexto la aplicación de chatbots han emergido como una solución pragmática y tecnológica para enriquecer esta área, brindando respuestas de manera automatizada y de forma ágil a las preguntas de los clientes. Es así que varios desarrollos de proyectos han surgido para investigar y aprovechar al máximo el potencial de los chatbots en la mejora continua de la experiencia del cliente. A continuación, se exponen algunos proyectos, cada uno dirigido a un campo particular en la aplicación de este tipo de sistemas automatizados.

(Rivas Villatoro, 2021) “PROPUESTA DE MEJORA EN LA GESTION DE LA ATENCION AL CLIENTE POR MEDIO DE UN ASISTENTE VIRTUAL TIPO CHATBOT PARA FORTALECER EL DESEMPEÑO DEL SERVICIO DEL PROYECTO DE EDUCACION CONTINUA”. El estudio se enfoca en mejorar el servicio al cliente proporcionado por el Proyecto de Educación Continua en la Sede Caribe de la Universidad de Costa Rica mediante el uso de un chatbot en plataformas digitales de mensajería. La investigación reconoce la necesidad de optimizar la gestión de consultas de los usuarios y sugiere la implementación de un chatbot como solución. Se analizan casos exitosos previos de asistentes virtuales en la atención al cliente, resaltando su eficacia y capacidad para generar ventajas competitivas. Se realiza una encuesta a los usuarios



y se llevan a cabo entrevistas con el personal del proyecto para comprender mejor las necesidades y diseñar una estrategia de mejora. El objetivo principal es desplegar un chatbot que pueda ofrecer respuestas automáticas a través de Facebook y WhatsApp, mejorando así la comunicación con los clientes y permitiendo atender varias consultas al mismo tiempo.

(Garibay Ornelas, 2020) “DISEÑO E IMPLEMENTACIÓN DE UN ASISTENTE VIRTUAL (CHATBOT) PARA OFRECER ATENCIÓN A LOS CLIENTES DE UNA AEROLINEA MEXICANA POR MEDIO DE SUS CANALES CONVERSACIONALES”. El proyecto en cuestión, explica y demuestra el proceso para la creación de un asistente virtual a través de la integración de un chat bot, con el propósito de solventar requerimientos y agilizar procedimientos en la atención al cliente de la empresa Club premier, optimizando y mejorando la experiencia del usuario al momento de presentar la necesidad de una interacción personalizada, habilitando mayor concurrencia en el servicio de atención a sus clientes. Posteriormente el estudio hace un enfoque de recolección de datos para focalizar los requerimientos más solicitados y postular medios por los cuales satisfacer estas necesidades. Por último, pero no en importancia el, estudio se enfoca en la implementación del chatbot a través de vías como whatsapp y Facebook además de un servicio web, para habilitar una amplia disponibilidad del servicio como tal.

(Rubio et al., 2022) “PROYECTO UBOT: ASISTENTE VIRTUAL PARA ENTORNOS VIRTUALES DE APRENDIZAJE”. EL enfoque de este artículo, es el de presentar y desarticular el funcionamiento de UBOT, un bot integrable a los entornos virtuales, con el propósito de entregar información académica y administrativa sobre procesos de la institución de manera clara y actualizada. Otorgando beneficios a los estudiantes a través del uso de esta herramienta en el ecosistema educativo. El proyecto cuenta ya con la integración y acabidad en un aula virtual, se encuentra en



fase de pruebas, permitiendo focalizar la etapa de correcciones, y de esta forma entregar una mejor experiencia de usuario a través de la optimización en la comunicación y las interacciones por parte del cuerpo institucional.

1.2.Planteamiento del Problema. (Formulación problemática)

En el contexto actual, donde la eficiencia de los servicios de atención al cliente es predominante, es necesario rebuscar entre las múltiples opciones que les permitan a las entidades ofrecer una mejoría en las experiencias con sus potenciales clientes o sus usuarios habituales. Esta en sí es una razón suficiente para comprender que el desarrollo de un chat bot personalizado le otorgaría grandes ventajas a un ambiente de atención personalizada donde de manera concurrente se presenten necesidades de tipo administrativas, institucionales y académicas.

El desarrollo de un bot personalizado para asistencia conlleva desafíos de toda índole como, operativos y de diseño que deben abordarse de manera efectiva para garantizar su eficacia en la mejora de procesos de comunicación y respuestas. Entre estos desafíos se incluyen la comprensión precisa de las necesidades y consultas de los usuarios, la personalización de respuestas según las preferencias y necesidades de la entidad a la cual se lo podría integrar.

Crear una integración fluida con sistemas y plataformas existentes, la adaptabilidad a diferentes contextos de interacción y la capacidad para ofrecer respuestas rápidas y precisas en tiempo real. Superar estos desafíos es esencial para garantizar que el chatbot cumpla con su propósito de mejorar la experiencia del usuario y optimizar los procesos de asistencia y comunicación en el ámbito específico a relacionarse.



1.3. Objetivos.

1.3.1. General

- Desarrollar un chatbot basado en inteligencia artificial que integre la tecnología de OpenAI y su API, para mejorar los procesos de comunicación y asistencia en el entorno académico del Instituto Tecnológico Superior Rumiñahui, garantizando precisión, seguridad y accesibilidad en las interacciones con los usuarios.

1.3.2. Específicos.

- Investigar y analizar las necesidades y preferencias de los usuarios mediante entrevistas, para diseñar un chatbot que responda eficazmente a sus consultas referentes a la atención al cliente dentro del entorno institucional del Instituto Tecnológico Superior Rumiñahui.
- Diseñar una arquitectura de chatbot modular y escalable, específicamente adaptada para la asistencia en servicio al cliente , que permita una fácil implementación y funcionamiento dentro de una aplicación web dedicada, garantizando su eficiencia y rendimiento óptimos.
- Realizar un modelo de procesamiento de lenguaje natural (NLP) utilizando la API de OpenAI, adaptado para comprender y responder preguntas administrativas, asegurando precisión y relevancia en las respuestas.
- Crear una interfaz de usuario intuitiva y amigable utilizando Bootstrap y Vue.js, que facilite la interacción de los usuarios con el chatbot, asegurando una experiencia de navegación fluida y accesible.



- Realizar pruebas exhaustivas de usabilidad y funcionalidad del chatbot, evaluando su desempeño en diferentes contextos, para garantizar que cumpla con los estándares de calidad y satisfaga las necesidades de los usuarios de manera efectiva.

1.4. Justificación. (Legal, social, técnica, académica)

La implementación de un chatbot personalizado en el entorno académico del Instituto Tecnológico Superior Rumiñahui se justifica desde diversas perspectivas: legal, social, técnica y académica.

Legal: La adopción de un chatbot debe cumplir con las normativas establecidas en la Ley Orgánica de Protección de Datos Personales (LOPDP) de Ecuador, que regula la recolección, almacenamiento y manejo de datos personales. Esta ley garantiza la privacidad y seguridad de la información de los usuarios, esencial en un entorno académico donde se maneja información sensible. Adicionalmente, es importante adherirse al Código de la Producción, Comercio e Inversiones y la Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos, que establecen los requisitos legales para la autenticidad y validez de las transacciones y servicios digitales.

Social: La integración de un chatbot contribuye significativamente a mejorar las interacciones y la accesibilidad a la información en el sector académico. Proporciona respuestas rápidas y precisas a consultas comunes, facilitando la comunicación entre los estudiantes y la institución. Esto no solo mejora la eficiencia del servicio de atención al cliente, sino que también fortalece la relación entre los estudiantes y la institución, creando un ambiente más accesible y receptivo.



Técnica: Desde un punto de vista técnico, el desarrollo de un chatbot implica la implementación de algoritmos avanzados y el uso de tecnologías de procesamiento de lenguaje natural (NLP) proporcionadas por la API de OpenAI. Además, requiere una arquitectura de sistema robusta que permita la recepción y procesamiento de peticiones de manera constante y eficiente. La elección de tecnologías web como HTML, CSS y JavaScript, junto con frameworks como Laravel y Vue.js, asegura un diseño de interfaz amigable y una experiencia de usuario óptima.

Académica: El desarrollo de un chatbot no solo promueve la innovación tecnológica dentro del entorno académico, sino que también facilita la investigación y el aprendizaje en el campo de la inteligencia artificial. Este proyecto servirá como recurso educativo para estudiantes y docentes, permitiéndoles explorar y comprender mejor las aplicaciones prácticas de la IA. Además, el chatbot ayudará a automatizar tareas rutinarias y repetitivas, liberando tiempo para que el personal académico se enfoque en actividades de mayor valor añadido.

Justificación general

El desarrollo de este chatbot se justifica plenamente como una herramienta que beneficiará significativamente al servicio de atención al cliente. En primer lugar, los chatbots pueden estar disponibles las 24 horas del día, los 7 días de la semana, proporcionando asistencia continua sin importar la hora o la ubicación geográfica del usuario. Esto es crucial en el entorno académico, donde los estudiantes y personal pueden necesitar asistencia fuera del horario normal de oficina. Además, los chatbots pueden procesar consultas y solicitudes al instante, reduciendo los tiempos de espera y mejorando la satisfacción del cliente al proporcionar respuestas precisas y ágiles.



La automatización de tareas rutinarias mediante chatbots también disminuye la necesidad de personal dedicado a estas funciones, lo que se traduce en ahorros financieros para la institución. La escalabilidad de los chatbots permite manejar múltiples conversaciones simultáneamente sin comprometer la calidad del servicio, lo que es especialmente útil durante períodos de alta demanda, como inscripciones o periodos de exámenes.

Los chatbots también pueden ofrecer personalización inteligente, recopilando datos sobre los usuarios y utilizando esa información para personalizar las interacciones. Pueden ofrecer recomendaciones basadas en preferencias individuales, mejorando así la experiencia del usuario. Sin embargo, es importante tener en cuenta que, aunque los chatbots han mejorado significativamente, aún pueden tener dificultades para entender el contexto y el tono de una conversación, y carecen de la empatía y habilidades sociales que solo un ser humano puede ofrecer. A pesar de estas limitaciones, los beneficios superan con creces las desventajas, haciendo que la implementación de un chatbot sea una decisión estratégica y beneficiosa para mejorar el servicio de atención al cliente en el entorno académico.

1.5. Alcance

Lograr la interacción efectiva a través de la integración de un motor disponible de inteligencia artificial, de manera que pueda ser pre entrenado para otorgar respuestas un contexto en específico, a través de un correcto uso del entorno de desarrollo y la modificación de los parámetros focalizados en cumplir tanto con buenas prácticas y optimizaciones del sistema general.

1.5.1. Resultados esperados

Se pretende desarrollar un chat bot a través de la integración de un motor disponible y desarrollado, basado en lenguaje humanizado existente, con el fin de pre entrenar este modelo para



que otorgue respuestas específicas sobre temas dentro del contexto de entramiento, posteriormente realizar las evaluaciones correspondientes para determinar tanto su coherencia como su precisión.

Analizar los resultados de eficiencia en tiempo y respuestas para determinar el uso de recursos necesarios, permitiendo tener un panorama claro sobre la escalabilidad ante un mayor número de interacciones.

2. Marco Teórico

2.1. Fundamentaciones

2.1.1. Fundamentación Legal para Chatbots en Ecuador

En el constante y evolutivo panorama digital, donde los chatbots se involucran con muchas de nuestras interacciones cotidianas, Ecuador no permanece ajeno a la necesidad de establecer directrices legales. En este contexto y de manera fundamentada, se exploran las siguientes regulaciones específicas que afectan el desarrollo y posible despliegue de chatbots en suelo ecuatoriano.

1. Ley Orgánica de Protección de Datos Personales (LOPD)

La LOPD emerge como un faro de protección en el océano de datos personales. Su propósito es salvaguardar la privacidad de los individuos al regular la recopilación, procesamiento y almacenamiento de información personal. Al desarrollar un chatbot, es imprescindible apearse a los principios y procedimientos establecidos por esta ley, garantizando así el respeto a los derechos de los usuarios (Ley Orgánica de Protección de Datos Personales., 2021).



2. Código de la Producción, Comercio e Inversiones

Este código, regula las actividades comerciales y de servicio en Ecuador. Aquí, los chatbots se convierten en actores clave en la atención al cliente y la prestación de servicios digitales. De manera tal que la transparencia, la calidad del servicio y la protección al consumidor son principios que deben sostener al chatbot en su interacción con los usuarios (Código Orgánico de la Producción, Comercio e Inversiones. Registro Oficial, 2010).

3. Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos

En este capítulo, la ley exige que sus mensajes sean auténticos y válidos, como sellos oficiales en un contrato. Al implementar chatbots para transacciones comerciales o servicios, se debe asegurar que cumplan con los requisitos legales establecidos en esta ley (Ley de Comercio Electrónico, Firmas y Mensajes de Datos., 2002).

Por consiguiente, estas regulaciones específicas delimitan el camino legal para los chatbots en Ecuador. Como cartógrafos modernos, debemos trazar rutas seguras y éticas, siempre consultando las fuentes oficiales y actualizadas.

2.1.2. Fundamentación Teórica

Bots

Los bots, también conocidos como agentes virtuales o programas de software automatizados, desempeñan un papel importante en la automatización de tareas y la interacción con los usuarios en diversos entornos digitales. Este tipo de sistemas informáticos, aunque en ocasiones son asociados con la inteligencia artificial (IA), abarcan una amplia gama de funciones y capacidades que van más allá de la simple ejecución de comandos predefinidos.



Evolución y Diversidad de los Bots

Aunque generalmente asociados con la IA, es importante destacar que no todos los bots necesariamente incorporan inteligencia artificial en su funcionamiento. Algunos se basan en scripts simples que siguen instrucciones predefinidas para llevar a cabo acciones específicas.

Además, los bots no se limitan al procesamiento de lenguaje natural (NLP) ni a las interfaces de texto. Aunque el NLP permite a los bots comprender y generar lenguaje humano, existen bots que funcionan sin esta capacidad, como los crawlers, encargados de recopilar información de diversas fuentes, y los bots informativos, que gestionan datos en canales específicos.

Aplicaciones de los Bots

La versatilidad de los bots se refleja en su diversidad de tipos y aplicaciones. Desde los chatbots, diseñados para simular conversaciones humanas y proporcionar información o resolver dudas, hasta los bots transaccionales, que actúan como intermediarios en procesos financieros o comerciales, el panorama de los bots es amplio y variado.

Además, los bots se han integrado en diversos sectores gracias a su disponibilidad y facilidad de uso. Plataformas como Facebook Messenger o Telegram han facilitado su adopción en una variedad de casos de uso, desde la atención al cliente hasta la automatización de procesos empresariales.



2.2. Chatbot

2.2.1. Definiciones

Se puede definir como

Un sistema de diálogo busca interactuar con los usuarios mediante intercambio de mensajes, con el objetivo de simular una comunicación natural. A menudo se utiliza como la primera interacción con clientes o en soporte técnico. Sin embargo, en la actualidad, su uso se ha extendido a la educación, especialmente con estudiantes y en sistemas de aprendizaje a distancia (Paz et al., 2017)

También ha sido definido como

“Según una redefinición, un chatbot es un programa informático que utiliza inteligencia artificial y procesamiento del lenguaje natural para automatizar respuestas y simular conversaciones humanas (Aquino & Adaniya, 2018)”

Sin embargo, según algunas consideraciones en el ámbito educativo podemos tomar en cuenta la siguiente definición entregada en una conferencia

Un chatbot es un software que interactúa de una manera más natural con las personas y permite automatizar procesos a la vez que acerca el conocimiento de una entidad. Para realizar esta labor, los Chatbots pueden utilizar Inteligencia Artificial para el procesamiento del lenguaje (Garrido Ortega, 2018)

De manera sintetizada podemos concordar en que, los chatbots son sistemas de diálogo o programas informáticos que en su mayoría utilizan inteligencia artificial y procesamiento del lenguaje natural para interactuar con usuarios mediante intercambio de mensajes, con el propósito de simular conversaciones humanas.



Generalmente han sido empleados en atención al cliente y soporte técnico, su uso se ha expandido al ámbito educativo, especialmente en sistemas de aprendizaje a distancia. Estos sistemas permiten automatizar respuestas y procesos, acercando el conocimiento de una entidad de manera más natural a los usuarios.

2.2.1. Chatbots y sus tipos

Aunque regularmente se considera a un chatbot como un sistema automatizado de respuestas y más aún en la actualidad se los asocia directamente a sistemas que integran inteligencia artificial, esto no es del todo cierto puesto a que existen tipos de chatbots que cumplen varias funciones, lógicamente focalizados en la contemplación de cuestionamientos, sin embargo, el enfoque que adoptan en base a su procesamiento es el que difiere.

A continuación, se muestran los tipos de chatbots que existen con una descripción de su funcionamiento.

Chatbots Basados en Reglas:

Estos chatbots son relativamente sencillos y operan siguiendo un conjunto predefinido de reglas. Responden a las entradas del usuario mediante respuestas predeterminadas que coinciden con patrones específicos, son útiles para tareas simples y predecibles, pero pueden ser limitados en su capacidad para comprender preguntas complejas o contextos variables (Church, 2023).

Chatbots Híbridos:

Estos chatbots combinan características de los basados en reglas y en IA. Utilizan reglas predefinidas para interacciones simples y recurren a la IA para manejar consultas más complejas o inesperadas. Esta combinación permite una experiencia de usuario más fluida y adaptable (Rahman, 2023).



Chatbots Contextuales:

Los chatbots contextuales tienen la capacidad de recordar el contexto de una conversación anterior y utilizar esa información para proporcionar respuestas más personalizadas y relevantes. Esto permite una experiencia de usuario más natural y coherente a lo largo de una conversación (Dilmegani, 2024).

Chatbots Basados en Asistentes Virtuales:

Estos chatbots están diseñados para proporcionar una amplia gama de servicios y funciones, como programación de citas, reservas, búsqueda de información, entre otros. Suelen integrarse con otros sistemas y servicios para ofrecer una experiencia más completa y útil (Dharmalingam, 2020).

Chatbots Específicos de Dominio:

Están diseñados para funcionar en un dominio o industria específica, como atención al cliente, ventas, educación, salud, entre otros, están optimizados para comprender y responder a consultas relacionadas con ese dominio particular (Dilmegani, 2024).

Chatbots Basados en Menús o Botones:

Estos chatbots son los más básicos. Los usuarios interactúan con ellos haciendo clic en opciones de un menú predefinido. Funcionan como árboles de decisión, donde cada clic lleva a otra serie de opciones hasta llegar a la más adecuada.

Aunque útiles para responder preguntas sencillas y repetitivas, pueden tener dificultades con solicitudes más complejas. Además, no permiten la entrada de texto libre (Church, 2023).



Chatbots Lingüísticos o Basados en Procesamiento de Lenguaje Natural (NLP):

Estos chatbots utilizan NLP para comprender y generar lenguaje humano. Pueden mantener conversaciones más naturales y contextuales. Son adecuados para atención al cliente y tareas más complejas(Rahaman, 2023).

Chatbots Basados en Inteligencia Artificial (IA):

Estos chatbots utilizan algoritmos de procesamiento de lenguaje natural (NLP) y aprendizaje automático para comprender y responder a las consultas de los usuarios.

Son más avanzados que los chatbots basados en reglas y pueden adaptarse mejor a diferentes contextos y conversaciones más complejas(Rahaman, 2023)

2.2.2. Aplicaciones

Dentro de su pragmatismo se puede encontrar como constante, el uso de los chatbots como asistentes que facilitan, optimizan y mejoran la realización de peticiones, es por esta razón que sus aplicaciones van a estar enfocadas netamente en esas funciones, en brindar una asistencia real para diferentes ambientes y panoramas

A continuación, se encuentran las aplicaciones a las que pueden ser sometidos estos sistemas:

Atención al Cliente

Los chatbots son expertos en resolver consultas comunes y ofrecer asistencia rápida. Desde rastrear pedidos hasta proporcionar detalles sobre productos o servicios, alivian la carga del servicio al cliente y mejoran la satisfacción al proporcionar respuestas precisas y ágiles.

Ventas y Marketing



En el ámbito de las ventas y el marketing, los chatbots interactúan con los clientes de manera personalizada. Ofrecen recomendaciones basadas en preferencias pasadas y facilitan el proceso de compra. Además, envían alertas de precios o promociones personalizadas, aumentando la participación del cliente y mejorando las tasas de conversión.

Automatización de Tareas Rutinarias

Desde programar citas hasta procesar formularios, los chatbots liberan tiempo valioso. Esta automatización aumenta la eficiencia operativa y permite que el personal se concentre en tareas estratégicas y de alto valor.

Gestión de Leads y Calificación

En la generación de leads y la gestión de clientes potenciales, los chatbots desempeñan un papel crucial al calificar leads y determinar su nivel de interés. Esto es especialmente útil en industrias como la automotriz y las ventas en general, donde la identificación y el seguimiento de leads pueden ser procesos laboriosos y consumidores de tiempo.

2.2.3. Ventajas y desventajas

Ventajas

Disponibilidad Continua

Los chatbots pueden estar disponibles las 24 horas del día, los 7 días de la semana. Esto permite a las empresas ofrecer asistencia constante sin importar la hora del día o la ubicación geográfica del usuario.

Eficiencia y Respuestas Instantáneas

Los chatbots procesan consultas y solicitudes al instante, reduciendo los tiempos de espera. Esto mejora la satisfacción del cliente al proporcionar respuestas precisas y ágiles.



Reducción de Costos Operativos

La automatización de tareas rutinarias mediante chatbots disminuye la necesidad de personal dedicado a estas funciones. Esto se traduce en ahorros financieros para la empresa.

Escalabilidad

Los chatbots pueden manejar múltiples conversaciones simultáneamente sin comprometer la calidad del servicio. Esto es especialmente útil en momentos de alta demanda.

Personalización Inteligente

Los chatbots recopilan datos sobre los usuarios y utilizan esa información para personalizar las interacciones. Pueden ofrecer recomendaciones basadas en preferencias individuales.

Desventajas

Limitaciones en la Comprensión del Lenguaje

Aunque han mejorado significativamente, los chatbots aún pueden tener dificultades para entender el contexto y el tono de una conversación. Esto puede llevar a respuestas inexactas o malinterpretadas.

Falta de Empatía y Habilidades Humanas

Los chatbots carecen de emociones y habilidades sociales. No pueden ofrecer empatía o comprensión en situaciones complejas.



2.3. Inteligencias artificiales y chatbots

2.3.1. Conceptos Fundamentales de Inteligencia Artificial

La inteligencia artificial (IA) se refiere a la capacidad de las máquinas para imitar las funciones cognitivas humanas, como aprender y resolver problemas. Los sistemas de IA están diseñados para analizar datos, aprender de ellos y tomar decisiones o realizar acciones basadas en ese aprendizaje. Aquí hay algunas técnicas clave:

- **Aprendizaje Automático:** Esta técnica permite que las máquinas aprendan de los datos y mejoren con el tiempo sin necesidad de ser programadas específicamente para cada tarea.
- **Procesamiento de Lenguaje Natural (NLP):** Permite a las máquinas entender y generar lenguaje humano, haciendo posible que interactúen de manera más natural con nosotros.
- **Redes Neuronales Profundas:** Inspiradas en el cerebro humano, estas redes se usan para reconocer patrones y clasificar grandes cantidades de datos.

2.3.1. Motores de Inteligencia Artificial para Chatbots.

son sistemas que utilizan técnicas avanzadas de IA para permitir que los chatbots comprendan y respondan a las consultas de los usuarios de manera más natural y efectiva. Algunos de los motores de IA más utilizados para chatbots incluyen:

OpenAI GPT (Generative Pre-trained Transformer): Un modelo de lenguaje basado en redes neuronales profundas, capaz de generar texto coherente y relevante a partir de una entrada dada. OpenAI ha desarrollado varias versiones de GPT, siendo GPT-4 una de las más avanzadas, que puede comprender y generar texto en varios contextos.



Dialogflow: Una plataforma de desarrollo de chatbots propiedad de Google que utiliza técnicas de NLP para crear interacciones naturales y efectivas con los usuarios.

IBM Watson Assistant: Utiliza IA avanzada para comprender y responder a las consultas de los usuarios, permitiendo crear experiencias de chatbot personalizadas y contextuales.

2.4. Plataformas y Herramientas de Desarrollo para Chatbots

2.4.1. Plataformas de Desarrollo de Chatbots integrales

Estas plataformas proporcionan las herramientas necesarias para crear, entrenar y desplegar chatbots. Algunas de las más destacadas incluyen:

Bootpress: Ofrece una interfaz fácil de usar para diseñar y entrenar chatbots, con integraciones para múltiples canales de comunicación como Google Assistant y Facebook Messenger.

Microsoft Bot Framework: Un conjunto completo de herramientas para construir, probar, desplegar y gestionar chatbots en diversas plataformas.

IBM Watson Assistant: Permite crear chatbots con capacidades avanzadas de NLP y personalización.

2.4.2. Herramientas de Desarrollo

Además de las plataformas previamente mencionadas, hay varias herramientas y lenguajes de programación que son fundamentales para facilitar el desarrollo de chatbots. Estas herramientas permiten a los desarrolladores crear, probar y mantener aplicaciones de manera eficiente. A continuación, se presentan algunas de las más importantes:

Node.js



Es un entorno de ejecución para JavaScript que permite a los desarrolladores construir aplicaciones backend que sean escalables y eficientes. Node.js es particularmente útil debido a su capacidad para manejar múltiples conexiones simultáneamente con alto rendimiento, lo cual es crucial para chatbots que deben procesar muchas interacciones a la vez.

Axios

Esta es una librería de JavaScript diseñada para realizar solicitudes HTTP. Axios es especialmente útil para interactuar con APIs en aplicaciones web, ya que permite enviar y recibir datos de manera sencilla y eficiente. Esto es vital para los chatbots que necesitan comunicarse con servidores externos para obtener información o procesar datos.

Postman

Es una herramienta esencial para probar y documentar APIs. Postman permite a los desarrolladores enviar solicitudes a sus APIs y verificar que las respuestas sean correctas. Esto es fundamental en el desarrollo y pruebas de chatbots, ya que estos suelen integrarse con múltiples servicios externos y necesitan asegurar que todas las interacciones sean correctas y eficientes.

Visual Studio Code

Un editor de código fuente ligero pero potente, desarrollado por Microsoft. VS Code es muy popular entre los desarrolladores por su soporte para múltiples lenguajes de programación, su extensa colección de extensiones y su capacidad para integrarse con herramientas de control de versiones como Git.



Git

Una herramienta de control de versiones que permite a los desarrolladores gestionar y hacer seguimiento de los cambios en el código fuente. Git es crucial para la colaboración en proyectos de desarrollo de software, permitiendo a los equipos trabajar en paralelo y fusionar cambios de manera eficiente.

PHP

Un lenguaje de programación de uso general especialmente adecuado para el desarrollo web. Laravel, un framework de PHP, es ampliamente utilizado para desarrollar aplicaciones web robustas y escalables, y es ideal para el backend de los chatbots.

Python

Es un lenguaje de programación con una curva de aprendizaje amigable, ampliamente utilizado en inteligencia artificial y desarrollo de chatbots. Bibliotecas como NLTK y TensorFlow permiten a los desarrolladores implementar capacidades avanzadas de procesamiento de lenguaje natural e inteligencia artificial.

StackAI

Se refiere a un conjunto de herramientas y tecnologías utilizadas en conjunto para desarrollar aplicaciones de inteligencia artificial. En el contexto de chatbots, esto incluye el uso de APIs de IA como OpenAI GPT, junto con frameworks y herramientas de desarrollo web.



2.4.3. Comparación de Características y Capacidades

- **Facilidad de Uso:** La facilidad de uso es crucial, especialmente para desarrolladores que pueden no tener mucha experiencia técnica. Algunas plataformas, como Dialogflow, ofrecen una interfaz intuitiva que facilita el diseño y entrenamiento de chatbots. En cambio, otras plataformas como Microsoft Bot Framework pueden ser más complejas y requerir un nivel más alto de conocimientos técnicos.

- **Capacidades de NLP (Procesamiento de Lenguaje Natural):** Las capacidades de procesamiento de lenguaje natural varían significativamente entre plataformas. IBM Watson Assistant, por ejemplo, ofrece capacidades avanzadas de NLP que pueden entender y procesar lenguaje complejo. Por otro lado, aunque Dialogflow es más accesible y fácil de usar, puede no ser tan potente en ciertos aspectos de NLP.

- **Integraciones:** La capacidad de integrarse con múltiples canales de comunicación y servicios es esencial. Microsoft Bot Framework y Dialogflow son muy buenos en esto, ofreciendo amplias opciones de integración que permiten a los chatbots comunicarse a través de diversas plataformas como Google Assistant, Facebook Messenger, y más.

- **Personalización:** La posibilidad de personalizar el chatbot para que se adapte a necesidades específicas es otro factor crucial. IBM Watson Assistant permite un alto grado de personalización, lo que lo hace adecuado para aplicaciones que requieren interacciones muy específicas y detalladas.

- **Lenguajes de Programación Soportados:** La compatibilidad con diferentes lenguajes de programación es importante para elegir la herramienta adecuada. Por ejemplo, Node.js es excelente



para aquellos que prefieren trabajar con JavaScript, mientras que Python es ideal para quienes están más enfocados en la inteligencia artificial y el procesamiento de lenguaje natural.

2.5. Uso de Lenguajes en la Creación de un Sitio Web

2.5.1. HTML, CSS y JavaScript

Cuando se trata de construir un sitio web, hay tres pilares fundamentales que todo desarrollador web debe conocer: HTML, CSS y JavaScript. Estos lenguajes son como las herramientas de un carpintero, esenciales para crear una estructura sólida, darle una apariencia atractiva y hacer que todo funcione sin problemas.

HTML (HyperText Markup Language)

Piensa en HTML como el esqueleto de tu sitio web. Es el lenguaje que se utiliza para crear la estructura básica de una página web. HTML permite organizar el contenido en encabezados, párrafos, listas, enlaces, imágenes y otros elementos. Es como armar las vigas y columnas de una casa, estableciendo la base sobre la cual se construirá todo lo demás.

CSS (Cascading Style Sheets)

Si HTML es el esqueleto, CSS es la piel y la ropa que viste un sitio web. CSS se utiliza para darle el estilo y diseño necesario a las páginas web. Con CSS, se pueden definir colores, fuentes, tamaños, márgenes, y la disposición de los elementos. Es como decorar una casa, eligiendo los colores de las paredes, los muebles y la distribución de los espacios para que todo luzca atractivo y funcional.



JavaScript

Por otro lado, mientras que HTML y CSS establecen la estructura y el estilo, JavaScript es lo que da vida a tu sitio web. JavaScript este es un lenguaje de programación que permite añadir interactividad y dinamismo a las páginas web. Es con JavaScript, que se pueden crear efectos visuales, validar formularios, interactuar con los usuarios en tiempo real y, lo más importante para nuestro caso, integrar y gestionar chatbots. Es como los sistemas eléctricos y de plomería de una casa, que permiten que todo funcione de manera eficiente y responda a las necesidades de los habitantes.

2.5.2. Frameworks

Son herramientas casi esenciales en el desarrollo de software moderno, proporcionando una base estructurada y consistente sobre la cual los desarrolladores pueden construir aplicaciones eficientes y mantenibles.

En el contexto del desarrollo web, permiten manejar la complejidad de las aplicaciones modernas al ofrecer soluciones predefinidas para problemas comunes, promoviendo el uso de buenas prácticas y patrones de diseño.

Actúan como una estructura o esqueleto para el desarrollo de aplicaciones, similar a cómo un andamio proporciona soporte durante la construcción de un edificio. Ofrecen una serie de componentes reutilizables y un flujo de trabajo estandarizado que ayuda a los desarrolladores a evitar la reinención de la rueda en cada proyecto. Algunas de sus funciones incluyen:



Estandarización: Establecen una estructura común para el código, lo que facilita la colaboración entre desarrolladores y la mantenibilidad del proyecto a largo plazo. Es como tener un manual de instrucciones que asegura que todos los trabajadores en un proyecto de construcción sigan los mismos procedimientos.

Abstracción: Abstraen las complejidades subyacentes del desarrollo, proporcionando componentes y herramientas que simplifican tareas repetitivas. Por ejemplo, gestionar sesiones de usuario, validar formularios o interactuar con bases de datos se vuelve más sencillo con un framework que proporciona componentes preconstruidos para estas tareas.

Reutilización de Código: Al ofrecer componentes y módulos reutilizables, permiten a los desarrolladores enfocarse en las características únicas de su aplicación sin preocuparse por los aspectos básicos ya resueltos por el framework. Es similar a usar piezas de Lego, donde bloques estándar pueden ser reutilizados para construir diferentes estructuras.

Mejores Prácticas: Incorporan patrones de diseño y mejores prácticas probadas, asegurando que el código sea eficiente, seguro y escalable. Esto es comparable a seguir las normas de construcción y seguridad en un proyecto de construcción, lo que garantiza que el edificio sea estable y seguro para sus ocupantes.

2.5.3. Utilidad de los Frameworks en Proyectos Web

En el desarrollo web, son particularmente valiosos debido a la complejidad inherente de las aplicaciones modernas, que deben ser rápidas, seguras, y capaces de manejar grandes cantidades de datos. Algunos beneficios específicos de utilizarlos en proyectos web incluyen:

Aceleración del Desarrollo



Reducción significativa del tiempo de desarrollo al proporcionar componentes preconstruidos y herramientas que agilizan las tareas comunes. Esto permite a los desarrolladores entregar proyectos más rápido y con menos errores.

Escalabilidad

Diseñados para soportar el crecimiento de la aplicación, permitiendo añadir nuevas funcionalidades y manejar un mayor volumen de tráfico sin requerir una reescritura completa del código.

Seguridad

Los frameworks suelen incluir medidas de seguridad integradas que protegen contra amenazas comunes, como inyecciones SQL, XSS (Cross-Site Scripting), y CSRF (Cross-Site Request Forgery). Estas protecciones son como los sistemas de seguridad de un edificio, que aseguran la integridad y protección contra intrusos.

Mantenibilidad

La estructura estandarizada y el código limpio promovido por los frameworks facilitan la actualización y el mantenimiento de la aplicación a lo largo del tiempo. Esto es crucial para la sostenibilidad de proyectos a largo plazo.

2.5.4. Ejemplos de Frameworks en Desarrollo Web

Vue.js

Es un framework progresivo de JavaScript que se utiliza para construir interfaces de usuario interactivas y aplicaciones de una sola página (SPA). Vue.js es conocido por su facilidad de integración y su capacidad para crear componentes reutilizables. Al igual que un juego de construcción modular, Vue.js permite a los desarrolladores añadir y organizar componentes de manera flexible, lo que facilita la gestión de proyectos complejos y dinámicos. También ofrece una curva



de aprendizaje amigable, lo que lo hace accesible tanto para principiantes como para desarrolladores experimentados.

Laravel

Es un framework de PHP que se utiliza para el desarrollo de aplicaciones web robustas y escalables. Laravel proporciona una serie de herramientas y características integradas, como la gestión de sesiones, la validación de formularios, y la interacción con bases de datos a través de Eloquent ORM.

Laravel es como tener un equipo de construcción experimentado que se encarga de los detalles complejos del backend, permitiendo a los desarrolladores centrarse en la lógica de negocio y en la creación de funcionalidades avanzadas. también promueve el uso de buenas prácticas y patrones de diseño, lo que resulta en un código limpio y mantenible.

2.6. Arquitectura de Software para Aplicaciones Web

2.6.1. Microservicios vs. Monolítico

Arquitectura Monolítica: Este enfoque implica desarrollar la aplicación como una única unidad. Es como construir una casa de una sola planta, donde todos los elementos están interconectados. La ventaja de esta arquitectura es su simplicidad y facilidad de desarrollo inicial, pero puede volverse difícil de escalar y mantener a medida que la aplicación crece. En un enfoque monolítico, todas las funciones están dentro de una sola base de código, lo que puede llevar a problemas de dependencia y dificultades para implementar cambios.



Arquitectura de Microservicios: En contraste, los microservicios dividen la aplicación en servicios pequeños e independientes, cada uno con su propia lógica y base de datos. Es como construir un conjunto de casas en un vecindario, donde cada casa tiene su propio sistema. Esta arquitectura facilita la escalabilidad y el mantenimiento, ya que cada servicio puede ser desarrollado, desplegado y escalado de manera independiente. Los microservicios permiten una mayor flexibilidad y pueden ser gestionados por equipos diferentes, cada uno centrado en un servicio específico.

2.6.2. MVC (Modelo-Vista-Controlador)

Modelo: Gestiona los datos y la lógica de negocio de la aplicación. Es como el cerebro de la aplicación, encargándose de procesar la información y tomar decisiones. En Laravel, el modelo se utiliza para interactuar con la base de datos, definir relaciones entre datos y aplicar reglas de negocio.

Vista: Representa la presentación de los datos en la interfaz de usuario. Es como la cara de la aplicación, mostrando la información de manera atractiva y accesible. En Vue.js, la vista está compuesta por componentes que pueden ser reutilizados y organizados para crear interfaces de usuario dinámicas.

Controlador: Maneja la interacción del usuario, actualizando el modelo y la vista según sea necesario. Es como el sistema nervioso de la aplicación, transmitiendo las instrucciones del usuario al cerebro y mostrando los resultados. Los controladores en Laravel reciben las solicitudes del usuario, procesan la lógica necesaria y devuelven la respuesta apropiada.

2.6.3. Experiencia de Usuario (UX) y Diseño de Interfaz de Usuario (UI)

Principios de UX



Investigación del Usuario: Métodos para entender las necesidades y comportamientos de los usuarios. Es como realizar entrevistas y encuestas para conocer qué espera la gente de tu casa, asegurándote de que todo esté en el lugar correcto. La investigación del usuario puede incluir técnicas como entrevistas, encuestas y pruebas de usabilidad para recopilar información valiosa sobre las expectativas y problemas de los usuarios.

Diseño Centrado en el Usuario: Crear interfaces intuitivas y fáciles de usar. Es como diseñar una cocina donde cada utensilio y aparato esté en el lugar más conveniente para su uso. Este enfoque se asegura de que el diseño sea accesible, eficiente y agradable para los usuarios, reduciendo la curva de aprendizaje y mejorando la satisfacción general.

Principios de UI:

Diseño Responsivo: Técnicas para asegurar que las interfaces funcionen bien en diferentes dispositivos y tamaños de pantalla. Es como tener muebles ajustables que se adaptan al espacio disponible. Un diseño responsivo utiliza CSS flexible y media queries para adaptar el diseño a diversos tamaños de pantalla, garantizando una experiencia de usuario consistente en dispositivos móviles, tabletas y escritorios.

Accesibilidad: Asegurar que las aplicaciones sean utilizables por personas con discapacidades. Es como construir una rampa en lugar de solo escaleras para que todos puedan acceder a tu casa. La accesibilidad implica seguir pautas como las WCAG (Web Content Accessibility Guidelines) para asegurarse de que todos los usuarios, incluidos aquellos con discapacidades visuales, auditivas, motoras o cognitivas, puedan interactuar con la aplicación de manera efectiva.



2.6.4. Seguridad en Aplicaciones Web

Autenticación y Autorización

OAuth: Un estándar abierto para la autorización, permitiendo a los usuarios compartir información con terceros de manera segura sin revelar sus credenciales. OAuth es ampliamente utilizado para integrar aplicaciones con servicios de terceros como Google, Facebook y Twitter, permitiendo a los usuarios iniciar sesión utilizando sus cuentas existentes.

JWT (JSON Web Tokens): Métodos para la autenticación segura en aplicaciones web, proporcionando una forma compacta y segura de transmitir información entre partes como un token firmado digitalmente. Los JWT son útiles para aplicaciones modernas debido a su capacidad para ser transmitidos fácilmente en HTTP headers y su verificación basada en firmas criptográficas.

Protección contra Vulnerabilidades:

XSS (Cross-Site Scripting): Proteger las aplicaciones contra este tipo de ataques asegurando que los datos ingresados por el usuario no se ejecuten como código en el navegador. Esto se puede lograr mediante la validación y escape de todas las entradas del usuario antes de mostrar los datos en la interfaz.

SQL Injection: Técnicas para prevenir la inyección de SQL, como el uso de consultas preparadas y la validación adecuada de entradas, para asegurar que las consultas a la base de datos sean seguras. La inyección de SQL es una vulnerabilidad crítica que puede permitir a los atacantes manipular las consultas a la base de datos, causando daños o robando información sensible.



2.7. APIs en el Desarrollo de Chatbots

Las APIs (Interfaces de Programación de Aplicaciones) son componentes esenciales en el desarrollo de chatbots y aplicaciones web. Actúan como intermediarios que permiten que diferentes sistemas de software se comuniquen entre sí. Las APIs facilitan el intercambio de datos y funciones, permitiendo que los desarrolladores integren servicios externos y añadan nuevas funcionalidades a sus aplicaciones sin tener que construir todo desde cero.

2.7.1. Concepto y Utilidad de las APIs

Definición de API

Una API es un conjunto de reglas y protocolos que permite que una aplicación interactúe con otra. Proporciona una manera estandarizada para que los sistemas se comuniquen, definiendo las formas en que las solicitudes de datos deben ser realizadas y cómo las respuestas deben ser formateadas.

Utilidad de las APIs en el Desarrollo de Chatbots:

Integración de Servicios Externos: Las APIs permiten a los chatbots acceder a servicios externos, como bases de datos, servicios de traducción, y herramientas de análisis. Por ejemplo, un chatbot puede usar una API de clima para proporcionar información meteorológica en tiempo real a los usuarios.

Interacción con Usuarios: Las APIs permiten que los chatbots interactúen con plataformas de mensajería y redes sociales, facilitando la comunicación con los usuarios a través de múltiples canales. Esto incluye integraciones con servicios como Facebook Messenger, Slack, y WhatsApp.



Extensibilidad y Flexibilidad: Utilizar APIs permite a los desarrolladores ampliar las capacidades de los chatbots añadiendo nuevas funcionalidades de manera sencilla. En lugar de desarrollar funcionalidades complejas desde cero, los desarrolladores pueden aprovechar APIs existentes que ya han sido optimizadas y probadas.

2.7.2. API de OpenAI y GPT

La API de OpenAI, especialmente el modelo GPT (Generative Pre-trained Transformer), es una de las herramientas más avanzadas disponibles para el desarrollo de chatbots. OpenAI GPT utiliza técnicas de aprendizaje profundo para generar texto coherente y relevante en respuesta a las entradas del usuario.

OpenAI GPT:

Capacidades del Modelo: OpenAI GPT es capaz de entender y generar texto en varios idiomas y contextos, lo que lo hace altamente efectivo para una amplia gama de aplicaciones. GPT-4, la última versión del modelo, ha mejorado significativamente en términos de comprensión del contexto y generación de texto, lo que permite interacciones más naturales y contextuales.

Funcionamiento: GPT-4 utiliza una arquitectura de red neuronal profunda que ha sido preentrenada en una gran cantidad de texto. Esto le permite comprender el lenguaje natural y generar respuestas coherentes basadas en el contexto de la conversación. Los desarrolladores pueden enviar una solicitud a la API con un texto de entrada, y GPT-4 responderá con una salida de texto generada.

Aplicaciones Prácticas: Los chatbots basados en GPT-4 pueden manejar una variedad de tareas, desde responder preguntas frecuentes y proporcionar asistencia al cliente hasta generar contenido



creativo y participar en conversaciones complejas. Por ejemplo, un chatbot educativo puede utilizar GPT-4 para explicar conceptos complejos a los estudiantes de manera accesible y comprensible.

Características de GPT-4:

Comprensión del Contexto: GPT-4 tiene una capacidad mejorada para entender el contexto de una conversación, lo que le permite generar respuestas más precisas y relevantes. Esto es especialmente útil en aplicaciones donde el contexto es crucial, como en el soporte técnico y la atención al cliente.

Generación de Texto Natural: La capacidad de GPT-4 para generar texto que suena natural y humano hace que los chatbots sean más atractivos y efectivos en sus interacciones con los usuarios. Esto mejora la experiencia del usuario y aumenta la eficacia del chatbot.

Versatilidad: GPT-4 puede ser utilizado en una amplia gama de industrias y aplicaciones, desde el comercio electrónico y la salud hasta la educación y el entretenimiento. Su capacidad para adaptarse a diferentes contextos y necesidades lo convierte en una herramienta valiosa para los desarrolladores de chatbots.



3. Marco Metodológico

3.1. Metodología de Investigación

3.1.1. Tipo de Investigación

El proyecto "AsIster" se enmarca en el paradigma constructivista, donde se busca entender y construir el conocimiento a partir de la interacción con el entorno y los usuarios. Este paradigma es adecuado para proyectos de desarrollo de software que requieren adaptaciones y mejoras continuas basadas en la retroalimentación de los usuarios.

3.1.2. Técnicas de recolección de información

Entrevistas: Se realizarán entrevistas estructuradas con usuarios para obtener información cualitativa detallada sobre sus experiencias y sugerencias.

Instrumentos de recolección

Guías de Entrevistas: Se elaborarán guías de entrevistas para estructurar las conversaciones y asegurarse de que se cubren todos los temas relevantes.

Tratamiento y análisis de la información

Análisis Cualitativo:

Análisis de Contenido: Se utilizará para identificar temas y patrones en las respuestas de las entrevistas.

Codificación Temática: Se llevará a cabo para organizar y clasificar los datos cualitativos en categorías significativas, facilitando la interpretación y el análisis de la información recolectada.



3.2. Metodología de Desarrollo

3.2.1. Kanban para la Gestión Documental

Objetivo: Utilizar Kanban para gestionar las actividades documentales y teóricas del proyecto, permitiendo una visualización clara del progreso y la organización de las tareas.

Implementación:

Tablero Kanban: Se utilizará un tablero Kanban en Notion, dividido en columnas que representarán el estado de las tareas: "To Do" (Por Hacer), "In Progress" (En Progreso), "In Review" (En Revisión) y "Done" (Hecho). Este tablero facilitará la visualización del flujo de trabajo y la priorización de tareas.

Tareas: Las tareas se distribuirán en las columnas correspondientes según su estado actual, lo que permitirá un seguimiento visual y una priorización efectiva de las actividades necesarias para el desarrollo documental del proyecto.

3.2.2. Extreme Programming (XP) para el Desarrollo de Software

Objetivo: Utilizar XP para el desarrollo del software, enfatizando prácticas como la programación en pares, el desarrollo iterativo y las pruebas continuas para asegurar la calidad del software.

Implementación:

Desarrollo Iterativo: El desarrollo se realizará en iteraciones cortas, permitiendo una retroalimentación continua y ajustes rápidos. Cada iteración incluirá planificación, diseño, codificación, pruebas y revisión.

Pruebas: Se implementarán pruebas unitarias, de integración y de sistema desde el inicio del desarrollo para detectar y corregir errores tempranamente. Las pruebas continuas asegurarán que el software cumpla con los requisitos especificados y funcione correctamente.



3.3.Fases del Proyecto

3.3.1. Fase de Investigación y Planificación

Objetivos:

Recolectar la información necesaria y definir los requisitos del sistema.

Establecer el alcance y los objetivos del chatbot AsIster.

Actividades:

Recolección de Información: Se recopilarán documentos PDF del Instituto Tecnológico Superior Rumiñahui que contengan información sobre procesos académicos, ofertas académicas, mallas curriculares, costos de matrículas y carreras. Además, se identificarán fuentes de datos adicionales que puedan ser útiles.

Definición del Alcance y Objetivos: Se detallarán las funcionalidades del chatbot y se establecerán objetivos específicos y métricas de éxito para el proyecto.

Elección de Tecnologías y Herramientas: Se confirmarán las tecnologías a utilizar, incluyendo Laravel, Vue.js, MongoDB/SQLSRV, Bootstrap, Breeze y Bootpress.

3.3.2. Fase de Diseño

Objetivos:

Diseñar la arquitectura del sistema y la interfaz de usuario.

Crear modelos de bases de datos y la estructura de la API.

Actividades:

Elaboración de Diagramas de Flujo y Wireframes: Se desarrollarán diagramas de flujo para representar la lógica del sistema y wireframes para el diseño de la interfaz de usuario.



Diseño de Modelos de Bases de Datos: Se diseñarán los modelos de bases de datos necesarios para almacenar la información que gestionará el chatbot.

Estructura de la API: Se definirá la estructura de la API que permitirá la comunicación entre el frontend y el backend del sistema.

Creación de Mockups y Wireframes de las Vistas: Se crearán mockups y wireframes detallados de las vistas del frontend para asegurar una experiencia de usuario coherente y atractiva.

3.3.3. Fase de Desarrollo

Objetivos:

Implementar el backend utilizando Laravel.

Desarrollar la interfaz de usuario con Vue.js y Bootstrap.

Integrar el modelo de NLP con la API de OpenAI.

Actividades:

Configuración del Entorno de Desarrollo: Se configurará el entorno de desarrollo necesario, incluyendo la instalación y configuración de Laravel y la base de datos.

Desarrollo de la API y Endpoints: Se desarrollarán los endpoints necesarios para que el chatbot pueda interactuar con la base de datos y otros servicios.

Implementación de la Lógica del Chatbot: Se integrará el modelo de NLP con la API de OpenAI y se implementará la lógica del chatbot para procesar las solicitudes de los usuarios.

Desarrollo del Frontend: Se desarrollará la interfaz de usuario utilizando Vue.js y Bootstrap, asegurando que sea responsiva y fácil de usar.



Pruebas Continuas: Durante esta fase, se realizarán pruebas unitarias y de integración para asegurar que cada componente funcione correctamente y se integre bien con el resto del sistema.

3.3.4. Fase de Pruebas y Validación

Objetivos:

Asegurar que el chatbot funcione correctamente y cumpla con los requisitos establecidos.

Realizar pruebas de usabilidad con usuarios finales para obtener retroalimentación y realizar ajustes necesarios.

Actividades:

Ejecución de Pruebas Unitarias y de Integración: Se realizarán pruebas exhaustivas del backend y frontend para detectar y corregir errores.

Pruebas de Usabilidad: Se llevarán a cabo pruebas de usabilidad con usuarios finales para evaluar la interacción con el chatbot y la satisfacción del usuario.

Recopilación de Retroalimentación: Se recopilará retroalimentación de los usuarios a través de entrevistas y focus groups, permitiendo realizar ajustes y mejoras en el sistema.

Corrección de Errores y Ajustes: Basado en la retroalimentación obtenida, se corregirán errores y se realizarán ajustes necesarios para mejorar la experiencia del usuario y la funcionalidad del chatbot.

3.4. Gestión de Riesgos

Identificación y Mitigación de Riesgos:

Riesgo Técnico: Posibles fallos en la integración de la API de OpenAI.



Mitigación: Realizar pruebas exhaustivas en un entorno de desarrollo antes del despliegue para identificar y corregir problemas potenciales.

Riesgo de Seguridad: Vulnerabilidades en la aplicación web.

Mitigación: Implementar medidas de seguridad como autenticación JWT, protección contra XSS y SQL Injection, y realizar auditorías de seguridad regulares para identificar y corregir vulnerabilidades.

3.5. Paradigma de la Investigación

El proyecto se enmarca en el paradigma constructivista, donde se busca entender y construir el conocimiento a partir de la interacción con el entorno y los usuarios. Este paradigma es adecuado para proyectos de desarrollo de software que requieren adaptaciones y mejoras continuas basadas en la retroalimentación de los usuarios.

3.6. Enfoque de la Investigación

Se utilizará un enfoque cualitativo para obtener una comprensión integral del impacto del chatbot AsIster. El enfoque cualitativo ayudará a entender las experiencias y percepciones de los usuarios.

3.7. Población

La población objetivo del estudio son los estudiantes y personal administrativo del Instituto Tecnológico Superior Rumiñahui. Este grupo incluye aquellos que utilizarán el chatbot para obtener información sobre procesos académicos, ofertas académicas, mallas curriculares, costos de matrículas y carreras.



4. Resultados y discusión

4.1. la Metodología Utilizada

El marco metodológico del proyecto "AsIster" describe los métodos y procedimientos que se utilizarán para desarrollar un chatbot integrado con la API de OpenAI, alojado en un sitio web utilizando Laravel y Vue.js. Este proyecto combinará metodologías Kanban y Extreme Programming (XP) para gestionar la documentación y el desarrollo del software, respectivamente. El propósito de este marco metodológico es proporcionar una guía clara y estructurada para asegurar que el proyecto se lleve a cabo de manera eficiente y efectiva.

Organización del Trabajo con Kanban

Tablero Kanban: Se utilizó un tablero Kanban en Notion, dividido en columnas que representan el estado de las tareas: "To Do" (Por Hacer), "In Progress" (En Progreso), "In Review" (En Revisión) y "Done" (Hecho). Este tablero facilitó la visualización del flujo de trabajo y la priorización de tareas.

Tareas: Las tareas se distribuirán en las columnas correspondientes según su estado actual, lo que permitirá un seguimiento visual y una priorización efectiva de las actividades necesarias para el desarrollo documental del proyecto.



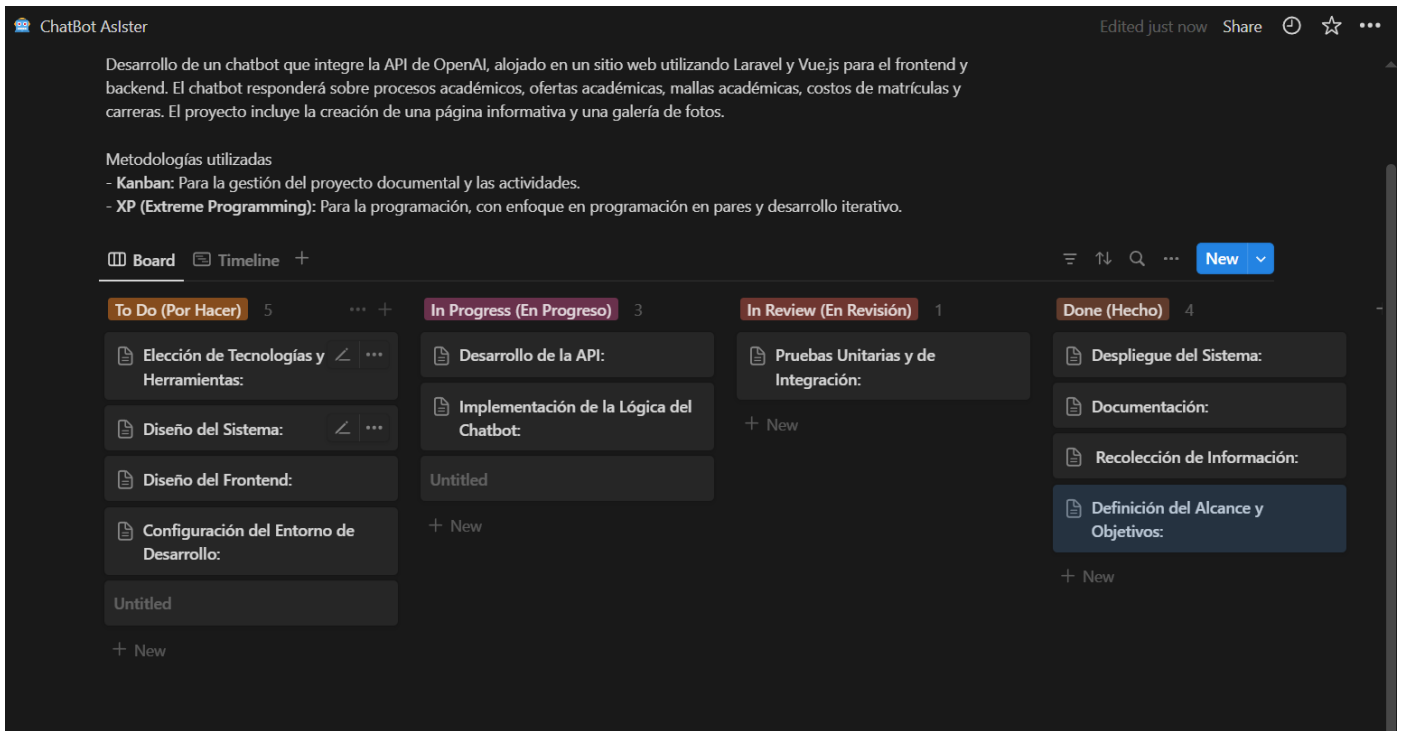


Fig 1. Tablero Kanban para visualización y gestión de tareas

Desarrollo de Software con XP

Desarrollo Iterativo: El desarrollo se realizará en iteraciones cortas, permitiendo una retroalimentación continua y ajustes rápidos. Cada iteración incluirá planificación, diseño, codificación, pruebas y revisión.

Pruebas: Se implementarán pruebas unitarias, de integración y de sistema desde el inicio del desarrollo para detectar y corregir errores tempranamente. Las pruebas continuas asegurarán que el software cumpla con los requisitos especificados y funcione correctamente.



4.1.1. Historias de usuarios

Para este proyecto, se han identificado las siguientes historias de usuario como críticas para la primera iteración. Estas historias cubrirán las funcionalidades básicas necesarias para registrar usuarios, permitirles iniciar sesión, interactuar con el chatbot, y para los administradores, gestionar el sistema a través de un panel de control.

Historia de Usuario 1: Registro de Usuario

TITULO	Registro de Usuario
NARRATIVA	
Como usuario, quiero registrarme en la plataforma para poder acceder a las funcionalidades del chatbot.	
CRITERIOS DE ACEPTACIÓN	
1. El formulario de registro debe solicitar nombre, correo electrónico y contraseña. 2. El usuario debe recibir una confirmación de registro exitoso.	



Historia de Usuario 2: Inicio de Sesión

TITULO	Inicio de Sesión
NARRATIVA	
<p>Como usuario registrado, quiero iniciar sesión para poder acceder a mi perfil y usar el chatbot.</p>	
CRITERIOS DE ACEPTACIÓN	
<p>El formulario de inicio de sesión debe solicitar correo electrónico y contraseña. El sistema debe validar las credenciales. El usuario debe ser redirigido a su perfil en caso de éxito o recibir un mensaje de error en caso de fallar.</p>	

Historia de Usuario 3: Interfaz de Usuario Inicial

TITULO	Interfaz de Usuario Inicial
NARRATIVA	
<p>Como usuario, quiero ver una interfaz amigable para interactuar con el chatbot para poder hacer preguntas y obtener respuestas de manera sencilla.</p>	
CRITERIOS DE ACEPTACIÓN	
<p>La interfaz debe tener un diseño limpio y fácil de usar. Debe incluir un campo de texto para ingresar preguntas y un área para mostrar las respuestas del chatbot. El chatbot debe responder de manera clara y coherente a las preguntas ingresadas.</p>	



Historia de Usuario 4: Acceso al Panel de Control

TITULO	Acceso al Panel de Control
NARRATIVA	
<p>Como administrador, quiero tener acceso a un panel de control para poder gestionar el sistema, incluyendo usuarios y estadísticas.</p>	
CRITERIOS DE ACEPTACIÓN	
<p>El panel de control debe mostrar estadísticas básicas del uso del chatbot. Debe permitir la administración de usuarios (crear, editar, eliminar). Debe incluir opciones para configurar parámetros del chatbot.</p>	

Historia de Usuario 5: Interacción con el Chatbot

TITULO	Interacción con el Chatbot
NARRATIVA	
<p>Como usuario, quiero hacer preguntas al chatbot para poder obtener respuestas útiles.</p>	
CRITERIOS DE ACEPTACIÓN	
<p>El chatbot debe poder procesar preguntas y dar respuestas útiles basadas en la información proporcionada. El tiempo de respuesta del chatbot debe ser razonable (menos de 3 segundos). Las respuestas deben ser claras y fáciles de entender.</p>	



Historia de Usuario 6: Gestión de Usuarios Avanzada

TITULO	Gestión de Usuarios Avanzada
NARRATIVA	
<p>Como administrador quiero poder gestionar los usuarios del sistema para tener control sobre quiénes pueden acceder a las funcionalidades del chatbot.</p>	
CRITERIOS DE ACEPTACIÓN	
<p>El administrador debe poder ver una lista de todos los usuarios registrados. El administrador debe poder crear, editar y eliminar usuarios. Las acciones de gestión de usuarios deben estar protegidas por permisos de administrador.</p>	

Historia de Usuario 7: Mejoras en la Interfaz de Usuario

TITULO	Mejoras en la Interfaz de Usuario
NARRATIVA	
<p>Como usuario, quiero una interfaz de usuario más intuitiva y atractiva para interactuar mejor con el chatbot.</p>	
CRITERIOS DE ACEPTACIÓN	
<p>La interfaz debe incluir mejoras visuales que faciliten la navegación y la interacción. Los cambios deben ser consistentes con la experiencia de usuario esperada en todas las plataformas.</p>	



Historia de Usuario 8: Implementación del Portafolio

TITULO	Implementación del Portafolio
NARRATIVA	
<p>Como usuario, quiero acceder a un portafolio que muestre la construcción del proyecto, información sobre el creador, y las tecnologías utilizadas para entender mejor la base del sistema y su desarrollo.</p>	
CRITERIOS DE ACEPTACIÓN	
<p>El portafolio debe ser accesible desde la barra de navegación y mostrar información clara y visualmente atractiva. El portafolio debe estar organizado en secciones que incluyan la construcción del proyecto, información del creador, y las tecnologías utilizadas.</p>	

Historia de Usuario 9: Mejora de la Navegación entre Rutas

TITULO	Mejora de la Navegación entre Rutas
NARRATIVA	
<p>Como usuario, quiero navegar fácilmente entre diferentes secciones de la aplicación sin experimentar demoras o problemas de carga.</p>	
CRITERIOS DE ACEPTACIÓN	
<p>La navegación entre diferentes rutas de la aplicación debe ser fluida y rápida. Las rutas deben estar correctamente definidas y manejar errores de navegación (como rutas no encontradas) de manera amigable.</p>	



4.1.2. Casos de Uso

Caso de Uso 1: Registro de Usuario	
ID:	CU-001
Título:	Registro de Usuario
Descripción:	Permitir a los nuevos usuarios registrarse en la plataforma proporcionando sus datos personales y de contacto.
Actor Principal:	Usuario
Precondiciones:	El usuario debe tener acceso a la página de registro.
Postcondiciones:	El usuario está registrado en el sistema y puede iniciar sesión.
Escenario de Éxito Principal:	<p>El usuario accede a la página de registro.</p> <p>El usuario completa el formulario de registro con su nombre, correo electrónico y contraseña.</p> <p>El sistema guarda los datos del usuario en la base de datos.</p> <p>El sistema notifica al usuario que el registro fue exitoso.</p>
Extensiones:	Si el correo electrónico ya existe en el registro, el sistema muestra un mensaje de error y solicita al usuario que ingrese un correo diferente.
Frecuencia de Uso:	Frecuente, cada vez que un nuevo usuario desea registrarse.
Prioridad:	alta

- El caso de uso "Registro de Usuario" describe el proceso mediante el cual un nuevo usuario puede registrarse en la plataforma. Este proceso implica que el usuario acceda a la página de registro, complete el formulario con su nombre, correo electrónico y contraseña, y que el sistema valide que el correo electrónico no esté en uso antes de guardar los datos en la base de datos y notificar al usuario que el registro fue exitoso. Esto permite a nuevos usuarios crear una cuenta para acceder a las funcionalidades del chatbot.



Caso de Uso 2: Inicio de Sesión	
ID:	CU-002
Título:	Inicio de Sesión
Descripción:	Permitir a los usuarios registrados iniciar sesión utilizando sus credenciales.
Actor Principal:	Usuario
Precondiciones:	El usuario debe estar registrado en el sistema.
Postcondiciones:	El usuario está autenticado y puede acceder a su perfil.
Escenario de Éxito Principal:	<p>El usuario accede a la página de inicio de sesión.</p> <p>El usuario ingresa su correo electrónico y contraseña.</p> <p>El sistema valida las credenciales del usuario.</p> <p>El sistema redirige al usuario a su perfil en caso de éxito.</p> <p>El sistema notifica al usuario que el inicio de sesión fue exitoso.</p>
Extensiones:	Si las credenciales son incorrectas, el sistema muestra un mensaje de error y solicita al usuario que intente nuevamente.
Frecuencia de Uso:	Frecuente, cada vez que un usuario registrado desea acceder a su cuenta.
Prioridad:	alta

- El caso de uso "Inicio de Sesión" detalla el procedimiento que permite a los usuarios registrados autenticarse en la plataforma. El usuario ingresa su correo electrónico y contraseña en la página de inicio de sesión. El sistema valida estas credenciales y, si son correctas, redirige al usuario a su perfil, notificándole que el inicio de sesión fue exitoso. En caso de credenciales incorrectas, el sistema mostrará un mensaje de error solicitando al usuario que lo intente nuevamente.



Caso de Uso 3: Interacción con el Chatbot	
ID:	CU-003
Título:	Interacción con el Chatbot
Descripción:	Permitir a los usuarios hacer preguntas al chatbot y recibir respuestas basadas en la API de OpenAI.
Actor Principal:	Usuario
Precondiciones:	El usuario debe haber iniciado sesión.
Postcondiciones:	El usuario recibe una respuesta a su pregunta.
Escenario de Éxito Principal:	El usuario accede a la interfaz del chatbot. El usuario ingresa una pregunta en el campo de texto. El sistema envía la pregunta a la API de OpenAI. La API procesa la pregunta y devuelve una respuesta. El sistema muestra la respuesta al usuario en la interfaz.
Extensiones:	Si la API de OpenAI no está disponible, el sistema muestra un mensaje de error.
Frecuencia de Uso:	Frecuente, cada vez que un usuario interactúa con el chatbot.
Prioridad:	alta

- El caso de uso "Interacción con el Chatbot" describe cómo un usuario puede interactuar con el chatbot después de haber iniciado sesión. El usuario ingresa una pregunta en el campo de texto de la interfaz del chatbot. La pregunta se envía a la API de OpenAI, que procesa la consulta y devuelve una respuesta que el sistema muestra al usuario. Este caso de uso garantiza que los usuarios puedan obtener respuestas útiles y precisas a sus preguntas mediante el chatbot.



Caso de Uso 4: Acceso al Panel de Control	
ID:	CU-004
Título:	Acceso al Panel de Control
Descripción:	Permitir a los administradores acceder a un panel de control para gestionar el sistema, incluyendo la visualización de estadísticas y la administración de usuarios.
Actor Principal:	Administrador
Precondiciones:	El usuario debe tener permisos de administrador.
Postcondiciones:	El administrador puede gestionar el sistema
Escenario de Éxito Principal:	El administrador accede al panel de control desde su perfil. El sistema muestra estadísticas básicas del uso del chatbot. El administrador selecciona la opción de administrar usuarios. El administrador puede crear, editar o eliminar usuarios. El administrador guarda los cambios realizados.
Extensiones:	Si el administrador no tiene permisos suficientes, el sistema muestra un mensaje de error.
Frecuencia de Uso:	Frecuente, cada vez que un administrador necesita gestionar el sistema.
Prioridad:	alta

- El caso de uso "Acceso al Panel de Control" explica cómo un administrador accede a un panel de control para gestionar el sistema. Desde su perfil, el administrador puede visualizar estadísticas del uso del chatbot, administrar usuarios (creando, editando o eliminando), y configurar parámetros del chatbot. Este caso de uso asegura que los administradores tengan las herramientas necesarias para gestionar eficientemente la plataforma y mantener su funcionamiento óptimo.



Caso de Uso 5: Preguntas y Respuestas del Chatbot	
ID:	CU-004
Título:	Preguntas y Respuestas del Chatbot
Descripción:	Permitir a los usuarios hacer preguntas al chatbot y obtener respuestas útiles.
Actor Principal:	Usuario
Precondiciones:	El usuario debe haber iniciado sesión.
Postcondiciones:	El usuario recibe una respuesta útil a su pregunta.
Escenario de Éxito Principal:	<p>El usuario accede a la interfaz del chatbot.</p> <p>El usuario ingresa una pregunta en el campo de texto.</p> <p>El sistema envía la pregunta a la API de OpenAI.</p> <p>La API procesa la pregunta y devuelve una respuesta.</p> <p>El sistema muestra la respuesta al usuario en la interfaz.</p>
Extensiones:	Si la API de OpenAI no está disponible, el sistema muestra un mensaje de error.
Frecuencia de Uso:	Frecuente, cada vez que un usuario necesita gestionar el sistema.
Prioridad:	alta

- El caso de uso "Preguntas y Respuestas del Chatbot" se enfoca en permitir a los usuarios hacer preguntas al chatbot y obtener respuestas útiles. Similar a la interacción general con el chatbot, este caso de uso asegura que las preguntas ingresadas por los usuarios sean procesadas por la API de OpenAI y que las respuestas sean mostradas rápidamente y de manera comprensible. Esto facilita una experiencia de usuario efectiva y satisfactoria con el chatbot.



Caso de Uso 6: Gestión de Usuarios	
ID:	CU-006
Título:	Gestión de Usuarios
Descripción:	Permitir al administrador gestionar a los usuarios del sistema.
Actor Principal:	Administrador
Precondiciones:	El administrador debe estar autenticado y tener permisos de administrador.
Postcondiciones:	Los usuarios pueden ser creados, editados o eliminados.
Escenario de Éxito Principal:	El administrador accede a la lista de usuarios. El administrador selecciona un usuario para editar o eliminar. El administrador puede crear un nuevo usuario, editar detalles de un usuario existente, o eliminar un usuario.
Extensiones:	Si el administrador no tiene permisos suficientes, se muestra un mensaje de error.
Frecuencia de Uso:	Frecuente
Prioridad:	alta

- Este caso de uso describe cómo el administrador del sistema puede gestionar los usuarios, permitiéndole crear, editar y eliminar usuarios dentro de la aplicación. Esto es primordial para mantener un control adecuado sobre quiénes tienen acceso a las funcionalidades del sistema, asegurando que solo los usuarios autorizados puedan interactuar con funciones críticas.



Caso de Uso 7: Mejoras en la Interfaz de Usuario	
ID:	CU-007
Título:	Mejoras en la Interfaz de Usuario
Descripción:	Este caso de uso se centra en la implementación de mejoras visuales y de usabilidad en la interfaz de usuario
Actor Principal:	Usuario
Precondiciones:	El sistema debe estar operativo y accesible.
Postcondiciones:	La interfaz es más amigable y fácil de usar.
Escenario de Éxito Principal:	El usuario accede a la aplicación y nota mejoras visuales en la interfaz. La navegación es más intuitiva y la interacción con la aplicación es más fluida.
Extensiones:	Si las mejoras visuales no se implementan correctamente, se realizan ajustes adicionales.
Frecuencia de Uso:	Constante
Prioridad:	Media

- Este caso de uso se centra en la implementación de mejoras visuales y de usabilidad en la interfaz de usuario, asegurando que la experiencia del usuario sea más intuitiva, atractiva y coherente en todas las plataformas. Esto incluye cambios en el diseño visual y mejoras en la navegación, lo que facilita la interacción de los usuarios con la aplicación.



Caso de Uso 8: Implementación del Portafolio	
ID:	8
Título:	Implementación del Portafolio
Descripción:	Este caso de uso aborda la creación e integración de un portafolio en la aplicación
Actor Principal:	Usuario, Administrador
Precondiciones:	El usuario debe estar autenticado.
Postcondiciones:	El usuario puede visualizar la información del portafolio.
Escenario de Éxito Principal:	El usuario selecciona la opción de portafolio en la barra de navegación. El sistema carga y muestra el portafolio con la información relevante.
Extensiones:	Si el portafolio no se carga correctamente, el sistema muestra un mensaje de error.
Frecuencia de Uso:	Frecuente
Prioridad:	Media

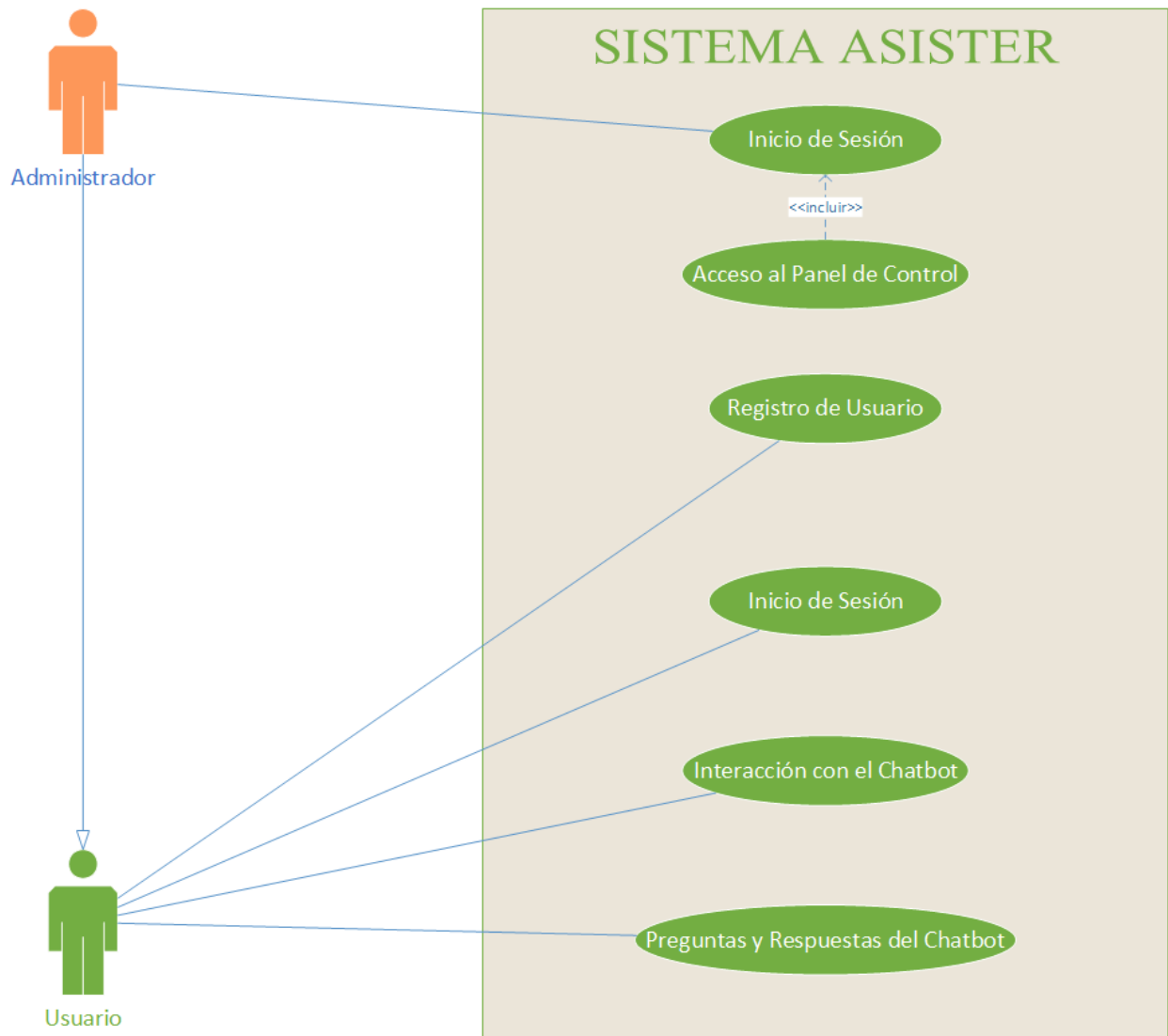
- Este caso de uso aborda la creación e integración de un portafolio en la aplicación, que muestra la construcción del proyecto, la información del creador y las tecnologías utilizadas. El portafolio proporciona a los usuarios una visión clara y organizada de los aspectos técnicos y creativos detrás del proyecto, accesible desde la barra de navegación



Caso de Uso 9: Mejora de la Navegación entre Rutas	
ID:	CU-009
Título:	Mejora de la Navegación entre Rutas
Descripción:	Este caso de uso se enfoca en mejorar la navegación entre las distintas rutas de la aplicación.
Actor Principal:	Usuario, Administrador
Precondiciones:	El sistema debe estar funcionando y accesible.
Postcondiciones:	El usuario puede navegar entre las diferentes rutas de la aplicación sin problemas.
Escenario de Éxito Principal:	<p>El usuario selecciona una opción en la barra de navegación. El sistema carga la vista correspondiente de manera rápida y fluida.</p>
Extensiones:	Si la ruta seleccionada no existe, el sistema redirige al usuario a una página de error amigable.
Frecuencia de Uso:	Frecuente
Prioridad:	Alta



4.1.3. Diagrama de Casos de uso



4.1.4. Planificación de las Iteraciones

Iteración 1

- Configuración del entorno de desarrollo.
- Implementación de las primeras funciones del backend con Laravel.
- Desarrollo de la interfaz de usuario inicial con Vue.js.
- Integración básica de la API de OpenAI.
- Pruebas unitarias iniciales.

Iteración 2

- Implementación de funcionalidades adicionales en el backend.
- Mejoras en la interfaz de usuario.

Iteración 3

- Finalización de todas las funcionalidades previstas.
- Optimización del rendimiento y la seguridad.
- Pruebas exhaustivas de usabilidad y funcionalidad.



4.1.5. Ejecución de las iteraciones

4.1.5.1. Construcción de la Iteración 1

Objetivos de la Iteración 1:

- Configuración del entorno de desarrollo.
- Implementación de las primeras funciones del backend con Laravel.
- Desarrollo de la interfaz de usuario inicial con Vue.js.
- Integración básica de la API de OpenAI.
- Pruebas unitarias iniciales.

Desarrollo:

- Configuración del Entorno de Desarrollo:
- Se configuró el entorno de desarrollo en base a las tecnologías seleccionadas: Laravel para el backend y Vue.js para el frontend.
- Se estableció la conexión con la base de datos utilizando SQL Server y se configuraron los archivos .env para gestionar las variables de entorno, incluyendo las claves de la API de OpenAI.



Evidencias:

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:MTF1cYpHEKU1pLuEfd3lwq0rq7yJ8cRc9BZKhoI+ia8=
APP_DEBUG=true
APP_URL=http://localhost:8000
FRONTEND_URL=http://localhost:8080

LOG_CHANNEL=stack
LOG_DEPRECATED_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=sqlsrv
DB_HOST=localhost
DB_PORT=1433
DB_DATABASE=Asister_DB
DB_USERNAME=sa
DB_PASSWORD=1223334444

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=cookie
SESSION_LIFETIME=120
SESSION_DOMAIN=localhost
SESSION_SECURE_COOKIE=false
SANCTUM_STATEFUL_DOMAINS=localhost:8080

MEMCACHED_HOST=127.0.0.1
```

Fig.2. Captura de pantalla del archivo .env con las configuraciones necesarias.



Implementación de las Primeras Funciones del Backend con Laravel:

- Se desarrollaron las rutas iniciales y los controladores básicos para la autenticación de usuarios.
- Se creó la lógica para el registro y la autenticación de usuarios utilizando Laravel.
- Se implementaron las primeras migraciones de base de datos para crear las tablas de usuarios.

Evidencias:

```
1  <?php
2
3  namespace App\Http\Controllers\Auth;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Auth;
8  use Illuminate\Validation\ValidationException;
9
10 class AuthenticatedSessionController extends Controller
11 {
12     public function store(Request $request)
13     {
14         $credentials = $request->validate([
15             'email' => ['required', 'email'], // Valida que el email sea requerido y tenga formato de email
16             'password' => ['required'], // Valida que la contraseña sea requerida
17         ]);
18
19         if (!Auth::attempt($credentials)) {
20             throw ValidationException::withMessages([
21                 'email' => __(''auth.failed''), // Lanza una excepción si las credenciales son incorrectas
22             ]);
23         }
24
25         $request->session()->regenerate(); // Regenera la sesión para evitar ataques de fijación de sesión
26
27         return response()->json([
28             'user' => Auth::user(), // Devuelve el usuario autenticado
29         ]);
30     }
31
32     public function destroy(Request $request)
33     {
34         Auth::guard('web')->logout(); // Cierra la sesión del usuario
35
36         $request->session()->invalidate(); // Invalida la sesión actual
37
38         $request->session()->regenerateToken(); // Regenera el token CSRF
39
40         return response()->json(null, 204); // Responde con un código 204 No Content
41     }
42 }
43
```

Fig.3. Código del controlador de autenticación de sesión



```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('email')->unique();
18             $table->timestamp('email_verified_at')->nullable();
19             $table->string('password');
20             $table->boolean('is_admin')->default(false);
21             $table->rememberToken();
22             $table->timestamps();
23         });
24     }
25
26     /**
27     * Reverse the migrations.
28     */
29     public function down(): void
30     {
31         Schema::dropIfExists('users');
32     }
33 };
34
```

Fig.4. Migración generada a través de ORM para crear la tabla de Usuarios en SQL SRV



```
1 <?php
2
3 namespace App\Http\Controllers\Auth;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\User;
7 use Illuminate\Auth\Events\Registered;
8 use Illuminate\Http\Request;
9 use Illuminate\Http\Response;
10 use Illuminate\Support\Facades\Auth;
11 use Illuminate\Support\Facades\Hash;
12 use Illuminate\Validation\Rules;
13
14 class RegisteredUserController extends Controller
15 {
16     /**
17      * Handle an incoming registration request.
18      *
19      * @throws \Illuminate\Validation\ValidationException
20      */
21     public function store(Request $request): Response
22     {
23         $request->validate([
24             'name' => ['required', 'string', 'max:255'],
25             'email' => ['required', 'string', 'lowercase', 'email', 'max:255', 'unique:'.User::class],
26             'password' => ['required', 'confirmed', Rules\Password::defaults()],
27         ]);
28
29         $user = User::create([
30             'name' => $request->name,
31             'email' => $request->email,
32             'password' => Hash::make($request->password),
33         ]);
34
35         event(new Registered($user));
36
37         Auth::login($user);
38
39         return response()->noContent();
40     }
41 }
42
```

Fig.5. Controlador usado para el registro de nuevos usuarios



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'Asister_DB', including tables like 'dbo.failed_jobs', 'dbo.migrations', 'dbo.password_reset_tokens', 'dbo.personal_access_tokens', and 'dbo.users'. The 'dbo.users' table is selected, and its columns are listed: id (PK, bigint, not null), name (nvarchar(255), not null), email (nvarchar(255), not null), email_verified_at (datetime, null), password (nvarchar(255), not null), is_admin (bit, not null), remember_token (nvarchar(100), null), created_at (datetime, null), and updated_at (datetime, null). The main window shows a SQL query: 'select * from users'. The Results pane displays the following data:

id	name	email	email_verified_at	password	is_admin	remember_token	created_at
1	Guillermo Alonso	guillermo.alonso@ister.edu.ec	NULL	\$2y\$12\$sc/qO2.FjNlyxS7hmF1HRRefs.oTpiP83T1Iz1wDFjBKKa...	1	NULL	2024-07-04 02:0
2	ejemplo2	mail2@ejemplo.com	NULL	\$2y\$12\$poPD9ESb61Lvb9.sznVde.00WSb1YXK3OL1ZckiY9E...	0	NULL	2024-07-07 17:0
3	ejemplo3	mail3@ejemplo.com	NULL	\$2y\$12\$V04sr5LhMm7MxJdWgyEH.nzDHDa08DeHMJYKolk.e...	0	NULL	2024-07-09 00:2
4	ejemplo4	mail4@ejemplo.com	NULL	\$2y\$12\$5R3QyaTOd2yXp.XeGoJBe7Z6Pguj2MBZXq010z8jk0...	0	NULL	2024-07-09 00:5
5	ejemplo6	ejemplo6@mail.com	NULL	\$2y\$12\$CT4AstybtQ8zMIHFvHDYvW.qqlafny4p29of0WmtBlshk...	0	NULL	2024-07-22 22:0
6	Yeison Vanegas	yeison@gmail.com	NULL	\$2y\$12\$11QPNmqE8Vho015uEqIqOUGrG07TRPKj1iINy3CO...	0	NULL	2024-07-27 16:2
7	pijo	jeronimogilbert@gmail.com	NULL	\$2y\$12\$VC9o.V3BZ4W08jYURK2DVuL3s3dpLD7d7h7ID.ZFw...	0	NULL	2024-07-28 06:4

Fig.5. Base de datos en SQL SRV tabla de usuarios generados desde el ORM en Laravel.

Desarrollo de la Interfaz de Usuario Inicial con Vue.js:

- Se diseñó y desarrolló la interfaz de usuario para las páginas de registro, inicio de sesión y el chatbot.
- Se integró Vue Router para manejar la navegación entre las diferentes vistas.
- Se configuró el estado inicial de la aplicación utilizando Vuex para gestionar la autenticación de usuarios.



Evidencias:

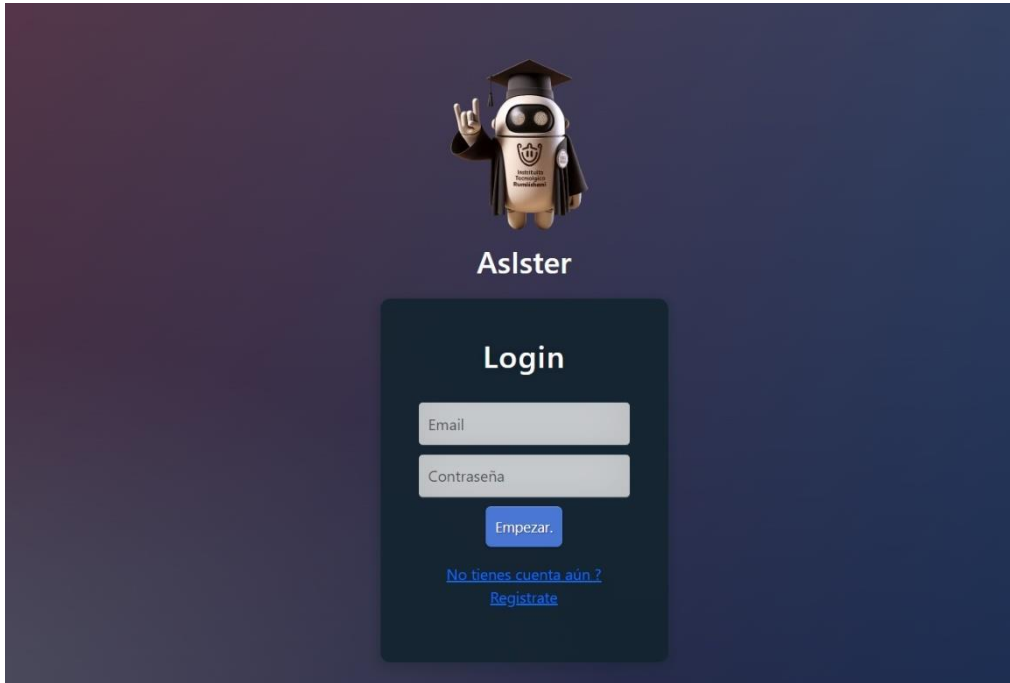


Fig.6. Interfaz de usuario de Login para la aplicación.

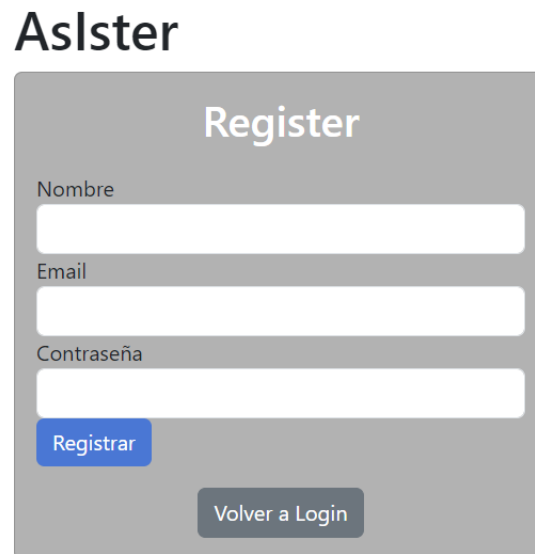


Fig.7. Pantalla de la interfaz de registro de usuarios.



```
1 // src/components/LoginUser.vue
2
3 <template>
4   <div class="login-container">
5     <div class="header">
6       
7       <h1>AsIster</h1>
8     </div>
9     <div class="login">
10      <h2>Login</h2>
11      <form @submit.prevent="login">
12        <div class="form-group">
13          <input
14            type="email"
15            v-model="email"
16            class="form-control"
17            placeholder="Email"
18            required
19            autocomplete="username"
20          />
21        </div>
22        <div class="form-group">
23          <input
24            type="password"
25            v-model="password"
26            class="form-control"
27            placeholder="Contraseña"
28            required
29            autocomplete="current-password"
30          />
31        </div>
32        <button type="submit" class="btn btn-primary btn-block btn-large">Empezar.</button>
33      </form>
34      <p class="text-center mt-3">
35        <a href="#" @click="goToRegister">No tienes cuenta aún ? Regístrate </a>
36      </p>
37    </div>
38  </div>
39 </template>
```

Fig.8. Código del componente LoginUser.vue.



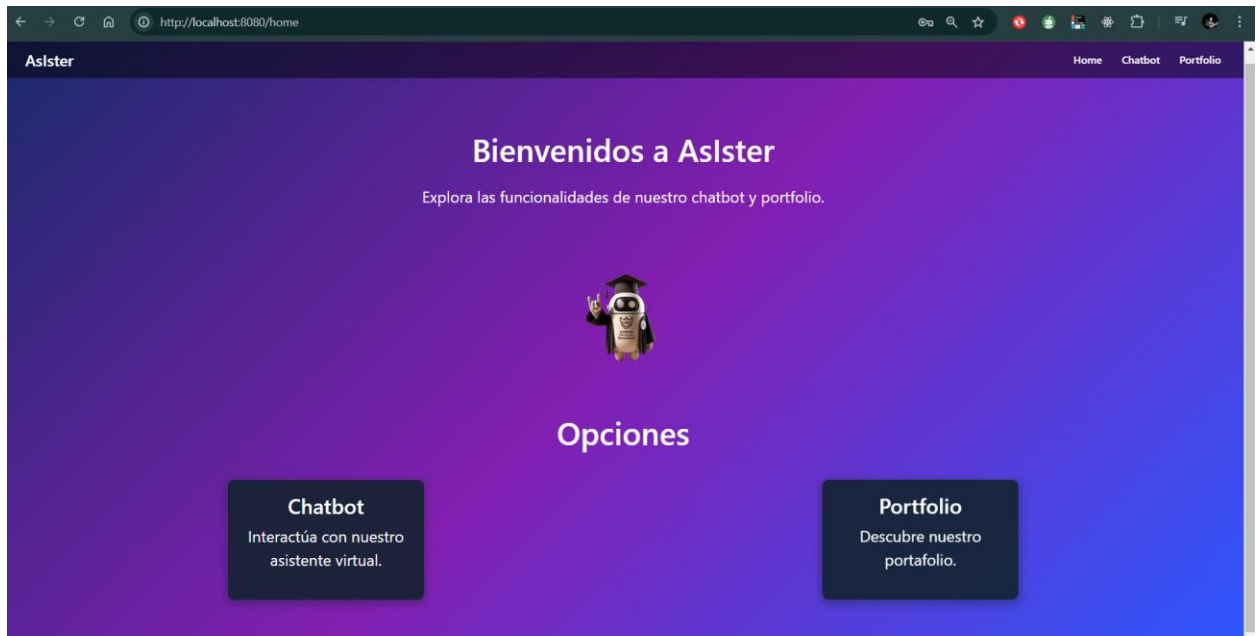


Fig.9. Captura de la pagina home en la aplicación.

Integración de la API open AI

- Se implementó una integración básica con la API de OpenAI utilizando Guzzle HTTP Client en Laravel.
- Se desarrolló un controlador que envía las consultas del usuario a la API de OpenAI y devuelve las respuestas al frontend.
- Se realizó una configuración básica para manejar las respuestas de la API y mostrarlas en la interfaz del chatbot.



Evidencias:

```
1 <?php
2 namespace App\Http\Controllers;
3
4 use Illuminate\Http\Request;
5 use GuzzleHttp\Client;
6 use Smalot\PdfParser\Parser;
7
8 class ChatbotController extends Controller
9 {
10     public function getResponse(Request $request)
11     {
12         $client = new Client();
13         $apiKey = env('OPENAI_API_KEY');
14         $assistantId = env('OPENAI_ASSISTANT_ID');
15         $pdfFilePath = storage_path('app/public/ister.pdf'); // Ruta del PDF con la información del ISTER
16
17         // Verificar que la clave API esté presente
18         if (!$apiKey) {
19             return response()->json(['error' => 'API key is missing'], 401);
20         }
21
22         // Extraer el texto del PDF
23         try {
24             $parser = new Parser();
25             $pdf = $parser->parsefile($pdfFilePath);
26             $text = $pdf->getText();
27         } catch (\Exception $e) {
28             return response()->json(['error' => 'Error parsing PDF: ' . $e->getMessage()], 500);
29         }
30
31         // Combinar el contenido del PDF con la pregunta del usuario
32         $combinedPrompt = $text . "\n" . $request->input('prompt');
33
34         try {
35             $response = $client->post('https://api.openai.com/v1/chat/completions', [
36                 'headers' => [
37                     'Authorization' => 'Bearer ' . $apiKey,
38                     'Content-Type' => 'application/json',
39                 ],
40                 'json' => [
41                     'model' => 'gpt-3.5-turbo',
42                     'messages' => [
43                         ['role' => 'user', 'content' => $combinedPrompt],
44                     ],
45                     'max_tokens' => 100, // Ajustar según sea necesario
46                 ],
47             ]);
48
49             return response()->json(json_decode($response->getBody()->getContents(), true));
50         } catch (\Exception $e) {
51             return response()->json(['error' => $e->getMessage()], 500);
52         }
53     }
54 }
```

Fig. 10. Integración Básica de la API de OpenAI a través del controlador





Fig.11. Captura de pantalla mostrando la respuesta de la API de OpenAI en la interfaz del chatbot.

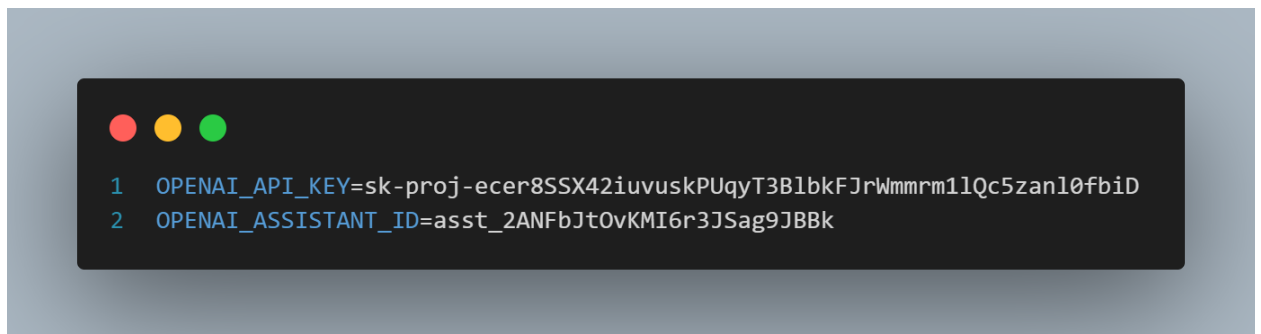


Fig.12. Captura de pantalla del archivo de configuración .env con la clave de la API.



4.1.5.2. Construcción de Iteración 2

Objetivos de la Iteración 2:

Implementación de funcionalidades adicionales en el backend.

Mejoras en la interfaz de usuario.

Revisión y ajuste de la documentación del código.

Casos de Uso Cubiertos:

CU-006: Gestión de Usuarios

CU-007: Mejoras en la Interfaz de Usuario

Desarrollo:

Gestión de Usuarios (CU-006):

Se implementaron las funcionalidades para permitir que el administrador pueda gestionar a los usuarios del sistema.

El administrador puede ver la lista de usuarios, crear nuevos usuarios, y editar o eliminar usuarios existentes.

Se aseguraron los permisos para que solo los administradores tengan acceso a estas funcionalidades.



Evidencias:

Usuarios Existentes

Nombre	Email	Rol	Acciones
Guillermo Alonso	guillermo.alonso@ister.edu.ec	Admin	Editar Eliminar
ejemplo2	mail2@ejemplo.com	Usuario	Editar Eliminar
ejemplo3	mail3@ejemplo.com	Usuario	Editar Eliminar
ejemplo4	mail4@ejemplo.com	Usuario	Editar Eliminar
ejemplo6	ejemplo6@mail.com	Usuario	Editar Eliminar
Yeison Vanegas	yeison@gmail.com	Usuario	Editar Eliminar
pijo	jeronimogilbert@gmail.com	Usuario	Editar Eliminar

Registrar Nuevo Usuario

Nombre

Email

Contraseña

Rol de Administrador

[Registrar](#)

Fig.13. Captura de pantalla del panel de gestión de usuarios.



Mejoras en la Interfaz de Usuario (CU-007):

Se realizaron mejoras visuales en la interfaz de usuario para hacerla más intuitiva y atractiva.

La navegación fue optimizada para facilitar la interacción de los usuarios con la aplicación.

Se aseguró la consistencia en la experiencia de usuario a través de todas las plataformas.

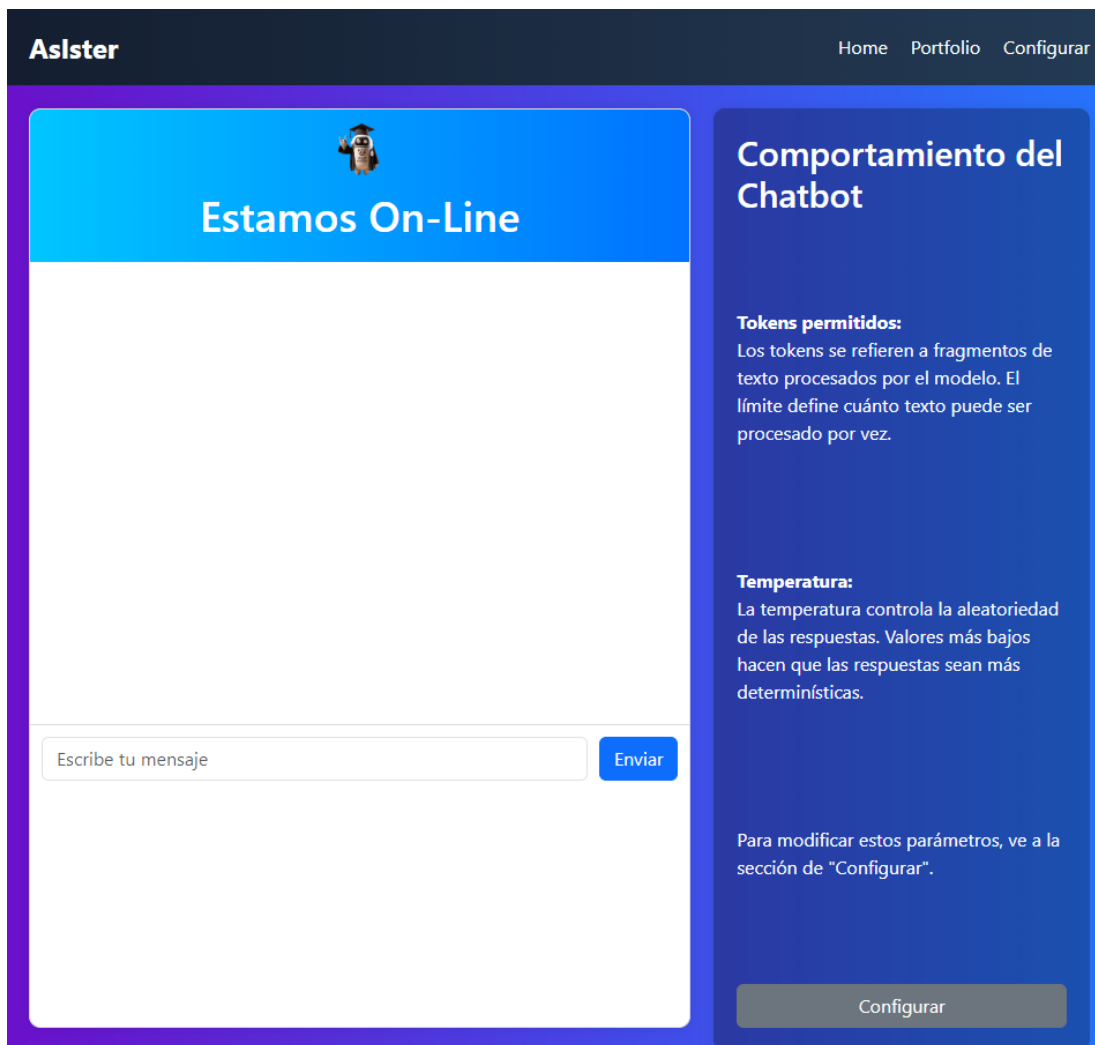


Fig.14. Captura de pantalla mostrando las mejoras en la interfaz de usuario.



4.1.5.3. Construcción de la Iteración 3

Objetivos de la Iteración 3:

Implementación del portafolio.

Mejora de la navegación entre rutas.

Casos de Uso Cubiertos:

CU-008: Implementación del Portafolio

CU-009: Mejora de la Navegación entre Rutas

Desarrollo

Implementación del Portafolio (CU-008):

Se desarrolló una nueva sección en la aplicación donde los usuarios pueden acceder al portafolio del proyecto.

El portafolio incluye información sobre la construcción del proyecto, el creador, y las tecnologías utilizadas, accesible desde la barra de navegación.

Evidencias:



Bienvenidos a Aslster

EXPLORA NUESTRO PROYECTO

Creación del Proyecto

El proyecto Aslster fue creado con el objetivo de proporcionar un asistente virtual capaz de interactuar con los usuarios y brindar información relevante de manera eficiente. Utilizando la API de OpenAI, se ha implementado un sistema de inteligencia artificial que permite respuestas precisas y contextualizadas. Este proyecto se desarrolló utilizando diversas tecnologías aprendidas en el Instituto Rumiñahui, aplicando conocimientos de desarrollo web y de inteligencia artificial para construir un sistema robusto y útil. La integración de Laravel en el backend y Vue.js en el frontend nos permitió crear una aplicación escalable y eficiente. Agradecemos al Instituto Rumiñahui por la formación brindada, la cual fue fundamental para la realización de este proyecto.



Fig.15. Captura de pantalla mostrando la vista del portafolio.



```
1  |-- About Section -->
2  <section id="about" class="bg-dark text-white">
3    <div class="container">
4      <div class="row text-center">
5        <div class="col-lg-12">
6          <h2 class="section-heading">Creación del Proyecto</h2>
7          <h3 class="section-subheading text-muted">Resumen de nuestro proyecto AsIster</h3>
8        </div>
9      </div>
10     <div class="row">
11       <div class="col-md-12">
12         <p class="lead">
13           El proyecto AsIster fue creado con el objetivo de proporcionar un asistente virtual capaz de
14           interactuar con los usuarios y brindar información relevante de manera eficiente. Utilizando
15           la API de OpenAI, se ha implementado un sistema de inteligencia artificial que permite respuestas
16           precisas y contextualizadas. Este proyecto se desarrolló utilizando diversas tecnologías aprendidas
17           en el Instituto Rumiñahui, aplicando conocimientos de desarrollo web y de inteligencia artificial para
18           construir un sistema robusto y útil. La integración de Laravel en el backend y Vue.js en el frontend nos
19           permitió crear una aplicación escalable y eficiente. Agradecemos al Instituto Rumiñahui por la formación brindada,
20           la cual fue fundamental para la realización de este proyecto.
21         </p>
22         
23       </div>
24     </div>
25   </div>
26 </section>
```

Fig.16. Código del componente *PortFolio.vue* implementando la lógica y el diseño del portafolio.

Mejora de la Navegación entre Rutas (CU-009):

Se optimizó la navegación entre rutas dentro de la aplicación, asegurando que las transiciones entre páginas sean rápidas y sin problemas.



Evidencias:

```
1 import { createRouter, createWebHistory } from 'vue-router';
2 import LoginUser from '../components/LoginUser.vue';
3 import RegisterUser from '../components/RegisterUser.vue';
4 import HomePage from '../components/HomePage.vue';
5 import AdminDashboard from '../components/AdminDashboard.vue';
6 import ChatBot from '../components/ChatBot.vue';
7 import PortFolio from '../components/PortFolio.vue';
8 import ManageUsers from '../components/ManageUsers.vue';
9 import EditUser from '../components/EditUser.vue'; // Asegúrate de importar este componente
10 import store from '../store';
11
12 const routes = [
13   { path: '/', component: LoginUser },
14   { path: '/register', component: RegisterUser },
15   { path: '/home', component: HomePage, meta: { requiresAuth: true } },
16   { path: '/admin', component: AdminDashboard, meta: { requiresAuth: true, requiresAdmin: true } },
17   { path: '/chatbot', component: ChatBot, meta: { requiresAuth: true } },
18   { path: '/portfolio', component: PortFolio, meta: { requiresAuth: true } },
19   { path: '/manage-users', component: ManageUsers, meta: { requiresAuth: true, requiresAdmin: true } },
20   { path: '/edit-user/:id', name: 'EditUser', component: EditUser, meta: { requiresAuth: true, requiresAdmin: true } },
21 ];
22
23 const router = createRouter({
24   history: createWebHistory(),
25   routes,
26 });
27
28 router.beforeEach((to, from, next) => {
29   const isAuthenticated = store.getters.isAuthenticated;
30   const isAdmin = store.getters.isAdmin;
31
32   if (to.matched.some(record => record.meta.requiresAuth)) {
33     if (!isAuthenticated) {
34       next('/'); // Redirige a login si no está autenticado
35     } else if (to.matched.some(record => record.meta.requiresAdmin) && !isAdmin) {
36       next('/home'); // Redirige a home si no es admin
37     } else {
38       next(); // Permite la navegación
39     }
40   } else {
41     next(); // Permite la navegación
42   }
43 });
44
45 export default router;
46
```

Fig.17. Código actualizado en el archivo router/index.js que define las rutas y manejadores de errores.



5. Conclusiones

A partir de la investigación, desarrollo y observación del proyecto AsIster, se concluye que:

1. **Aplicación Real de lo Aprendido:**

La investigación y desarrollo de AsIster permitió aplicar de manera efectiva los conocimientos adquiridos en inteligencia artificial y desarrollo web. Este proyecto demostró cómo estos conocimientos pueden ser transformados en soluciones innovadoras que mejoran los servicios, particularmente en el entorno académico.

2. **El Valor de la Inteligencia Artificial para Mejorar Servicios:**

La integración de IA en AsIster reveló el considerable potencial de los chatbots para optimizar la atención al cliente. Al proporcionar respuestas rápidas y precisas, no solo se mejora la experiencia del usuario, sino que también se incrementa la eficiencia operativa.

3. **Flexibilidad y Potencial de Crecimiento:**

Aunque AsIster no fue implementado en un entorno real, el proyecto demostró la adaptabilidad y escalabilidad de las tecnologías utilizadas. Con las adaptaciones adecuadas, estas soluciones tienen el potencial de ser aplicadas en diferentes sectores y contextos.

4. **La Importancia de la Actualización Continua:**

El desarrollo de AsIster subrayó la importancia de mantenerse actualizado con las nuevas tecnologías, especialmente en el campo de la inteligencia artificial. La investigación continua y la exploración de nuevas herramientas son esenciales para mejorar y adaptarse a un entorno tecnológico en constante evolución.

5. **Inspiración y Futuro Profesional:**

La creación de AsIster no solo fue un ejercicio técnico, sino también un reflejo del deseo



de explorar y aplicar tecnologías que puedan tener un impacto significativo. Este proyecto reafirmó la importancia de la investigación y el desarrollo tecnológico, impulsando futuras innovaciones en la carrera profesional.

6. Recomendaciones

En base a la experiencia adquirida con el proyecto AsIster, se recomienda lo siguiente:

1. Mantenerse Actualizado con las Tecnologías:

Es crucial mantenerse al día con las actualizaciones y cambios en las tecnologías utilizadas. Esto se logra mediante la revisión constante de la documentación oficial, lo cual permite adaptar cualquier sistema de manera más eficiente y asegurar su competitividad.

2. Iniciar con una Investigación Sólida:

Antes de iniciar el desarrollo, es fundamental comenzar con una investigación profunda. Esto implica comprender las tecnologías, probarlas en entornos controlados y construir sobre ese conocimiento, reduciendo errores y optimizando el proceso de desarrollo.

3. Diversificar las Fuentes de Información para los Bots:

Para mejorar la eficacia de los chatbots, se recomienda utilizar diferentes fuentes de absorción de información, como documentos en formato Markdown, bases de datos no relacionales y otros medios. Esto amplía las capacidades del bot y mejora su adaptabilidad a diferentes tipos de datos.



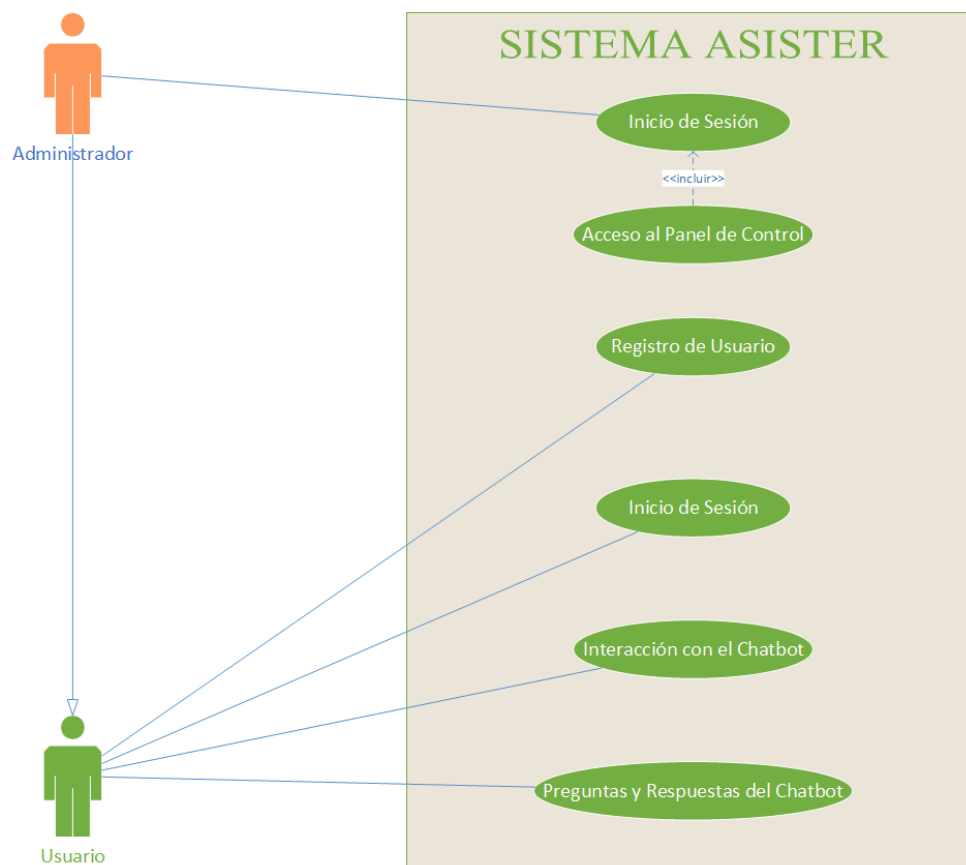
4. Experimentar con Bases de Datos No Relacionales:

Considerar el uso de bases de datos no relacionales es útil para manejar roles pre-programables en chatbots y otros sistemas. Estas bases de datos ofrecen una flexibilidad y escalabilidad que puede superar a las bases de datos tradicionales en ciertos escenarios.

5. Fomentar una Cultura de Aprendizaje Continuo:

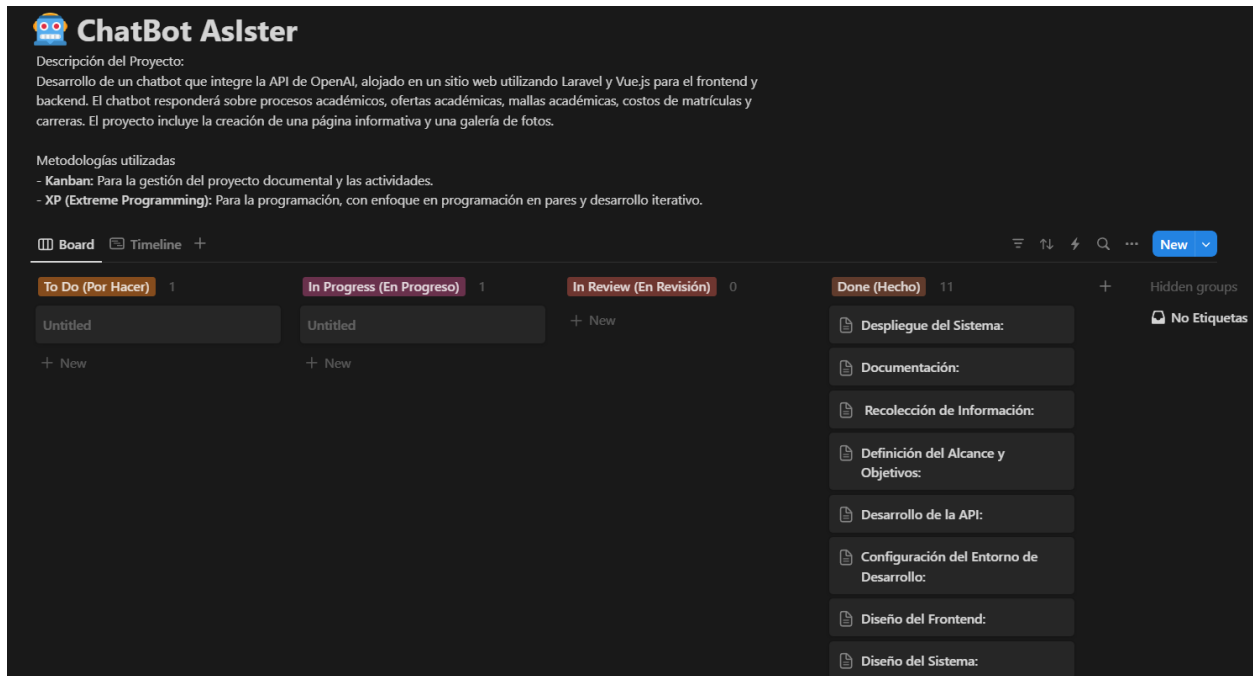
Para mantenerse relevante en el campo tecnológico, es esencial fomentar una cultura de aprendizaje continuo. Esto incluye estar al tanto de las últimas tendencias y experimentar con nuevas tecnologías y metodologías, asegurando así un avance constante.

7. Anexos

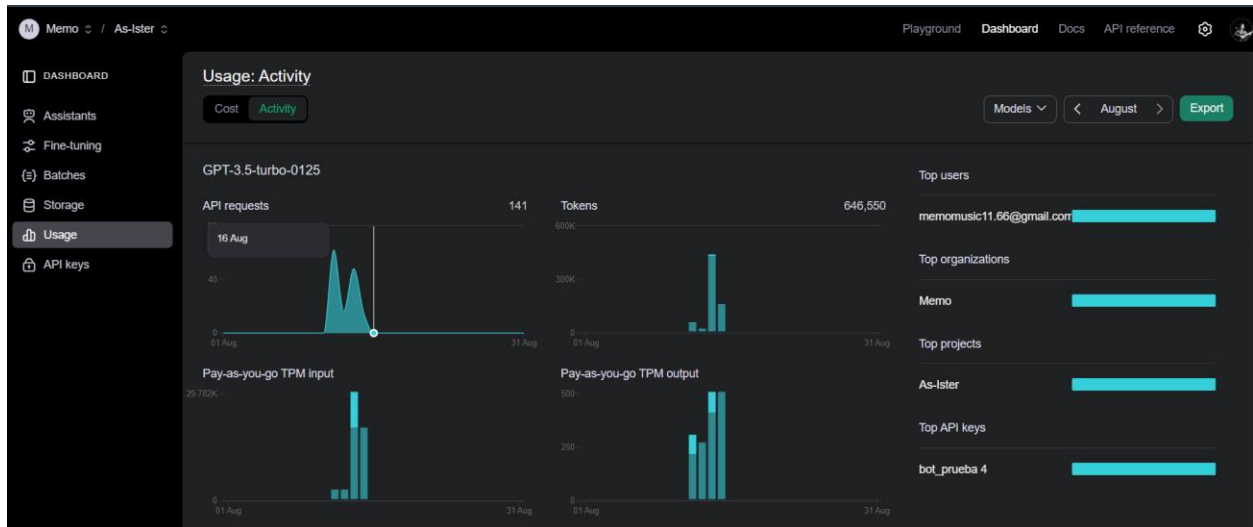


Anexo 1. Diagrama de caso de uso implementado a través de UML construido en Microsoft Visio.



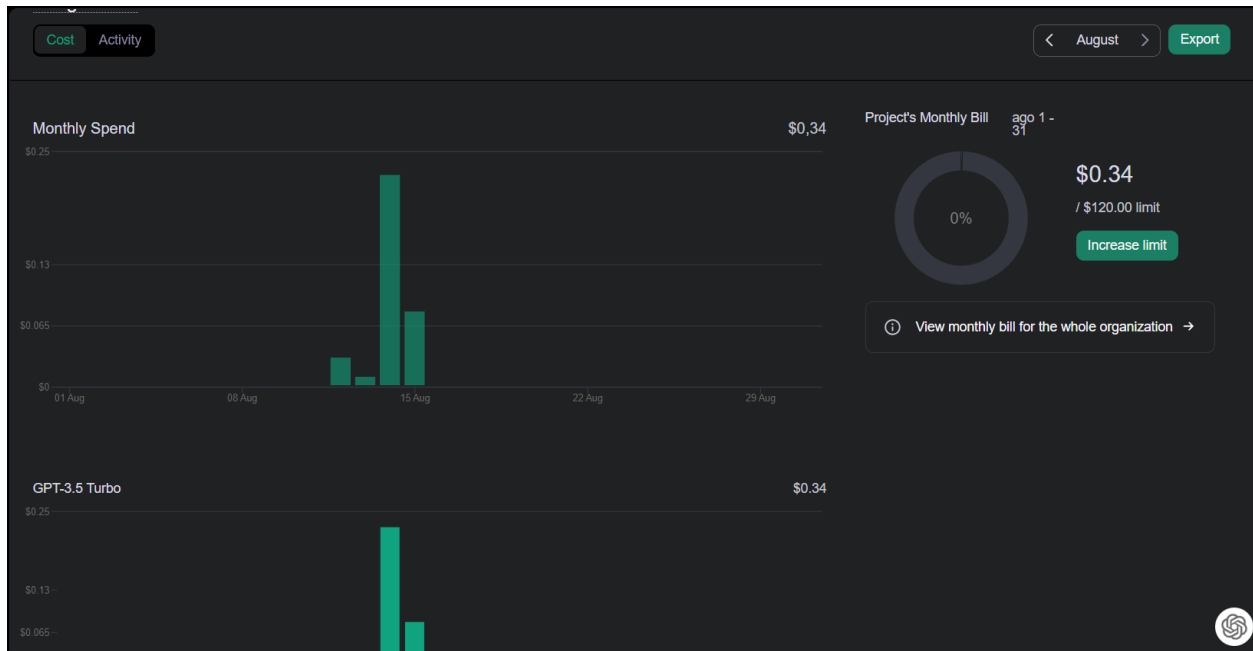


Anexo 2. Tablero Kanban utilizado en la metodología de desarrollo a través de la herramienta Notion.



Anexo 3. Panel de Actividad de consumo de la API de OpenAI durante las pruebas del Bot.





Anexo 4. Panel de consumo mensual regulado por el uso de tokens enviados y recibidos en cada solicitud a la API de OpenAi desde el controlador en Laravel.

API keys + Create new secret key

Project API keys have replaced user API keys.
We recommend using project based API keys for more granular control over your resources. [Learn more](#) View user API keys

As an owner of this project, you can view and manage all API keys in this project.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that has leaked publicly.

View usage per API key on the [Usage page](#).

NAME	SECRET KEY	CREATED	LAST USED	CREATED BY	PERMISSIONS
bot_prueba 4	sk-...fb1D	7 jul 2024	15 ago 2024	Memo Alonso	All
bot_prueba 3	sk-...qKMo	7 jul 2024	Never	Memo Alonso	All
Secret key	sk-...v0dE	26 jun 2024	Never	Memo Alonso	All
test_bot	sk-...4oII	26 jun 2024	Never	Memo Alonso	All

Anexo 5. Generación de cada una de las API Keys utilizadas durante el desarrollo y pruebas del bot (fechas en base a las iteraciones).



8. Bibliografía.

- Aquino, V. H. de O., & Adaniya, M. H. A. da C. (2018). Desenvolvimento e aplicações de Chatbot. *Revista Terra & Cultura: Cadernos de Ensino e Pesquisa*; v. 34 n. esp. (2018): *Ciência da Computação*. <http://periodicos.unifil.br/index.php/Revistateste/article/view/297>
- Bautista-Villegas, E. (2022). Metodologías ágiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel. *Revista Amazonía Digital*, 1(1), Article 1. <https://doi.org/10.55873/rad.v1i1.168>
- Church, B. (2023, septiembre 5). 5 types of chatbot and how to choose the right one. *IBM Blog*. <https://www.ibm.com/blog/chatbot-types>
- Código Orgánico de la Producción, Comercio e Inversiones. Registro Oficial, 587 Código Orgánico de la Producción, Comercio e Inversiones § Suplemento 587. (2010).
- Dharmalingam, N. (2020, octubre 13). *Top Applications of Chatbots (with Use Cases)*—*Whizlabs Blog*. <https://www.whizlabs.com/blog/applications-of-chatbots/>



Dilmegani, C. (2024, 01). *Top 37 Chatbot Applications & Use Cases in 2024.*

AIMultiple: High Tech Use Cases & Tools to Grow Your Business.

<https://research.aimultiple.com/chatbot-applications/>

Frain, B. (2022). *Responsive Web Design with HTML5 and CSS: Build future-proof responsive websites using the latest HTML5 and CSS techniques.* Packt Publishing Ltd.

Franganillo, J. (2022). Contenido generado por inteligencia artificial: Oportunidades y amenazas. *Anuario ThinkEPI*, 16. <https://doi.org/10.3145/thinkepi.2022.e16a24>

Garibay Ornelas, F. A. (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.* <http://infotec.repositorioinstitucional.mx/jspui/handle/1027/402>

Garrido Ortega, M. (Director). (2018, noviembre). *Chatbots en educación* [Video recording]. UNIA. Área de Innovación / Servicio Audiovisual. <https://dspace.unia.es/handle/10334/4111>

Ley de Comercio Electrónico, Firmas y Mensajes de Datos., 557. § Suplemento 557. (2002).

Ley Orgánica de Protección de Datos Personales., 459. § Suplemento 459. (2021).



Lu, H., Li, Y., Chen, M., Kim, H., & Serikawa, S. (2018). Brain Intelligence: Go beyond Artificial Intelligence. *Mobile Networks and Applications*, 23(2), 368-375. <https://doi.org/10.1007/s11036-017-0932-8>

Paz, F. J., Silveira, C., Krassmann, A., & Tarouco, L. M. R. (2017). Perspectivas tecnológicas para o aprimoramento de chatbots educacionais em AIML. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, 20, 7-15.

¿Qué es una API y cómo funciona? (s. f.). Recuperado 26 de marzo de 2024, de <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

¿Qué es una interfaz de programación de aplicaciones (API)? | IBM. (s. f.). Recuperado 26 de marzo de 2024, de <https://www.ibm.com/mx-es/topics/api>

¿Qué son los bots de Inteligencia Artificial y cómo ayudan a las personas? - Planeta Chatbot. (s. f.). Recuperado 5 de febrero de 2024, de <https://planetachatbot.com/que-son-bots-de-inteligencia-artificial-y-como-ayudan-a-personas/>

Rahaman, S. (2023, octubre 20). *The Different Types of Chatbots: AI, NPL, Rule-Based & More.* <https://www.fullview.io/blog/types-of-chatbots>, <https://www.fullview.io/blog/types-of-chatbots>



- Redacción. (2023, junio 3). Introducción a los asistentes virtuales y chatbots. *Inteligencia Artificial*. <https://inteligenciaartificial.science/introduccion-a-los-asistentes-virtuales-y-chatbots/>
- Rivas Villatoro, L. M. (2021). *Propuesta de mejora en la gestion de la atencion al cliente por medio de un asistente virtual tipo Chatbot para fortalecer el desempeño del servicio del proyecto de educacion continua* .pdf. <https://www.kerwa.ucr.ac.cr/bitstream/handle/10669/84442/Propuesta>
- Rubio, J. M., Neira-Peña, T., Molina, D., Vidal-Silva, C., Rubio, J. M., Neira-Peña, T., Molina, D., & Vidal-Silva, C. (2022). Proyecto UBOT: Asistente virtual para entornos virtuales de aprendizaje. *Información tecnológica*, 33(4), 85-92. <https://doi.org/10.4067/S0718-07642022000400085>
- Salazar, J. B., & Campos, P. G. (s. f.). *Modelo Para Seguridad de la Información en TIC*.
- Surwase, V. (s. f.). *REST API Modeling Languages—A Developer’s Perspective*. 2(10).
- Valarezo Pardo, M. R., Honores Tapia, J. A., Gómez Moreno, A. S., & Vines Sánchez, L. F. (2018). Comparación de tendencias tecnológicas en aplicaciones web. *3c Tecnología: glosas de innovación aplicadas a la pyme*, 7(3), 28-49.



visual, P. el equipo de periodismo. (s. f.). *¿Qué es la IA? Una guía sencilla para entender la inteligencia artificial - BBC News Mundo*. News Mundo. Recuperado 5 de febrero de 2024, de <https://www.bbc.com/mundo/resources/identity-74697280-e684-43c5-a782-29e9d11fecf3>

