Sistema de chatbot con IA para brindar respuestas a servicio de soporte técnico y usuarios del proyecto Itel de Oracle

Arnold Sebastián Beltrán Puentes

Asesora

Dra. Mónica Andrea Rico Martínez

Universidad Nacional Abierta y a Distancia UNAD

Ingeniería Telecomunicaciones

Escuela de Ciencias Básicas, Tecnología e Ingeniería

2025

Nombre Director de Trabajo de Grado

Jurado

Jurado

2025

Agradecimientos

En primer lugar, doy gracias a Dios y a la Universidad Nacional Abierta y a Distancia UNAD que me acompañaron en este proceso de formación.

Gracias enormes a la Ingeniera y Licenciada, Doctora Andrea Rico Martínez que con mucho cariño y disposición brindó el apoyo incondicional mediante este acompañamiento como asesora de este proyecto aplicado.

Gracias al ing. Diego Amador, Líder de Operaciones de Oracle y a la Ingeniera Derlis Mendoza de la empresa DEMIL400 por brindar la confianza necesaria para hacer realidad este proyecto aplicado.

Gracias a los recursos obtenidos por la página de la universidad como fuentes de internet y conocimiento experto en donde ofrece una amplia gama de información y conocimiento por explorar.

Gracias a las empresas que hicieron partícipe en este acompañamiento aplicando las tecnologías propuestas en este proyecto.

Gracias a todos en general que de alguna u otra manera aportaron a este proyecto:

A nuestros familiares

A nuestros amigos

A nuestros tutores

Resumen

A la fecha, las empresas buscan optimizar sus procesos y mejorar la eficiencia operativa, especialmente en el soporte técnico de primer nivel. La alta demanda en mesas de servicio genera sobrecarga en los equipos de trabajo, afectando la calidad del servicio y la resolución de incidentes. En este contexto, los chatbots con inteligencia artificial (IA) representan una alternativa viable para automatizar respuestas, reducir tiempos de espera y optimizar la gestión de requerimientos.

Este trabajo de grado desarrolla un chatbot con IA para el soporte técnico en el proyecto Itel de Oracle de Claro Soluciones. El chatbot fue diseñado aplicando procesamiento de lenguaje natural (PLN), aprendizaje automático y cumpliendo metodologías ágiles como Scrum. Su implementación permite responder preguntas frecuentes, gestionar requerimientos de soporte, monitorear servicios de Itel, capacitar y reforzar saberes a los técnicos y enviar notificaciones de estado de servicios y monitoreo a través de WhatsApp, entre otros servicios.

Este desarrollo se realizó de acuerdo con los objetivos propuestos: levantamiento de los requerimientos e información necesaria, diseño de arquitectura, análisis de lenguajes de programación, implementación del modelo IA y pruebas funcionales del chatbot. Se emplearon herramientas como Python y módulos de tokenización como TensorFlow, además de conectarse con las bases de datos Oracle para garantizar precisión en las respuestas entre otras librerías para su correcto funcionamiento.

Los resultados evidencian el cumplimiento de los objetivos del proyecto. El chatbot automatizó el 80% de las consultas frecuentes, reduciendo en un 60% los tiempos de respuesta. Además, mejoró la gestión de requerimientos mediante la creación y consulta de casos,

facilitando la resolución de tareas repetitivas. En pruebas de desempeño, demostró estabilidad en escenarios de alta demanda sin comprometer con la calidad del servicio.

La implementación de esta herramienta contribuye a la eficiencia operativa del soporte técnico, optimizando el uso de recursos humanos y mejorando la experiencia del usuario. Su integración con bases de datos y sistemas de monitoreo, incluyendo el envío de notificaciones por WhatsApp, garantiza su adaptabilidad a nuevas necesidades y generaciones del negocio.

Palabras clave: Chatbot, inteligencia artificial, soporte técnico, mesa de servicios, procesamiento de lenguaje natural, Scrum, telecomunicaciones.

Abstract

Currently, companies seek to optimize their processes and improve operational efficiency, particularly in first-level technical support. The increasing demand in service desks creates operational overload, affecting service quality and incident resolution. In this context, artificial intelligence (AI) chatbots offer a viable solution for automating responses, reducing wait times, and improving request management.

This research project focuses on developing an AI-powered chatbot for technical support in Claro Soluciones' Itel Oracle project. The chatbot was designed using natural language processing (NLP), machine learning, and agile methodologies such as Scrum. Its implementation enables it to respond to frequently asked questions, manage support requests, monitor Itel services, train technical staff, and send service status notifications via WhatsApp.

The development followed key objectives: requirements gathering, system architecture design, programming language evaluation, AI model implementation, and functional testing. The chatbot was built using Python, tokenization techniques with TensorFlow, and integrated Oracle databases to ensure response accuracy.

The results confirm the project's success. The chatbot automated 80% of frequent queries, reducing response times by 60%. Additionally, it improved request management by facilitating case creation and tracking, streamlining the resolution of repetitive tasks. Performance testing demonstrated system stability under high-demand conditions without compromising service quality.

The integration of this tool enhances operational efficiency in technical support, optimizing human resource allocation and improving user experience. Its ability to connect with

databases and monitoring systems, including WhatsApp notifications, ensures adaptability to future business needs.

Keywords: Chatbot, artificial intelligence, technical support, service desk, natural language processing, Scrum, telecommunications.

Tabla de contenido

Índice de figuras 20

Índice de tablas 19

Introducción 22

Generalidades de la investigación 26

Planteamiento del problema 26

Justificación 29

Objetivos 31

Objetivo General 31

Objetivo Específico 32

Marco de Referencia 32

Marco teórico 32

Estado del arte 41

A nivel internacional 41

A nivel nacional 42

A nivel local 44

Metodología 45

Fases del proyecto 45

Fase 1: Levantamiento de información de requerimientos 45

Actividades programadas 45

Fase 2: Diseño del software 45

Actividades programadas 45

Fase 3: Análisis de lenguajes de programación 46

Actividades programadas 46

Fase 4: Desarrollo del prototipo 46

Actividades programadas 46

Fase 5: Pruebas y validación 47

Actividades programadas 47

Levantamiento de los requerimientos del sistema de chatbot 48

Información del proyecto 48

Propósito 49

Definición del proyecto 49

Alcance del proyecto 49

Módulo de procesamiento de lenguaje natural (PLN) 49

Módulo de generación y gestión de requerimientos (REQ) 50

Módulo de integración con sistemas externos 50

Módulo de análisis y reportes programados 50

Módulo de entrenamiento y mejora continua 50

Módulo de interfaz de usuario (UI) 51

Módulo de documentación y ayuda 51

Módulos de diagnóstico de telecomunicaciones 51

Beneficios del proyecto 51

Descripción del entorno operativo 52

Funcionalidades principales 54
Respuesta a preguntas frecuentes 54
Generación y gestión de requerimientos de soporte 54
Capacitación y asistencia a técnicos 54
Monitoreo y diagnóstico de red 55
Seguridad y privacidad de datos 55
Interfaz de usuario 55
Requerimientos funcionales, no funcionales y otros requerimientos 55
Requerimientos funcionales 56
Procesamiento de lenguaje natural (PLN) 56
Requerimientos aplicados 56
Autenticación de usuarios 57
Requerimientos aplicados 57
Respuesta a preguntas frecuentes 57
Requerimientos aplicados 57
Generación y seguimiento de requerimientos 57
Requerimientos aplicados 58
Capacitación y soporte a técnicos 58
Requerimientos aplicados 58
Laboratorio de pruebas de telecomunicaciones 58
Requerimientos aplicados 59
Sistema de gestión y monitoreo 59

Requerimientos aplicados 59
Requerimientos no funcionales 60
Rendimiento 60
Requerimientos aplicados 60
Escalabilidad 61
Requerimientos aplicados 61
Seguridad 61
Requerimientos aplicados 61
Usabilidad 61
Requerimientos aplicados 62
Mantenibilidad 62
Requerimientos aplicados 62
Fiabilidad 62
Requerimientos aplicados 62
Compatibilidad 63
Requerimientos aplicados 63
Otros Requerimientos 63
Mejora de la interfaz - modo oscuro 63
Requerimientos aplicados 64
Interconexión con GPT 64
Requerimientos aplicados 64

Alertas y notificaciones 64
Requerimientos aplicados 64
Reglas de negocio 65
Privacidad y protección de datos 65
Uso adecuado 65
Registro y monitoreo de actividades 65
Actualización y mantenimiento 66
Capacitación y soporte 66
Cumplimiento legal 66
Impacto empresarial 66
Mejora de la eficiencia operativa 67
Optimización del talento humano 67
Incremento de la satisfacción del cliente 68
Escalabilidad y adaptabilidad 68
Cumplimiento con normativas y mejora de la seguridad 69
Resultados esperados 69
Implementación de metodología Scrum 71
Diseño y aplicación de metodologías agiles en el proyecto 71
Metodología Scrum aplicada al chatbot de Itel 75
Descripción de Scrum 75

Componentes clave de Scrum 75
Roles definidos 75
Scrum Master 75
Scrum Product Owner 77
Scrum Developer 78
Diseño, comparación y pruebas de desempeño para seleccionar las herramientas a implementar 81
Criterios de evaluación de lenguajes de programación 81
Rendimiento 82
Algunos puntos para destacar 82
Consideraciones 82
Escalabilidad 82
Algunos puntos para destacar 83
Consideraciones 83
Facilidad de desarrollo 83
Algunos puntos para destacar 83
Consideraciones 84
Compatibilidad e integración 84
Algunos puntos para destacar 84
Consideraciones 85
Seguridad 85
Algunos puntos para destacar 85

Consideraciones 85

Mantenimiento y soporte 86

Algunos puntos para destacar 86

Consideraciones 86

Comparación entre lenguajes de programación para chatbots 86

Desarrollo de la aplicación con IA – sistema de chatbot 91

Arquitectura del sistema 91

Backend 91

Frontend 92

Integración con IA/PLN 92

Bases de datos 92

Herramientas y tecnologías utilizadas 93

Flujo de interacción del usuario 95

Descripción del flujo de interacción 95

Procesos matemáticos 95

Descripción del flujograma matemáticas 96

Consulta de preguntas frecuentes 97

Descripción del flujograma preguntas frecuentes 97

Selección de opciones y sugerencias 98

Descripción del flujograma opciones y sugerencias 99

Creación de REQ. en caso de una falla 100

Descripción del flujograma para la creación de REQ 101

Consultas de REQ 102
Descripción del flujograma para las consultas de REQ 103
Actualización y entrenamiento del modelo chatbot 105
Descripción del flujograma para el entrenamiento del modelo 105
Selección modelo avanzado 107
Descripción del flujograma en cuanto a la selección del modelo avanzado 107
Monitoreo sistemas de telecomunicaciones 108
Descripción del flujograma del flujo de monitoreo de sistemas de telecomunicaciones. 109
Implementación y despliegue 110
Proceso de implementación en el equipo cliente 110
Preparación del entorno de desarrollo y pruebas 110
Configuración del equipo servidor de producción 111
Configuración del entorno de producción 111
Mantenimiento del sistema y actualizaciones futuras 112
Chatbot puesta en producción 113
Enlaces repositorio del código y video 113
Algunas pruebas iniciales 114
Evidencia del reCAPTCHA de Google 114
Presentación inicial de la interfaz 116
Interacciones básicas simples 116
Interacciones matemáticas 117

Selección de modelo del chatbot para tareas más complejas y solución de errores de la mesa de servicios 120

Lectura en voz alta, me gusta, no me gusta y copiar respuesta. 122

Cambio de interfaz a modo claro / modo oscuro 123

Reproducción multimedia 124

Acceso a archivos digitales 125

Pruebas funcionales al chatbot 127

Pruebas de desempeño 128

Resultado pruebas de entrenamiento 128

Resultados pruebas de rendimiento 132

Análisis acerca de los resultados 133

Resultado de las pruebas del modelo chatbot para Itel Oracle 135

Propiedades de los equipos para el laboratorio del chatbot 135

Propiedades del equipo 1 135

Propiedades del equipo 2 136

Propiedades del equipo 3 137

Requerimientos mínimos del modelo chatbot 137

Comparativa de rendimiento en equipos locales y sede cliente 139

Muestra de rendimiento del sistema y consumo de chatbot en producción 140

Pruebas de rendimiento del sistema con htop 140

Pruebas de rendimiento del chatbot con htop 141

Pruebas de rendimiento de hardare y software y recursos consumidos del chatbot con bashtop 143

Resumen final del estado de CPU 144

Recomendaciones y mejoras futuras 146

Ampliación del entrenamiento del sistema 146

Optimización de la interfaz de usuario 146

Ampliación de la capacidad de hardware 147

Escalabilidad del sistema 147

Migración a la infraestructura OCI de Oracle 148

Promoción y uso del chatbot en nuevos proyectos 148

Compatibilidad para carga de archivos 148

Gráficas de monitoreo de servicios 149

Conclusiones 150

Referencias bibliográficas 153

Índice de tablas

Tabla 1 *Información del proyecto* 48

Tabla 2 *Hardware y software requerido y opcionales* 52

Tabla 3 *Comparativa entre metodologías ágiles vs tradicionales* 73

Tabla 4 *Razones para usar metodologías ágiles en el diseño del chatbot Itel* 74

Tabla 5 *Responsabilidades y limitaciones Scrum Master* 76

Tabla 6 *Responsabilidades y limitaciones Scrum Product Owner* 77

Tabla 7 *Responsabilidades y limitaciones Scrum Developer* 79

Tabla 8 *Resumen comparativa lenguajes de desarrollo para chatbots* 86

Tabla 9 *Herramientas utilizadas para el desarrollo del chatbot* 93

Tabla 10 *Enlace código del desarrollo y video tutorial funcionamiento* 113

Tabla 11 *Evaluación del nivel de complejidad en chatbot* 133

Tabla 12 *Resultados en pruebas de desempeño* 134

Tabla 13 *Información características equipo 1* 135

Tabla 14 *Información características equipo 2* 136

Tabla 15 *Información características equipo sede cliente* 137

Tabla 16 *Requerimientos mínimos del chatbot en cuanto a hardware - software* 137

Tabla 17 *Informe de monitoreo con htop con más procesos activos* 141

Tabla 18 *Informe del consumo de PC con chatbot* 142

Tabla 19 *Informe consumo general de proceso en equipo de producción* 144

Índice de figuras

Figura 1 *Redes neuronales recurrentes (RNN)* 35

Figura 2 *Metodologías de desarrollo de software tradicionales vs ágiles* 72

Figura 3 *Insignia Scrum Master* 75

Figura 4 *Insignia Scrum Product Owner* 77

Figura 5 *Insignia Scrum Developer* 79

Figura 6 *Gráfica comparativa entre lenguajes de programación para chatbots* 90

Figura 7 *Diagrama de flujo proceso matemático* 96

Figura 8 *Diagrama de flujo consultas frecuentes* 97

Figura 9 *Diagrama de flujo de opciones y sugerencias* 99

Figura 10 *Diagrama de flujo creación de REQ* 101

Figura 11 *Diagrama de flujo consulta de REQ* 103

Figura 12 *Diagrama de flujo actualización y entrenamiento del modelo* 105

Figura 13 *Diagrama de flujo para la selección del modelo avanzado o IA* 107

Figura 14 *Diagrama de flujo monitoreo de servicios de telecomunicaciones* 109

Figura 15 *Presentación de bienvenida del chatbot con reCAPTCHA* 114

Figura 16 *Prueba reCAPTCHA* 115

Figura 17 *Presentación de bienvenida o interfaz del chatbot* 116

Figura 18 *Interacciones básicas simples* 117

Figura 19 *Operaciones matemáticas básicas en chatbot* 118

Figura 20 *Comprobación de cálculo matemático en Mathway* 119

Figura 21 *Operaciones algebraicas x + y* 120

Figura 22 *Cambio de modelo básico a modelo avanzado con IA* 121

Figura 23 *Respuesta generada por modelo avanzado IA* 122

Figura 24 *Íconos opción "Me gusta", "No me gusta", leer respuesta en voz alta* 123

Figura 25 *Cambio modo claro y oscuro* 124

Figura 26 *Reproducción multimedia en chatbot* 124

Figura 27 *Reproducción de videos YouTube* 125

Figura 28 *Visualización de archivos digitales como manuales y tutoriales* 126

Figura 29 *Evolución de la pérdida y precisión durante el entrenamiento etapa 1* 129

Figura 30 *Evolución de la pérdida y precisión durante el entrenamiento etapa 2* 130

Figura 31 *Evaluación de entrenamiento mediante épocas en chatbot* 131

Figura 32 *Muestra entre entrenamiento en cuanto a precisión y pérdida* 132

Figura 33 *Gráfica comparativa entre hardware de los equipos y requisitos del chatbot* 139

Figura 34 *Htop con procesos activos del sistema* 140

Figura 35 *Htop con múltiples procesos del chatbot* 142

Figura 36 *Bashtop para análisis gráfico general* 143

Introducción

El creciente avance del sector tecnológico y la necesidad de optimizar la atención al cliente por alta demanda de usuarios han impulsado la adopción de soluciones automatizadas de vanguardia, uno de ellos son los chatbots inteligentes, con interacción programada o reconocimiento de patrones. Este trabajo de proyecto de grado se enfoca en el desarrollo de un sistema de asistente virtual, diseñado para proporcionar soluciones de soporte técnico de nivel 1 y automatizar la gestión de consultas técnicas en la empresa Claro Soluciones, dentro del contexto del proyecto Itel de Oracle.

El desarrollo de este chatbot enfrentó numerosos retos a lo largo del proceso, incluyendo múltiples iteraciones de prueba y error para adaptar las herramientas existentes del negocio y garantizar la compatibilidad entre los diversos módulos. Estos desafíos permitieron identificar oportunidades de mejora y fortalecer el modelo mediante un enfoque incremental. El chatbot fue desarrollado utilizando el lenguaje de programación Python, aplicando técnicas avanzadas de procesamiento de lenguaje natural (PLN), como la tokenización y selección de palabras clave, para comprender y responder preguntas frecuentes de manera precisa y eficiente. Además, se implementó el desarrollo de la aplicación mediante el marco de trabajo Scrum, lo cual facilitó una gestión estructurada y dinámica del desarrollo del software.

El objetivo principal propuesto, además de desarrollar un prototipo funcional de chatbot, fue integrar inteligencia artificial para proporcionar respuestas de acuerdo con las exigencias del negocio, procesar solicitudes dirigidas y dar seguimiento a requerimientos en soporte técnico y mesa de servicios. Este sistema no solo optimiza la eficiencia del soporte técnico, sino que también libera recursos humanos para tareas más complejas y especializadas, aumentando así el valor agregado de las operaciones.

En cuanto a este documento, inicia con el planteamiento del problema, que identifica la necesidad de automatizar la gestión de soporte técnico debido al incremento en la demanda de servicios en entornos remotos. La justificación destaca la necesidad y el impacto positivo de los chatbots en la eficiencia operativa y la atención al cliente, respaldado por datos de algunos sectores. Además, se establecen objetivos generales y específicos, como el diseño de un prototipo de chatbot utilizando metodologías ágiles como Scrum, y la inclusión de técnicas avanzadas de procesamiento de lenguaje natural (PLN) y aprendizaje automático.

Más adelante, se describen los módulos funcionales del chatbot, incluyendo el procesamiento de lenguaje natural, la generación y gestión de requerimientos (REQ), el monitoreo de servidores, la capacitación para técnicos y apoyos en cuanto a pruebas de telecomunicaciones. También se analizan los beneficios esperados, como la reducción de carga laboral, la mejora en la satisfacción del cliente y el cumplimiento de normativas de seguridad y privacidad, entre otros. Además, se presentan las fases del proyecto, desde la definición de requerimientos hasta la creación de prototipos. La arquitectura del sistema está compuesta por un backend que integra algoritmos de IA y PLN, y un frontend desarrollado con Flask para HTML que también va incluída en Python para proporcionar una interfaz más amigable. Este enfoque permite una entrega incremental y la adaptación continua a las necesidades del proyecto.

Luego, se realiza un análisis comparativo de lenguajes de programación para evaluar su idoneidad en el desarrollo del chatbot. Python se destaca como la opción ideal por su flexibilidad, soporte para bibliotecas de IA y facilidad de integración. También se describen pruebas de desempeño que validan la eficiencia del sistema en escenarios de alta demanda y su capacidad para manejar interacciones simultáneas sin comprometer la calidad del servicio.

El desarrollo del chatbot incluye la implementación de técnicas de aprendizaje automático para entrenar al sistema con bases de datos personalizadas, garantizando respuestas precisas y relevantes. Se integran módulos de diagnóstico de telecomunicaciones y generación de reportes automatizados, que contribuyen a la mejora continua del sistema y reporte al equipo de altos niveles en caso de no tener alcance común. Adicionalmente, se detallan los algoritmos y modelos utilizados, como redes neuronales recurrentes (RNN) y módulos como TensorFlow y NLTK, para optimizar el procesamiento del lenguaje natural.

Por último, se llevaron a cabo suficientes pruebas para evaluar la funcionalidad y el desempeño del chatbot. Estas pruebas incluyeron casos de uso reales y escenarios simulados en el entorno local ubicado en Claro Soluciones en la sede Centro Comercial de Plaza Claro, permitiendo identificar áreas de mejora y garantizar la calidad del producto final. Los resultados evidencian una mejora significativa en la gestión de soporte técnico, cumpliendo con los objetivos establecidos.

A pesar de ser una herramienta poderosa y útil para el negocio, es importante aclarar en esta tesis con su modelo de chatbot básica no busca competir con otras soluciones existentes, como OpenAI, Meta, Microsoft, DeepSeek, entre otros, que utilizan modelos de lenguaje avanzada u otras tecnologías de código abierto. Esta herramienta está diseñada exclusivamente para el cliente interno de Oracle y Claro Soluciones. Además, este documento puede ser de utilidad como referencia en futuras investigaciones aplicadas a chatbots y automatizaciones de soporte técnico a nivel local.

Asimismo, se reconoce que este chatbot al ser su primera versión, su precisión de respuestas sea alrededor de un 80%, el cual, es bastante considerable, también es susceptible a errores, lo que puede generar respuestas incorrectas o no alineadas con las expectativas del

negocio. Una posible solución es que requiere una mayor cantidad de información y entrenamientos constantes en su modelo de base de datos básico para mejorar la precisión de las respuestas esperadas.

Generalidades de la investigación

En el presente capítulo se expone las generalidades de la investigación, dando un contexto a través del planteamiento del problema, la justificación, el estado del arte y marco de referencia, para finalmente culminar con la exposición de los objetivos con su respectiva metodología.

Planteamiento del problema

IBM citó lo siguiente en su página web oficial: Un chatbot es una aplicación software que simula la conversación humana con un usuario final. No todos los chatbots están equipados con inteligencia artificial (IA) ("Inteligencia artificial," 2024) pero los chatbots modernos utilizan cada vez más técnicas de IA conversacional como el procesamiento del lenguaje natural (PLN) (*¿Qué es el procesamiento del lenguaje natural (PLN)?*, 2021) para comprender las preguntas de los usuarios y automatización de respuestas. Lo anterior indica permitir automatizar procesos repetitivos rutinarios dentro de las organizaciones. Para comprender mejor qué es un chatbot, se recomienda consultar en siguiente cita dentro del marco de referencia: (*¿Qué es un chatbot?*, n.d.).

Durante la pandemia que se vivió en el año 2020 en Colombia, mientras que el actor de esta tesis formaba parte del Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC) (*MINTIC Colombia - Inicio*, n.d.) en el cargo de técnico de soporte la mesa de servicios, en su momento y por decisiones directivas y gubernamentales se implementó el trabajo en casa o "teletrabajo", luego, mediante este cambio irreversible se presentaron nuevas ideas y oportunidades; a lo largo de ello, se evidenció la necesidad de una solución viable para la gestión de soporte técnico debido al aumento en la demanda notable de atención al usuario por la transición a modalidades de trabajo remoto. Aunque en su momento se implementaron canales de

comunicación tecnológica de mensajería instantánea como Microsoft Teams para abordar esta necesidad en atención al usuario, la capacidad de respuesta se vio limitada por la saturación en las líneas telefónicas de “call center” y la carga de trabajo de los técnicos de la mesa de servicios de MinTIC desde casa en cuanto a la atención del chat en el canal de Teams.

En este contexto, se desarrolló de manera provisional un asistente virtual a su máxima versión 1.1.4. basado en un chatbot implementado con Python para Microsoft Teams (código que reposa en GitHub), con la finalidad de automatizar la resolución de consultas de preguntas frecuentes y la generación de requerimientos de forma rápida (Sebastianbel01, 2022). Para esta versión, este asistente demostró una capacidad significativamente mayor para manejar casos diarios repetitivos en comparación con el trabajo manual de los colegas de la mesa de servicios, demostrando una optimización de los procesos. Sin embargo, la operación del proyecto se vio interrumpida debido a conflictos sindicales por parte del equipo de trabajo en su momento, el cual se concluye la falta de preparación por parte del equipo técnico para este “boom” tecnológico que se avecinaría.

A pesar de este contratiempo, la experiencia del modelo piloto dejó en claro el potencial de un sistema inteligente para mejorar la eficiencia y la calidad del soporte técnico de la mesa de servicio de ese entonces, ya que, como se mencionó, se optimizó el proceso de respuesta oportuna a los usuarios de la mesa de servicios de MinTIC. Por otra parte, se observó una ingeniosa oportunidad de centralizar los manuales de soporte técnico, así como material multimedia para capacitar al personal antiguos y nuevos de acuerdo con las políticas cambiantes de la empresa, se reiteró la importancia de contar con un elemento moderno, coordinado e imparcial como un chatbot con IA amigable, capaz de no solo brindar soporte técnico básico a los usuarios, también de ofrecer entrenamiento, orientación y respuestas rápidas a los técnicos

actuales y ¿Por qué no? laboratorio de pruebas de telecomunicaciones, más infortunadamente no lograron ver ese potencial, lo cual, el proyecto fue dejado a un lado.

En este sentido y a la fecha de hoy, la empresa Iteractiva que participa en el proyecto Itel de Oracle para la empresa Claro Soluciones toma la importancia en acompañar el proceso de la implementación de un modelo del chatbot con IA en su versión básica que brinda todo lo anterior propuesto en este documento. Esta empresa se especializa en desarrollo de aplicaciones, mesa de servicios y otros procesos tecnológicos. También ofrece servicios de vanguardia que incluyen diversos procesos relacionados con bases de datos, servicios de capa media (MDW) y soporte tecnológico de monitoreo de nivel 1, siendo en este último donde participa el chatbot.

Se propuso entonces para el presente proyecto trabajar en conjunto con la empresa Claro Soluciones en el marco del proyecto llamado Itel de Oracle y liderado por el ingeniero Diego Amador. Allí teniendo en cuenta la necesidad expuesta, se planteó desarrollar un sistema o prototipo de chatbot con IA amigable que no solo contribuya a la resolución de problemas técnicos básicos de Nivel 1 (N1) (*¿Qué es y para qué sirve el Soporte Técnico Nivel 1?*, 2020, p. 1), también proporciona soporte continuo y actualizado tanto a los técnicos actuales como a los nuevos miembros del equipo técnico y usuarios finales.

Entonces, se planteó la siguiente pregunta de investigación: ¿Cómo desarrollar y operar un prototipo de software de asistente virtual con IA para una mesa de servicio técnico, que sea económico, sostenible, robusto, escalable y que ofrezca a su vez soluciones alternativas que faciliten la retención y transferencia del conocimiento técnico entre los usuarios y personal técnico?

Justificación

La revista Semana en el año 2020 citó lo siguiente: "*Además, se espera que el uso de chatbots siga creciendo a nivel mundial, con el 80% de las empresas adoptándolos en un año y los bancos automatizando hasta el 90% de sus servicios mediante estos sistemas para el 2022*" (Semana, 2020). A la fecha actual, el uso de chatbots se ha incrementado significativamente en el país y en el ámbito mundial de una manera desbordada. Otro dato interesante según el portal WebFindYou, indicaba que, en Colombia, el uso de chatbots creció mucho, aumentando un 380% entre febrero y junio solo en el año de 2020 (*Uso de Chatbot en Colombia Creció un 380 % en los Últimos Meses*, n.d.). Esto formaba parte de un proceso más grande de transformación digital que está ayudando a Colombia a destacarse en tecnologías de vanguardia. Este crecimiento representaba una noticia positiva en cuanto al uso de este tipo de aplicaciones de cualquier modelo disponible. Se concluye que, el uso de chatbots mejora la atención al cliente al brindar respuestas rápidas y efectivas, facilitando la comunicación con los clientes, ofreciendo formas de entrenamiento y manejando diversas consultas, incluyendo, creatividad ilimitada. Con el tiempo y robustez en el código para cada modelo, estas herramientas se automatizan, perfeccionan y aprenden por sí mismas, generando respuestas casi como lo hace un humano, pero de manera más eficiente y rápida.

Dicho esto, la necesidad de mejorar la atención al cliente y optimizar los procesos de soporte técnico no es algo novedoso hoy en día, especialmente con la transición a modalidades de trabajo remoto cada vez más gracias al avance de las telecomunicaciones y demás ingenierías participantes, incluyendo universidades que adoptaban ideas de vanguardia e investigación tecnológica y que llevan la delantera investigadora como la Universidad Nacional Abierta y a Distancia UNAD donde surge este proyecto. Empresas líderes como Claro, Microsoft y Cisco

destacan el impacto positivo de la implementación de chatbots en la calidad del servicio al cliente, reduciendo la sobrecarga de trabajo de los agentes de soporte y resolviendo problemas comunes (*IA Generativa En Las Empresas*, n.d.).

En el caso específico de la mesa de servicios que se vivió en MinTIC, se identificó una oportunidad de mejora para la atención al cliente de manera semi automatizada a través de un asistente virtual que pueda reconocer patrones para gestionar preguntas frecuentes y ofrecer respuestas rápidas en los canales de comunicación disponibles. Esta iniciativa buscaba no solo satisfacer la demanda de usuarios con alta frecuencia de consulta, sino también liberar recursos humanos para tareas más especializadas y complejas.

La experiencia previa con la atención al cliente demostró que el servicio mediante chat humano y telefónico era limitado en su capacidad de atención masiva, lo que llevaba a una alta demanda, pero una baja oferta, con una atención disminuida. Además, la falta de un sistema automatizado reflejaba la necesidad de innovación para mejorar la eficiencia y la calidad del servicio de soporte técnico, incluyendo, falta de madurez tecnológico de marca para la empresa que no logre adaptarse.

Aunque la información disponible sobre la adopción de tecnologías inteligentes como chatbots personalizados es limitada, la demanda de este recurso es clara, especialmente en empresas de tamaño mediano y pequeño que buscan mejorar su servicio al cliente y optimizar sus procesos internos. Este proyecto, al ser desarrollado bajo código abierto y con madurez en su desarrollo, tiene el potencial de ofrecer una solución accesible, barato, efectivo y personalizado para una amplia gama de aplicaciones inimaginables y compatibles en sus organizaciones.

Este proyecto de chatbot es importante e ideal ejemplar para otros estudiantes e investigadores y desarrolladores e interesados que guste desarrollar aplicaciones novedosas para

su negocio. Se conoce entonces que a la fecha de hoy los chatbots continúa en crecimiento, ya que es una herramienta consideradamente valiosa con potencial para implementarse en otras entidades sin ánimo de lucro, como, gubernamentales, empresas del sector privado, organizaciones educativas, salud, bienestar y otros. Su adopción contribuye a la mejora continua en la calidad del servicio al cliente, la eficiencia operativa y la analítica de datos en sectores como el documental, académico, salud, analítica de datos, entre otros.

La implementación de chatbots de soporte técnico personalizados permiten que las empresas que adopten este tipo de tecnologías mejoren significativamente la eficiencia operativa al automatizar la resolución de consultas frecuentes, reduciendo la carga laboral de los agentes, aumentando la satisfacción del cliente con respuestas rápidas y cercanas a la realidad, y asegura una transferencia continua de conocimientos así como reconocimiento de patrones para la corrección de fallas de nivel 1, todo lo cual contribuye a una mayor competitividad y mejora en la optimización de recursos. Otro aspecto que puede mejorar dentro de la cadena de negocio de una empresa es robustecer su portafolio de servicios complementarios que quieran con servicios de chatbots con inteligencia artificial.

Objetivos

Objetivo General

Desarrollar un prototipo de chatbot con inteligencia artificial utilizando Python, con el fin de proporcionar respuestas frecuentes, procesar automatizaciones dirigidas y seguimiento a requerimientos en soporte técnico y mesa de servicios de la empresa Claro Soluciones en el proyecto de Itel Oracle.

Objetivo Específico

Realizar el levantamiento de información de requerimientos e información de la empresa para entrenar el chatbot, abarcando sus complementos, código utilizado y programación. Esto permitirá que el equipo técnico y de desarrollo comprenda claramente la estructura y los componentes del sistema a través de documentación detallada.

Diseñar la herramienta de software a través de metodologías ágiles como Scrum.

Analizar los diferentes lenguajes de programación independientes de Python para evaluar posibles integraciones o mejoras en el rendimiento del chatbot mediante estudios comparativos y pruebas de desempeño con distintos lenguajes de programación.

Desarrollar la aplicación con la implementación de inteligencia artificial, teniendo en cuenta el diseño propuesto.

Realizar pruebas piloto del chatbot implementado en la empresa Claro Soluciones bajo el proyecto de Itel Oracle para detectar fallas y áreas de mejora en futuras versiones mediante diversas técnicas permitidas por negocio en el entorno controlado.

Marco de Referencia

En este capítulo de esta fase del proyecto, se presenta el marco conceptual de acuerdo con las definiciones dadas en cada investigación que pueden respaldar este proyecto de acuerdo con la teoría consultada en fuentes confiables.

Marco teórico

Chatbot (¿Qué es un chatbot?, n.d.): Un chatbot es un programa informático que emplea inteligencia artificial (IA) y procesamiento del lenguaje natural (NLP) que logra entender las preguntas de los usuarios y ofrecer respuestas automatizadas, imitando una conversación humana. A lo largo del tiempo, los chatbots han evolucionado desde versiones basadas en texto

con respuestas predefinidas hacia sistemas más sofisticados que utilizan comprensión del lenguaje natural (NLU). Estos chatbots modernos aprenden y se adaptan a las necesidades de los usuarios y del negocio, mejorando la experiencia del cliente en diversos contextos mediante la interacción conversacional.

Python (*El tutorial de Python*, n.d.), (*Introduction to Python*, n.d.) & (*Python*, n.d.) Como lo indica en la siguiente nota extraída del Tutorial de Python, "*Python es un lenguaje de programación potente y fácil de aprender.*" Esta herramienta de características especiales es un lenguaje de programación de alto nivel, interpretado y multipropósito. Es conocido por su sintaxis clara y legible, lo que facilita su aprendizaje y uso. Python es ampliamente utilizado en diversos campos, como desarrollo web, análisis de datos, inteligencia artificial, automatización de tareas y más. Es reconocido por su enfoque en la legibilidad del código y su amplia comunidad de desarrolladores, que ha contribuido al desarrollo de numerosas librerías y frameworks que amplían las capacidades del lenguaje. Este es el cerebro principal de la ejecución del código propuesto en la herramienta.

Procesamiento de Lenguaje Natural (PLN) (*¿Qué es el procesamiento del lenguaje natural (PLN)?*, 2021): El procesamiento de lenguaje natural (PLN) se puede decir que es una disciplina que combina la lingüística computacional con modelos estadísticos, de aprendizaje automático para permitir que las computadoras comprendan, interpreten y generen texto y voz de manera efectiva, lo cual resulta fundamental en aplicaciones que van desde la traducción automática y el reconocimiento de voz, hasta los chatbots y asistentes virtuales como Siri y Alexa, siendo capaz de ejecutar tareas como la evaluación de sentimientos, la autenticación de usuarios mediante la voz y la generación de contenido bajo demanda; su importancia radica en que permite a las máquinas interactuar con el lenguaje humano a través de técnicas avanzadas

como el etiquetado de parte de la oración, la desambiguación de palabras y la generación de lenguaje natural, empleando enfoques modernos que han evolucionado del PLN basado en reglas a modelos basados en redes neuronales, que permiten la optimización de procesos empresariales críticos, la detección de spam, el análisis de redes sociales, la creación de resúmenes automáticos y el análisis de grandes volúmenes de datos sin estructura, utilizando tecnologías como el Natural Language Toolkit (NLTK) y Python, que proporcionan las herramientas necesarias para desarrollar capacidades complejas como el razonamiento semántico y el análisis sintáctico, facilitando que las soluciones empresariales optimicen la eficiencia operativa y mejoren la interacción entre humanos y máquinas en aplicaciones de misión crítica.

Red Neuronal Recurrente (RNN) (*¿Qué son las redes neuronales recurrentes?*, 2023): IBM explica que, las redes neuronales recurrentes (RNN) son un tipo de red neuronal artificial diseñada para trabajar con datos secuenciales o series temporales. Estos modelos son ampliamente utilizados en tareas como traducción de idiomas, reconocimiento de voz y subtítulos de imágenes debido a su capacidad para comprender la secuencialidad de los datos, aunque para esta actividad, se utilizará para el procesamiento de texto a nivel de una base de datos personalizable.

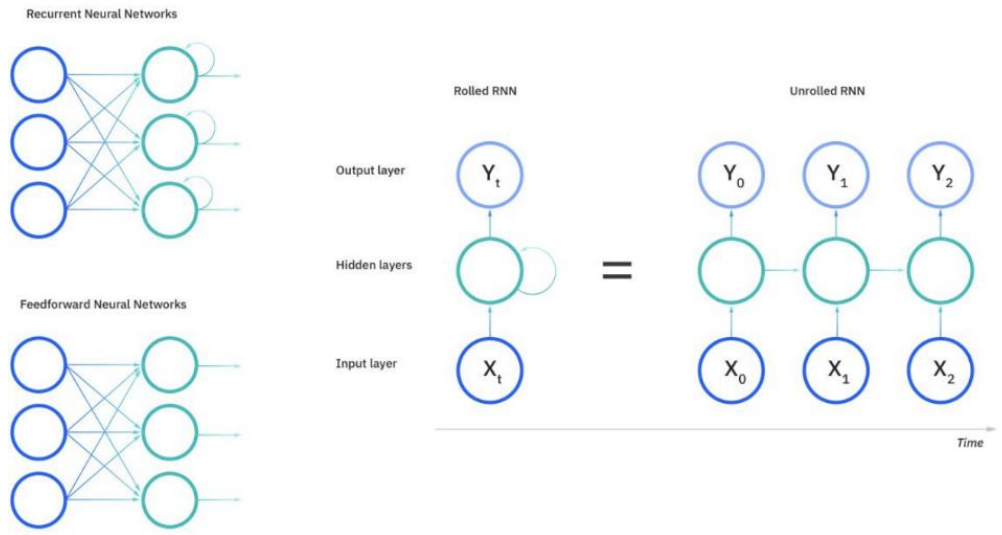
En la Figura 1 se ilustra el funcionamiento de las Redes Neuronales Recurrentes (RNN), lo que les permite utilizar información de entradas anteriores para influir en las salidas actuales. A diferencia de otras redes neuronales como las CNN (*¿Qué son las redes neuronales convolucionales?*, 2024) no aplicada para este proyecto, las RNN dependen de elementos anteriores en la secuencia para determinar la salida actual. Un ejemplo breve, dado un conjunto de palabras o caracteres, una RNN puede predecir la siguiente palabra o carácter en una

secuencia, permitiendo aplicaciones como la autocompletación en teclados de smartphones o la generación de texto coherente en chatbots.

Además, las RNN comparten parámetros en cada capa de la red, a diferencia de las redes de propagación hacia delante. Utilizan el algoritmo de retro propagación a través del tiempo (BPTT) para ajustar los parámetros del modelo, considerando la secuencia completa de datos en el proceso de entrenamiento.

Figura 1

Redes neuronales recurrentes (RNN)



Nota: La figura 1 muestra la diferencia entre las Redes Neuronales Alimentadas hacia Adelante (Feedforward Neural Networks) y las Redes Neuronales Recurrentes (RNN, por sus siglas en inglés). En las RNN, las capas ocultas tienen conexiones recurrentes que permiten procesar información secuencial y mantener un estado interno que captura dependencias temporales, a diferencia de las redes alimentadas hacia adelante, donde la información solo fluye en una dirección. Además, la representación "desplegada" (unrolled) de una RNN muestra cómo se procesan las entradas (X) a lo largo del tiempo, generando salidas (Y) en diferentes pasos

temporales, lo que permite modelar secuencias dinámicas y dependencias a lo largo del tiempo. *Fuente:* Tomada de: (*¿Qué son las redes neuronales convolucionales?*, 2024)

GitHub ("GitHub," 2024) & (*GitHub · Build and Ship Software on a Single, Collaborative Platform*, 2024)**:** Es una plataforma de red social de desarrollo colaborativo que permite a desarrolladores de software y fanáticos trabajar juntos en proyectos de código abierto y cerrado. Utiliza el sistema de control de versiones Git, que ayuda a los equipos a seguir y gestionar cambios en el código a lo largo del tiempo. También permite compartir proyectos, colaborar y aprender de otros.

Módulo Pickle (*¿Cómo Usar Pickle Para Guardar y Cargar Variables En Python? – Barcelona Geeks*, n.d.): Pickle es un módulo de Python que facilita serializar o guardar variables y cargarlas cuando sea necesario. A diferencia de la serialización JSON, Pickle convierte el objeto en una string binaria. JSON es específico del texto, pero Pickle es específico de Python y puede serializar las clases personalizadas que JSON no puede serializar.

Módulo Pandas (Team, n.d.) & (*Pandas - Python Data Analysis Library*, n.d.): Pandas es una librería de Python utilizada para el análisis y manipulación de datos. Proporciona estructuras de datos flexibles y eficientes, como DataFrames, que permiten realizar operaciones complejas en conjuntos de datos estructurados. Pandas facilita la limpieza y transformación de datos, el filtrado, la agregación y el análisis estadístico. Es una herramienta ampliamente utilizada en el ámbito del análisis de datos y la ciencia de datos debido a su capacidad para manejar y procesar grandes conjuntos de datos de manera eficiente. Con esta herramienta, se utiliza para la creación del diccionario y la consulta de preguntas y respuestas frecuentes solicitadas por el usuario, así como en sus pruebas piloto.

Módulo JSON (*Trabajando con JSON - Aprende desarrollo web | MDN*, 2022): Como lo indica en el articulado, JSON significa "*JavaScript Object Notation*" y es un formato de texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. Es comúnmente utilizado para transmitir datos en aplicaciones web (por ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o viceversa). Esta herramienta se utiliza ampliamente en aplicaciones web y servicios API para representar y transmitir datos estructurados. De acuerdo con otras fuentes, JSON se basa en una estructura de pares clave-valor y permite la representación de diferentes tipos de datos, como cadenas de texto, números, booleanos, listas y objetos anidados. Es fácilmente comprensible tanto por los humanos como por los programas, lo que lo convierte en un formato popular para el intercambio de datos entre diferentes sistemas y lenguajes de programación. En caso de aplicar, esta opción es utilizada para una base de datos inicial que contiene los parámetros entre preguntas y respuestas.

Módulo NLTK (*Tutorial NLTK*, 2024) : NLTK es una biblioteca de Python que contiene herramientas y programas para el procesamiento estadístico del lenguaje. Es una de las bibliotecas más poderosas para hacer que las máquinas comprendan el lenguaje humano y respondan de manera adecuada. Algunas de las funcionalidades que ofrece NLTK incluyen tokenización de palabras, etiquetado de partes del discurso, derivación, lematización, búsqueda de sinónimos, entre otros servicios.

Módulo TensorFlow Keras (sistemasgeniales, 2020): De acuerdo con la fracción del texto, TensorFlow es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas, y desarrollado por Google para satisfacer sus necesidades de sistemas capaces de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos

al aprendizaje y razonamiento usados por los humanos. Se utiliza ampliamente para construir y entrenar modelos de aprendizaje automático, incluyendo redes neuronales profundas. TensorFlow permite crear y ejecutar flujos de trabajo de aprendizaje automático de manera eficiente, distribuida y escalable. Proporciona una amplia gama de herramientas y funcionalidades para el procesamiento de datos, la construcción de modelos y la evaluación de resultados. TensorFlow es utilizado en diversas aplicaciones, como reconocimiento de imágenes, procesamiento de lenguaje natural, análisis de datos y más mediante la opción de Keras de TensorFlow. Con esta librería, se desarrollará el entrenamiento de la AI local en base a la database creada de manera anticipada, con ella, se crearán algoritmos y parones para el desarrollo de las palabras más relevantes y frecuentes. Con ello, el chatbot tiene la capacidad de dar respuestas formuladas un poco más precisa con la realidad similar a la respuesta dada por un humano.

Módulo RANDOM (*Qué significa Random*, n.d.), Según el artículo en relación, Random es una palabra del inglés traduciendo al español como aleatorio, fortuito o casual. Random, en este sentido, es aquello que depende del azar; que no obedece a ningún método o criterio, y cuyo resultado resulta siempre desconocido e inesperado. Estas funciones permiten generar números aleatorios enteros, números decimales, selecciones aleatorias de elementos de una lista y otras operaciones relacionadas con la aleatoriedad. La generación de números aleatorios es útil en diversos escenarios, como juegos, simulaciones, pruebas y muchas otras aplicaciones donde se requiere cierto grado de aleatoriedad en los datos o resultados. Para este chatbot, al dar múltiples respuestas, Random puede seleccionar cualquiera de las respuestas para el usuario mediante el chatbot propuesto que vayan relacionado con la misma pregunta.

Módulo OpenAI (*Openai/Openai-Python*, 2020/2024): El módulo OpenAI en Python es una interfaz de programación de aplicaciones (API) que permite acceder a los servicios y modelos desarrollados por OpenAI, una empresa de inteligencia artificial conocida por sus avances en el aprendizaje avanzado y el procesamiento del lenguaje natural. Con el chatbot propuesto, permite la interacción entre el código API de OpenAI y el propio chatbot, sirviendo como un apoyo durante el entrenamiento adicional para la corrección de errores. El modelo proporcionado se entrena de manera eficiente para funcionar de manera autónoma en el futuro, sin depender del modelo que ofrece OpenAI. Esta función estará disponible para todo el personal que seleccione el modelo avanzado del chatbot.

Módulo Flask (*¿Qué es Flask Python?*, 2023): Es un framework ligero y flexible para Python que facilita la creación de aplicaciones web tipo HTML. El módulo Flask se utiliza para mostrar mensajes temporales al usuario en el navegador web. Cuando se utiliza Flask, se puede enviar un mensaje como respuesta a una solicitud HTTP y ese mensaje se mostrará al usuario en la siguiente página que visite. Esto es útil para mostrar mensajes de éxito, errores, advertencias o cualquier otra información que desees comunicar temporalmente al usuario. Con este módulo, será la cara representativa del chatbot en cuanto a la interfaz web únicamente.

Módulo DuckDuckGo (*DuckDuckGo - Your Protection, Our Priority.*, n.d.) & ("DuckDuckGo," 2025) considerado como el "Motor de búsqueda independiente" y escogido para este proyecto de chatbot. Según Wikipedia, DuckDuckGo es un motor de búsqueda que prioriza la privacidad del usuario y evita el rastreo de datos. A diferencia de otros buscadores convencionales, este motor de búsqueda independiente no almacena información personal ni personaliza los resultados en función del historial de búsqueda del usuario. Este enfoque garantiza una búsqueda más anónima y neutral, permitiendo acceder a información relevante sin

comprometer la seguridad o privacidad. Esta herramienta cuenta con un módulo no oficial creado por fanáticos y expertos, lo cual hace importancia mencionar en este documento. Este módulo solo se activa en caso de que el chatbot no cuente con una respuesta adecuada, buscando por este medio y brinda como máximo de tres referencias halladas en internet, lo que garantiza que el chatbot de alguna manera brinda la respuesta esperada por el usuario.

Visual Studio Code o VS Code (*Qué es Visual Studio Code y qué ventajas ofrece | OpenWebinars*, n.d.) & (*Visual Studio Code - Code Editing. Redefined*, n.d.): Es considerado un buen entorno de edición de código fuente desarrollado por Microsoft que, en este caso, se utilizará principalmente como editor para Python; se trata de un software libre y multiplataforma compatible con Windows, GNU/Linux y macOS, lo que lo hace accesible para una amplia variedad de entornos de desarrollo. Entre sus características más destacadas se encuentra su compatibilidad multiplataforma, permitiendo a los desarrolladores trabajar en distintos sistemas operativos sin inconvenientes; su funcionalidad "IntelliSense" nativo, que ofrece autocompletado y resaltado de sintaxis para agilizar la escritura de código; una depuración integrada, que posibilita ejecutar el código con puntos de interrupción y una consola interactiva sin necesidad de herramientas externas; soporte nativo para Git, permitiendo a los usuarios revisar diferencias, realizar commits y gestionar versiones directamente desde el editor; y, finalmente, su capacidad extensible y personalizable, que brinda la posibilidad de instalar extensiones para agregar compatibilidad con nuevos lenguajes, temas y servicios, optimizando así la experiencia de desarrollo de acuerdo con las necesidades del usuario.

YouTube (*YouTube*, n.d.) & ("YouTube," 2024): YouTube es una plataforma de medios sociales y sitio web que permite a los usuarios cargar, ver y compartir videos, facilitando la difusión de contenido a nivel global. Fundada en 2005 por Chad Hurley, Steve Chen y Jawed

Karim, se ha convertido en una de las plataformas de video más populares del mundo. A través de canales abiertos, ofrece una amplia gama de videos caseros, películas, documentales, música y transmisiones en vivo. Con este último, se realizarán algunos tutoriales explicando procesos sencillos dentro de la operación del proyecto y tutoriales diversos del negocio.

Estado del arte

A continuación, se presentan diferentes investigaciones aplicadas en este proyecto desde varios puntos de vista con el problema de investigación, el desarrollo tecnológico y las aplicaciones como categorías principales del presente proyecto. Estos artículos científicos fueron consultados directamente desde la biblioteca de la universidad UNAD, así como de fuentes externas respetando el talento intelectual de sus contribuyentes.

A nivel internacional

Uno de los artículos que se referencia en este material es el de Jack Lasky a modo de investigación cuyo reposa en la biblioteca de la universidad “Chatbots” (Lasky, 20230301), Los chatbots han evolucionado gracias a la inteligencia artificial (IA), incluyendo el aprendizaje profundo y el procesamiento del lenguaje natural. Se dividen en dos tipos: stateless (sin memoria) y stateful (con memoria). Se utilizan ampliamente en aplicaciones comerciales para interactuar con clientes y en aplicaciones no comerciales, como el apoyo a pacientes con demencia. Aunque han avanzado, todavía enfrentan desafíos como la comprensión de respuestas complejas y la aceptación por parte de ciertos usuarios. El objetivo principal según el documento es presentar una visión general de la evolución de los chatbots, desde sus primeras versiones como ELIZA y PARRY hasta los chatbots modernos que utilizan tecnologías avanzadas de inteligencia artificial. Además, el texto destaca las aplicaciones comerciales y no comerciales de los chatbots, así como los desafíos y críticas que enfrentan en su uso. según planteado por Jack.

Un libro interesante considerado a nivel internacional del editorial Ra-Ma escrito por varios profesionales como Emilio Sora Olivas y Pablo Rodríguez Belenguer, entre otros, “Inteligencia Artificial – Casos prácticos con aprendizaje profundo” (*Inteligencia Artificial. Casos prácticos con Aprendizaje Profundo - Grupo Editorial RA-MA*, n.d.) y un ejemplar hallado en la biblioteca distrital Bibliored, trata de varias opciones con respecto al uso de los módulos para el uso de la inteligencia artificial. Adicional a ello, el libro abarca desde las primeras ideas sobre neuronas artificiales hasta aplicaciones avanzadas como Modelos Generativos y Aprendizaje Reforzado. Además, ofrece ejemplos prácticos y un repositorio de código en Google Colab. Dentro de este libro, el capítulo 4, "Modelos neuronales orientados a datos temporales", destaca el uso de redes recurrentes para el tratamiento de datos y la creación de chatbots. En este capítulo, se presentan varios casos prácticos que demuestran una efectividad del 80% utilizando el método SARIMA. El objetivo es ofrecer un concepto sobre la retención y el entrenamiento temporal de información recurrente. Aunque no se detalla la creación de un chatbot específico, se proporcionan ideas prácticas para desarrollarlo utilizando módulos claves como Keras y tensorflow dentro de las RNN.

A nivel nacional

Uno de los artículos que destaca es el de la revista Semana a modo de artículo propuesto por la empresa Ibang “La startup colombiana que con chatbots busca ayudar a las empresas en medio de la crisis” (Semana, 2020). En respuesta a la expansión del coronavirus, la tecnología ha sido un aliado crucial. Entre las herramientas destacadas se encuentran los chatbots, sistemas automatizados inteligentes que utilizan inteligencia artificial para interactuar con los usuarios y brindar servicios de atención al cliente. Durante la pandemia, los chatbots han sido especialmente importantes para las empresas, ya que permiten mantener la atención al cliente

mientras se protege la salud de los trabajadores. Sus ventajas incluyen respuestas inmediatas, escalabilidad, disponibilidad las 24 horas y estandarización del servicio. Además, son económicamente viables, ya que un caso por chatbot cuesta menos del 20% de una atención humana. Esta industria está en auge y se espera que un 80% de las empresas a nivel mundial utilicen chatbots en un año. Según Business Insider, los bancos podrían automatizar el 90% de sus servicios mediante chatbots para 2022. Empresas como iBang han donado la implementación de chatbots para apoyar a otras compañías durante la emergencia. El uso de chatbots ha experimentado un aumento significativo durante la pandemia, demostrando su valor en tiempos de crisis y revelando oportunidades para mejorar las operaciones empresariales. En cuanto a la tesis de J. A. García-García mencionada anteriormente como primer punto, también puede ser relevante para Colombia ya que se enfoca en la atención al cliente en línea.

Otro artículo que respalda la eficiencia de los chatbots a nivel nacional se encuentra publicado en la biblioteca de la UNAD ebsco gracias al profesional Pedro Ramos de Santis "Satisfacción del cliente en la logística: un análisis de chatbots en las empresas líderes de Colombia, Perú y Ecuador." (Ramos De Santis, 20240401) en el editorial especifica que, Los chatbots han avanzado significativamente desde sus inicios, adoptando tecnologías modernas y encontrando aplicaciones amplias y útiles, aunque deben superar ciertos desafíos para mejorar su aceptación y efectividad. Según esto, los chatbots son sistemas autónomos diseñados para brindar respuestas a preguntas frecuentes y guiar a los usuarios en diferentes consultas diferencia de sus modelos y versiones. Las empresas que han adoptado esta tecnología experimentan ventajas como una mejor atención al cliente, ahorro de tiempo y recursos humanos, segmentación de clientes y una comunicación mejorada. Los chatbots también ofrecen beneficios para los usuarios, como respuestas inmediatas, ubicuidad, accesibilidad y eficiencia en las

consultas. El objetivo del documento en relación de estos asistentes virtuales con Colombia es analizar y proponer mejoras en el uso de chatbots como herramienta para optimizar el servicio al cliente y la atención técnica, destacando las ventajas y desafíos específicos en el contexto colombiano. Dentro del texto, se encontró una estadística con el uso de estas herramientas automatizadas, el cual se toma como referencia para un análisis más profundo e interesante.

A nivel local

A pesar de no encontrarse información precisa, se conoce que locales comerciales que utilizan sistemas de chatbots con IA mediante otras aplicaciones como Facebook, Telegram, WhatsApp, entre otras herramientas, puede utilizar lenguajes Python y JavaScript de manera independiente con preguntas y respuestas frecuentes, los cuales tienen en cuenta al humano para preguntas más conclusas o que requieran tiempo. Por otro lado, otros fanáticos prefieren configurar estas herramientas mediante una interacción mediante token conectados a herramientas de interacción artificial como ChatGPT (“ChatGPT,” 2023), al considerarse una herramienta poderosa por el momento y en auge tanto en popularidad como evolución, lo cual interactúa con el usuario similar a la de un humano. Otras aplicaciones similares realizan la misma labor, ejemplo, Bard (Gemini), Copilot (Microsoft) y DeepSeck (deepseck) entre muchas a la fecha, aunque, estas últimas no se tratará en este documento. En caso dado de querer publicar esta tesis a un nivel dentro de las referencias bibliotecarias de la universidad, se debe preferir como un análisis a nivel local, ya que es una de estas opciones básicas y consideradas en su contexto.

Metodología

A continuación, se presentan las metodologías aplicadas, detalladas a través de la enumeración de fases. Además, se incluyen los objetivos relacionados, junto con una descripción detallada de las Actividades Programadas significativas.

Fases del proyecto

A continuación, se relacionan las fases del proyecto para llevar a cabo esta actividad:

Fase 1: Levantamiento de información de requerimientos

Objetivo relacionado: Realizar el levantamiento de información de requerimientos e información de la empresa para entrenar el asistente virtual, abarcando sus complementos, código utilizado y programación. Esto permitirá que el equipo técnico y de desarrollo comprenda claramente la estructura y los componentes del sistema a través de documentación detallada.

Actividades programadas

Reunión inicial con el equipo de trabajo para definir los requerimientos.

Recolección de información sobre los procesos y necesidades dentro del proyecto de Itel.

Análisis del entorno técnico actual (hardware, software, redes).

Documentación de los componentes del sistema, complementos y código utilizado.

Revisión y validación de la información recolectada con el equipo técnico y de desarrollo.

Fase 2: Diseño del software

Objetivo relacionado: Diseñar la herramienta de software utilizando metodologías ágiles como Scrum.

Actividades programadas

Formación del equipo Scrum (Product Owner, Scrum Master, equipo de desarrollo).

Definición del backlog del producto con user stories detalladas.

Planificación de sprints, asignación de tareas y establecimiento de metas.

Diseño de la arquitectura del chatbot y sus componentes principales.

Creación de prototipos y maquetas iniciales para revisión.

Fase 3: Análisis de lenguajes de programación

Objetivo relacionado: Analizar los diferentes lenguajes de programación independientes de Python para evaluar posibles integraciones o mejoras en el rendimiento del chatbot mediante estudios comparativos y pruebas de desempeño con distintos lenguajes de programación.

Actividades programadas

Identificación de lenguajes de programación alternativos a Python.

Realización de estudios comparativos sobre las características y rendimiento de cada lenguaje.

Desarrollo de pruebas de desempeño utilizando diferentes lenguajes.

Análisis de resultados y evaluación de posibles integraciones o mejoras en cuanto a producto piloto.

Fase 4: Desarrollo del prototipo

Objetivo relacionado: Desarrollar la aplicación con la implementación de inteligencia artificial, teniendo en cuenta el diseño propuesto.

Actividades programadas

Implementación del chatbot utilizando Python.

Integración de técnicas de inteligencia artificial para procesamiento de lenguaje natural.

Desarrollo de funciones básicas como respuestas frecuentes y automatizaciones dirigidas.

Realización de pruebas unitarias y de integración.

Revisión continua y ajustes basados en el feedback aplicando metodologías ágiles Scrum.

Fase 5: Pruebas y validación

Objetivo relacionado: Realizar pruebas al piloto del chatbot implementado en el proyecto de Itel Oracle para detectar fallas y áreas de mejora en futuras versiones mediante diversas técnicas permitidas en el entorno controlado.

Actividades programadas

Implementación del piloto del chatbot en un entorno controlado asignado para el proyecto Itel de Oracle.

Ejecución de pruebas funcionales y de estrés para detectar fallas.

Recopilación de feedback de los usuarios y stakeholders.

Identificación de áreas de mejora y ajustes necesarios.

Documentación de resultados y lecciones aprendidas.

Levantamiento de los requerimientos del sistema de chatbot

En este capítulo se detalla la estructura y los requisitos del proyecto de desarrollo de un chatbot con IA, abarcando desde el propósito y alcance del software, hasta las funcionalidades, requerimientos técnicos y operativos. Además, se describen las clases de usuarios, las interfaces y los criterios de evaluación de desempeño, proporcionando un marco integral para la implementación exitosa del sistema.

Información del proyecto

La siguiente tabla dentro del criterio técnico describe la estructura organizacional y los roles clave en el proyecto "Sistema de Chatbot con IA para soporte técnico en el proyecto Itel de Oracle," desarrollado para la empresa Claro Soluciones, incluyendo la fecha de preparación, los responsables del proyecto y el cliente asociado.

Tabla 1

Información del proyecto

Categoría	Descripción
Empresa / Organización	Oracle
Proyecto	Sistema de Chatbot con IA para soporte técnico en el proyecto Itel de Oracle
Fecha de preparación	20/06/2024
Cliente	Claro Soluciones
Patrocinador principal	Diego Arnamdo Amador
Gerente / Líder de Proyecto	Luis Andrés Bernal

Gerente / Líder de Análisis de negocio y requerimientos	William Betancourt Figueredo

Nota: Tabla que contiene información del proyecto de acuerdo con el documento de requerimientos de software.

Fuente: Autoría propia

Propósito

El propósito de este apartado es proporcionar una guía exhaustiva para el diseño del sistema de chatbot con IA destinado a mejorar el soporte técnico en el proyecto Itel de Oracle. Se tiene como objetivo principal, el cual es, especificar y detallar todos los aspectos necesarios para la construcción y puesta en marcha del chatbot. Los puntos específicos que se cubrirán incluyen:

Definición del proyecto

Nombre del Proyecto: Sistema de Chatbot con IA para soporte técnico en el proyecto Itel de Oracle.

Versión del Software actual: 1.0.0 (Puede variar según actualizaciones)

Alcance del proyecto

El alcance del proyecto implica el diseño y desarrollo de los siguientes módulos:

Módulo de procesamiento de lenguaje natural (PLN)

Este módulo se encarga de entender y procesar el lenguaje humano. Utiliza técnicas de PLN para interpretar las preguntas o comandos del usuario y generar respuestas adecuadas. Ayudará en la implementación de modelos de aprendizaje automático utilizando entrenamiento personal de bases de datos combinado con redes convulsionadas y recurrentes (RNN) o GPT para el análisis de preguntas y la generación de respuestas más progresivas.

Módulo de generación y gestión de requerimientos (REQ)

Este módulo automatiza la creación y seguimiento de tiquetes de tipo requerimientos de soporte, en adelante (REQ) de soporte cuando el chatbot no puede resolver una consulta por sí mismo. Ayudará en la funcionalidad para registrar un nuevo REQ, asignarlo a un técnico, mesa de servicio o dependencia correspondiente y proporcionar actualizaciones del estado del REQ. mediante el módulo Selenium por interno.

Módulo de integración con sistemas externos

Este nos facilita la conexión y comunicación entre el chatbot y otros sistemas o servicios externos, como bases de datos o plataformas de mensajería. Un ejemplo clave es consultas de búsqueda de información de bases de datos de Oracle como paquetes, promociones, estado de las líneas de auditoría global y servicio de monitoreo configurados para la toma de decisiones.

Módulo de análisis y reportes programados

El chatbot incluye funcionalidades para monitoreo y generación de reportes sobre el estado de los servidores y herramientas Oracle del proyecto. También permite generar informes en tiempo real a través de dashboards para otros proyectos de Oracle, como SPA y PCA también propuestos por el cliente. Estos informes pueden enviarse automáticamente al canal de WhatsApp en los canaels y grupos establecidos. Adicionalmente, el módulo proporciona gráficos sobre el entrenamiento diario programado, logs que evidencian el tiempo promedio de respuesta, y soporte de laboratorio enfocado en las telecomunicaciones.

Módulo de entrenamiento y mejora continua

Este módulo permite realizar el entrenamiento y la actualización de las bases de datos directamente a través del chatbot. Los usuarios pueden cargar preguntas y respuestas de manera directa al modelo, facilitando su personalización y mejora. Además, el chatbot incluye un sistema

de entrenamiento interno y mejora continua, que optimiza su capacidad para generar respuestas correctas o lo más cercanas posibles a la respuesta esperada.

Módulo de interfaz de usuario (UI)

La capa visible del chatbot que interactúa con los usuarios, proporcionando una interfaz gráfica amigable mediante el uso de Flask. Se tendrá una ventana de chat mediante el canal web como aplicación móvil compatible (si lo permite negocio), aplicando diseño de la interfaz de este módulo y aplicando diseño HTML5 para asegurar una experiencia de usuario única.

Módulo de documentación y ayuda

Proporciona documentación y ayuda para los usuarios finales y administradores del sistema mediante la opción de preguntas frecuentes y sugerencias, tutoriales para el uso del chatbot entre otros servicios adicionales. También incluye contenido de interés mediante consulta de apoyo DuckDuckGo en caso de no incluirse una respuesta directa o correcta.

Módulos de diagnóstico de telecomunicaciones.

Este módulo proporciona ayuda de verificación del funcionamiento de equipos y servidores de telecomunicaciones del proyecto Oracle - Itel. Este realiza procesos importantes como ping, telmet, tracert, Nslookup u otras herramientas adicionales según las necesidades específicas del entorno operativo.

Beneficios del proyecto

Eficiencia operativa: Mejora la eficiencia del soporte técnico mediante la automatización de respuestas a preguntas frecuentes.

Reducción de carga laboral: Libera recursos humanos para concentrarse en tareas más complejas.

Mejora en la atención al cliente: Provee respuestas rápidas y precisas, aumentando la satisfacción del cliente.

Descripción del entorno operativo

A continuación, se describe en una tabla detallando los requisitos técnicos necesarios para la implementación del chatbot propuesto, incluyendo tanto hardware como software necesarios o compatibles:

Tabla 2

Hardware y software requerido y opcionales

Categoría	Requisito	Descripción
Hardware	Computador	PC con al menos 8 GB de RAM y CPU de 4 núcleos para administradores y técnicos de soporte.
	Telecomunicaciones	Router y switches con soporte para tráfico de alta velocidad y redundancia.
	Sistema Operativo	Linux (Debian, Ubuntu) o Windows 10 o superiores para virtualización.
	Base de Datos	Oracle SQL para consultas, Microsoft Excel para almacenamiento de datos del chatbot.
Software	Lenguaje de Programación	Python 3.x para el desarrollo del chatbot y sus módulos.
	Frameworks de IA y PLN	TensorFlow, Keras, NLTK (Natural Language Toolkit) para el procesamiento de lenguaje natural.

Biblioteca de Interfaz Gráfica	Flask para aplicaciones web.
Sistema para Gestión de REQ	Módulos Selenium para interacción por procesamiento interno o de segundo plano. (Requiere interfaz gráfico para este procedimiento)
Seguridad y Encriptación	SSL/TLS para encriptación de datos en tránsito, herramientas de cifrado para datos en reposo, servicio reCAPTCHA y conexión VPN.
Herramientas de Desarrollo	Visual Studio Code, PyCharm o herramientas compatibles para el desarrollo y depuración del código.
Bibliotecas de Red y Diagnóstico	Librerías para ping, telnet, tracert, nslookup para el módulo de verificación de funcionamiento.
Plataforma de Monitoreo	Módulos con sintaxis creadas compatibles para las bases de datos configurados en proyecto Oracle mediante Python 3. x. de acuerdo con el modelo de negocio.

Nota: Tabla que contiene información de hardware y software requeridos y opcionales que requiere para la implementación de un chatbot.

Fuente: Autoría propia

De acuerdo con la información anterior, el chatbot se integrará con los siguientes canales de la empresa mediante la conexión directa con las herramientas como APEX de Gestión y Monitoreo proporcionado por negocio facilitando la comunicación interna y el soporte técnico en tiempo real mediante acceso directo. Se accede además con el servicio VPN FortiClient proporcionado por el cliente, el cual facilita seguridad y encriptación de la información.

Funcionalidades principales

A continuación, se detalla mediante descripción y beneficio cómo el chatbot automatizará el soporte técnico dentro de sus funcionalidades principales como la resolución de problemas, respuesta eficiente a preguntas frecuentes, gestión REQ, capacitación de empleados, monitoreo y diagnóstico de red en los servicios actuales, actualización continua basada en el aprendizaje, implementación de medidas de seguridad y privacidad de datos, y la provisión de una interfaz de usuario intuitiva y personalizable para mejorar la experiencia del usuario.

Respuesta a preguntas frecuentes

El chatbot puede responder automáticamente a una amplia variedad de preguntas frecuentes relacionadas con soporte técnico, basándose en una base de datos de respuestas predefinidas y aprendizaje continuo. Esto ayuda ahorra tiempo al resolver rápidamente las consultas comunes de los usuarios sin intervención humana.

Generación y gestión de requerimientos de soporte

Cuando el chatbot no puede resolver una consulta, ofrece opción para la generación de un REQ, el cual se asigna a una dependencia correspondiente para su seguimiento. Permite a los usuarios verificar el estado del REQ. en tiempo real y documentar cuando sea necesario. Esto ayuda a mejorar la eficiencia del proceso de soporte técnico y asegura que todas las consultas sean atendidas adecuadamente.

Capacitación y asistencia a técnicos

El chatbot proporciona recursos de capacitación y asistencia técnica a los nuevos agentes y al personal existente, ayudando a resolver problemas técnicos de manera más efectiva. Cuenta con información KEDB Wiki proporcionado por el proyecto, lo que facilita la transferencia de conocimiento y reduce el tiempo de capacitación para nuevos técnicos.

Monitoreo y diagnóstico de red

El chatbot incluye herramientas para verificar el estado de servidores y equipos de red mediante comandos como ping, telnet, tracert, nslookup y envío de reportes programadas sea por el canal de WhatsApp como en su interfaz. Esto permite a los técnicos diagnosticar y solucionar problemas de conectividad, rendimiento de la red y estado de servicios de servidores y manejados virtuales de manera eficiente.

Seguridad y privacidad de datos

Se implementa cifrado de datos y políticas de privacidad para proteger la información de los usuarios y cumplir con las normativas de protección de datos de acuerdo con las leyes vigentes del país. Esto asegura que los datos de los usuarios estén protegidos contra accesos no autorizados y posibles brechas de seguridad.

Interfaz de usuario

El chatbot ofrece una interfaz de usuario amigable y fácil de usar, con opciones de personalización y accesos directos para adaptar la experiencia del usuario a sus necesidades específicas del negocio, lo que mejora la experiencia del usuario al hacer que la interacción con el chatbot sea simple y eficiente.

Requerimientos funcionales, no funcionales y otros requerimientos

A continuación, encontrará una descripción detallada de los requerimientos funcionales del chatbot, los cuales abarcan las funcionalidades específicas que el sistema debe cumplir, desde la autenticación de usuarios hasta la generación REQ. y el procesamiento de lenguaje natural. Además, se incluyen los requerimientos no funcionales que establecen los criterios de rendimiento, seguridad, usabilidad, escalabilidad, y mantenibilidad del sistema, garantizando su operación eficiente y segura. Por último, se presentan los otros requerimientos que contemplan

mejoras adicionales como la integración de un modo oscuro en la interfaz, interconexión con otros modelos como GPT y otros elementos clave para una elaboración exitosa de esta herramienta.

Requerimientos funcionales

Para este punto incluye una descripción detallada de las funcionalidades específicas que el sistema debe cumplir, abarcando desde la aprobación de reCAPTCHA por usuario y la generación REQ. de soporte en caso de ser requerido, hasta la capacidad de responder preguntas frecuentes utilizando técnicas avanzadas de procesamiento de lenguaje natural, asegurando que cada interacción del usuario con el chatbot sea eficiente, precisa y contribuya a la optimización de los procesos de soporte técnico de la empresa.

Se incluye una lista numerada en cuanto a los requerimientos funcionalidades del chatbot:

Procesamiento de lenguaje natural (PLN)

El chatbot debe ser capaz de entender y procesar el lenguaje natural en el que los usuarios formulan sus preguntas y comandos.

Requerimientos aplicados

REQ-1: El chatbot debe utilizar técnicas avanzadas de PLN para interpretar y responder a consultas formuladas en lenguaje natural.

REQ-2: El chatbot debe reconocer sinónimos, variaciones de frases y errores ortográficos comunes para proporcionar respuestas precisas.

REQ-3: La aplicación debe adaptarse y aprender continuamente de las interacciones para mejorar su capacidad de comprensión y respuesta. En caso de no, tener la opción para su entrenamiento en cuanto a retroalimentaciones manuales y API GPT.

Autenticación de usuarios

Garantizar que solo usuarios autorizados puedan acceder a las funcionalidades del chatbot.

Requerimientos aplicados

REQ-4: El usuario debe acceder a la herramienta del chatbot a través de las credenciales de APEX proporcionado por negocio para lograr su interacción con la herramienta a través de una VPN activado.

REQ-5: El chatbot debe implementar un sistema de autenticación basado en reCAPTCHA de seguridad, dado que el acceso está protegido a través de una VPN.

Respuesta a preguntas frecuentes

Proveer respuestas rápidas y automatizadas a las consultas más comunes de los usuarios.

Requerimientos aplicados

REQ-6: El chatbot debe acceder a una base de datos de preguntas frecuentes y devolver respuestas precisas en tiempo real.

REQ-7: La aplicación debe permitir a los administradores y técnicos autorizados actualizar la base de datos de preguntas frecuentes con nuevas preguntas y respuestas y entrenar el modelo.

REQ-8: El chatbot debe registrar todas las interacciones con usuarios mediante logs de seguimiento adecuado con fecha y hora del sistema para analizar la efectividad de las respuestas y ajustar las respuestas futuras según el feedback recibido.

Generación y seguimiento de requerimientos

Sugiere y brinda paso a paso en la creación de REQ. de soporte mediante la interacción directa a la herramientita de gestión MyIT cuando el chatbot no puede resolver una consulta

directamente, adicionalmente, acceder a la gestión REQ. para la documentación de comentarios en tiempo real.

Requerimientos aplicados

REQ-9: El chatbot debe brindar paso a paso para la generación de REQ. en segundo plano dentro de la herramienta de MyIT en caso de no lograr resolver una consulta básica o si el usuario solicita la creación de un nuevo REQ.

REQ-10: El sistema debe permitir a los usuarios consultar el estado de sus REQ. a través del chatbot.

Capacitación y soporte a técnicos

Proporcionar recursos de capacitación y asistencia técnica a los empleados, especialmente a los nuevos técnicos.

Requerimientos aplicados

REQ-11: El chatbot debe ofrecer módulos de capacitación interactivos que guíen a los técnicos a través de procedimientos estándar, o al menos, tomar la información de KEDB de la Wiki proporcionado por el proyecto para la orientación a los técnicos y usuarios finales.

REQ-12: El sistema debe permitir a los técnicos acceder a guías y manuales actualizados sobre procedimientos de soporte.

Laboratorio de pruebas de telecomunicaciones

Facilitar la verificación y monitoreo de equipos de telecomunicaciones utilizando herramientas de diagnóstico integradas.

Requerimientos aplicados

REQ-13: El chatbot debe ejecutar pruebas de conectividad, como ping, telnet, estado de memoria y CPU a solicitud del usuario o automáticamente como parte de un proceso de diagnóstico.

REQ-14: El sistema debe realizar un seguimiento de las rutas de red mediante tracert y proporcionar un análisis de posibles puntos de falla.

REQ-15: El chatbot debe resolver nombres de dominio a direcciones IP utilizando nslookup y ofrecer la posibilidad de verificar la disponibilidad de servicios.

REQ-16: El chatbot debe reportar estado de los servicios PCA y SPA a través de canal de WhatsApp dentro de ciertas horas y a ciertos canales o grupos de negocio.

Sistema de gestión y monitoreo

Proveer herramientas para gestionar y monitorear el rendimiento del chatbot y los sistemas asociados, como bases de datos, redes, manejados WebLogic.

Requerimientos aplicados

REQ-17: El chatbot debe incluir comandos específicos donde los administradores y técnicos puedan monitorear el rendimiento del sistema en tiempo real, incluyendo tiempos de respuesta de las bases de datos Venecia y Triara, estado de los manejados de WebLogic, etc.

REQ-18: El sistema debe generar alertas y notificaciones automáticas para los administradores cuando se detecten tiempos de respuesta altos o fallas en las transacciones en fase 1 y fase 2. Este proceso hará parte del monitoreo en cuanto a servicios de servidores y telecomunicaciones.

REQ-19: El chatbot debe enviar reporte por el canal de WhatsApp el estado de los manejados WebLogic como el estado del filesystem de Gestor documental (SPA), así como el estado de los servicios PCA y los servicios de Aprovisionador e Itel.

Requerimientos no funcionales

Los requerimientos no funcionales abarcan especificaciones detalladas sobre el rendimiento del sistema, asegurando que responda rápidamente a las consultas de los usuarios, así como medidas de seguridad robustas para proteger la información sensible; también incluyen aspectos de usabilidad para garantizar una experiencia de usuario intuitiva y accesible, y mantenibilidad, permitiendo que el sistema sea fácilmente actualizable y gestionable a lo largo del tiempo.

Rendimiento

El chatbot debe funcionar con un alto nivel de eficiencia para manejar múltiples interacciones simultáneas sin degradar la experiencia del usuario.

Requerimientos aplicados

REQ-20: El chatbot debe responder a las consultas de los usuarios en un tiempo máximo de 10 segundos bajo una carga estándar de usuarios.

REQ-21: El sistema debe ser capaz de manejar al menos 200 sesiones concurrentes sin pérdida significativa de información y rendimiento.

REQ-22: El rendimiento del chatbot debe ser monitoreado continuamente, y cualquier caída en la velocidad y procesamiento de respuesta debe ser detectada y corregida automáticamente.

Escalabilidad

El sistema debe ser capaz de escalar para manejar un aumento en el volumen de usuarios y datos sin necesidad de rediseños significativos.

Requerimientos aplicados

REQ-23: La interfaz del chatbot debe ser escalable horizontal y verticalmente para soportar un crecimiento exponencial en el número de usuarios.

REQ-24: El sistema debe permitir la integración de nuevos módulos o servicios sin afectar la funcionalidad existente.

REQ-25: La infraestructura debe ser capaz de expandirse para manejar picos en la demanda sin interrupciones del servicio.

Seguridad

El chatbot debe garantizar la protección de los datos y la privacidad de los usuarios, implementando medidas de seguridad robustas.

Requerimientos aplicados

REQ-26: Todos los datos sensibles manejados por el chatbot, incluidos los datos de autenticación, deben ser cifrados tanto en tránsito como en reposo.

REQ-27: El sistema debe incluir medidas de protección contra ataques comunes, como inyecciones SQL, ataques de fuerza bruta y ataques de denegación de servicio (DDoS).

REQ-28: El chatbot debe estar alineado con las normativas de seguridad y privacidad descrita en este documento, asegurando el cumplimiento legal en todas sus operaciones.

Usabilidad

El chatbot debe ser fácil de usar y accesible para todos los usuarios, independientemente de su nivel de experiencia técnica.

Requerimientos aplicados

REQ-29: La interfaz de usuario del chatbot debe ser intuitiva, con un diseño que permita a los usuarios realizar consultas y recibir respuestas sin necesidad de formación previa.

REQ-30: El chatbot debe ser accesible a través de múltiples dispositivos, incluidos ordenadores, tablets y smartphones, manteniendo una experiencia de usuario coherente.

REQ-31: El sistema debe incluir opciones de accesibilidad, como modo lectura en voz alta para personas con incapacidad visual.

Mantenibilidad

El chatbot debe ser fácil de mantener y actualizar, permitiendo la adición de nuevas funciones y la corrección de errores con un mínimo de interrupciones.

Requerimientos aplicados

REQ-32: El código del chatbot debe estar bien documentado y estructurado de manera modular para facilitar el mantenimiento y la actualización.

REQ-33: Las actualizaciones del sistema deben poder implementarse sin necesidad de apagar o reiniciar el chatbot, minimizando el tiempo de inactividad.

REQ-34: El sistema debe incluir mecanismos de prueba automática que permitan verificar la funcionalidad después de cada actualización o modificación.

Fiabilidad

El chatbot debe ser altamente fiable, garantizando un tiempo de actividad cercano al 100% y minimizando los errores del sistema.

Requerimientos aplicados

REQ-35: El chatbot debe garantizar una disponibilidad mínima del 99.9% durante el horario laboral.

REQ-36: El sistema debe ser capaz de recuperarse automáticamente de fallos menores sin intervención manual.

REQ-37: Los logs de errores y eventos deben ser generados y almacenados automáticamente en archivo llamado logs para permitir una rápida resolución de problemas.

Compatibilidad

El chatbot debe integrarse sin problemas con los sistemas existentes en la empresa y con los servicios de terceros que se utilicen.

Requerimientos aplicados

REQ-38: El chatbot debe ser compatible con las bases de datos Venecia y Triara de Claro Soluciones, asegurando que las consultas y actualizaciones se realicen sin errores.

REQ-39: El chatbot debe estar diseñado para ser interoperable con otros sistemas de comunicación, como Apex, servicio de mensajería y correos electrónicos corporativos.

Otros Requerimientos

Los otros requerimientos incluyen mejoras adicionales que enriquecen la funcionalidad del chatbot, tales como la implementación de un modo oscuro en la interfaz para mayor comodidad del usuario, la integración con modelos avanzados de lenguaje como GPT para respuestas más precisas; además, se contempla la muestra de alertas y notificaciones del servicio, lo que garantiza una experiencia más completa, flexible y adaptada a las necesidades específicas de los usuarios.

Mejora de la interfaz - modo oscuro

Implementación de un modo oscuro para la interfaz del chatbot, permitiendo a los usuarios cambiar entre el modo claro y oscuro según su preferencia.

Requerimientos aplicados

REQ-40: El chatbot debe incluir una opción en la configuración que permita a los usuarios activar o desactivar el modo oscuro.

REQ-41: La interfaz debe ser completamente funcional y mantener la usabilidad en ambas configuraciones (claro y oscuro).

REQ-42: El modo oscuro debe ser accesible desde todos los dispositivos compatibles, incluyendo ordenadores, tablets y smartphones (Este último si el área de negocio lo autorice).

Interconexión con GPT

Integración del chatbot con un modelo de lenguaje avanzado como GPT para mejorar la calidad y relevancia de las respuestas.

Requerimientos aplicados

REQ-43: El chatbot debe ser capaz de realizar consultas a un modelo GPT para generar respuestas más precisas y naturales en casos complejos.

REQ-44: El sistema debe permitir la actualización y entrenamiento continuo del modelo chatbot con modelo GPT para adaptarse a nuevas tendencias y necesidades del usuario.

Alertas y notificaciones

Habilitar la capacidad del chatbot para enviar notificaciones basadas en el comportamiento de los servicios del proyecto Itel de Oracle.

Requerimientos aplicados

REQ-45: El chatbot debe enviar alertas y notificaciones proactivas a los equipos de los técnicos según el estado del servicio de monitoreo del proyecto Itel de Oracle mediante canal de WhatsApp.

Reglas de negocio

A continuación encontrará las reglas y políticas que regirán el uso y operación del chatbot, de acuerdo con el documento de privacidad y seguridad de Oracle compatible nuevas aplicaciones, incluyen la restricción del acceso solo a personal autorizado, la autenticación mediante sistemas seguros, la protección de datos sensibles mediante cifrado, la revisión y actualización periódica de las respuestas generadas por el chatbot, el monitoreo continuo del rendimiento y la seguridad, y el cumplimiento estricto de todas las normativas de protección de datos, asegurando que todas las operaciones y transacciones se realicen dentro de un marco seguro y legalmente conforme.

Privacidad y protección de datos

Política: La privacidad de los datos de los usuarios será prioritaria y se manejará con estricta confidencialidad.

Normativa: Los datos sensibles serán cifrados tanto en tránsito como en reposo utilizando estándares de cifrado como SSL/TLS y política de captcha.

Uso adecuado

Política: El chatbot deberá ser utilizado exclusivamente para fines relacionados con el soporte técnico y las operaciones de la empresa.

Normativa: Cualquier uso indebido del chatbot, como el uso para fines personales o la manipulación de datos, será sancionado según las políticas internas de la empresa.

Registro y monitoreo de actividades

Política: Todas las interacciones y actividades realizadas a través del chatbot serán registradas y monitoreadas para garantizar la seguridad y el cumplimiento de las políticas.

Normativa: Se mantendrán logs detallados de todas las actividades, los cuales serán revisados periódicamente por el equipo de seguridad de TI.

Actualización y mantenimiento

Política: El chatbot será actualizado regularmente para mejorar su funcionalidad y seguridad.

Normativa: Se realizarán actualizaciones periódicas y mantenimiento preventivo basado en un calendario establecido, y cualquier actualización de emergencia será implementada de inmediato para resolver vulnerabilidades críticas.

Capacitación y soporte

Política: Los usuarios recibirán capacitación adecuada para utilizar el chatbot de manera efectiva y eficiente.

Normativa: Se proporcionarán manuales de usuario, tutoriales y sesiones de capacitación regular para todos los usuarios, y se establecerá un canal de soporte dedicado para resolver cualquier duda o problema.

Cumplimiento legal

Política: El uso del chatbot deberá cumplir con todas las leyes y regulaciones aplicables, incluyendo aquellas relacionadas con la protección de datos y la privacidad.

Normativa: Se realizará una revisión legal periódica para asegurar que el chatbot cumpla con las normativas locales e internacionales relevantes, y se implementarán cambios conforme sea necesario para garantizar el cumplimiento continuo.

Impacto empresarial

El impacto del chatbot dentro del proyecto Itel de Oracle. se espera que sea transformador, afectando positivamente múltiples áreas de la organización y alineándose con los

objetivos estratégicos de la empresa. A continuación, se detallan los principales efectos esperados:

Mejora de la eficiencia operativa

Automatizar de tareas repetitivas: El chatbot permitirá la automatización de tareas repetitivas de soporte técnico, como la respuesta a preguntas frecuentes, la generación de requerimientos (REQ) y el monitoreo de sistemas, lo que reducirá significativamente la carga de trabajo manual de los empleados.

Reducir costos operativos: Al automatizar procesos que anteriormente requerían intervención humana, se reducirá la necesidad de recursos humanos adicionales, lo que contribuirá a una optimización de los costos operativos de la empresa.

Optimizar tiempo de respuesta: La capacidad del chatbot para proporcionar respuestas inmediatas y resolver consultas en tiempo real mejorará la eficiencia de los procesos de soporte, minimizando los tiempos de espera y aumentando la productividad general del equipo.

Optimización del talento humano

Enfocar en actividades estratégicas: Al liberar a los empleados de tareas repetitivas y de bajo valor añadido, se permitirá que el personal se enfoque en actividades más complejas, estratégicas y creativas, como la innovación de productos, la mejora de servicios y la gestión de relaciones con clientes clave.

Desarrollar habilidades especializadas: Con la reducción de la carga de tareas operativas, los empleados podrán dedicarse al desarrollo de habilidades más especializadas, lo que incrementará la capacidad de la empresa para abordar desafíos técnicos avanzados y proyectos de alto impacto.

Incremento de la satisfacción del cliente

Respuestas rápidas y precisas: La capacidad del chatbot para proporcionar respuestas inmediatas y precisas a las consultas de los clientes mejorará la experiencia del usuario, reduciendo la frustración y aumentando la satisfacción general del cliente.

Disponibilidad 24/7: Con el chatbot operando de manera continua, los clientes podrán recibir soporte en cualquier momento, independientemente de la hora o del día, lo que generará una mayor confianza y lealtad hacia la empresa.

Personalización de la experiencia del cliente: El chatbot podrá ofrecer un soporte más personalizado al aprender de las interacciones previas con los usuarios, ajustando sus respuestas y recomendaciones para satisfacer mejor las necesidades individuales de cada cliente.

Escalabilidad y adaptabilidad

Capacidad de crecimiento sin barreras: El diseño escalable del chatbot permitirá que la empresa maneje un crecimiento exponencial en la demanda de soporte sin necesidad de realizar inversiones significativas en infraestructura adicional o personal.

Adaptación a nuevas necesidades: A medida que el proyecto crece y evoluciona o surjan otros proyectos de Oracle, el chatbot podrá ser actualizado y adaptado para incluir nuevas funcionalidades, integrar tecnologías emergentes y ajustarse a cambios en las necesidades del negocio y del mercado.

Facilitación de la innovación continua: La plataforma flexible del chatbot servirá como una base sólida para la innovación, permitiendo que el proyecto Itel de Oracle implemente rápidamente nuevas ideas y tecnologías que mejoren la competitividad y el liderazgo en el mercado.

Cumplimiento con normativas y mejora de la seguridad

Gestión segura de datos: El chatbot asegurará que los datos de los clientes y las interacciones sean manejados de manera segura, cumpliendo con todas las normativas de protección de datos como el GDPR, lo que mitigará riesgos legales y mejorará la confianza del cliente en la empresa.

Resultados esperados

El producto final esperado incluirá un software de chatbot elaborado en Python, completamente funcional y optimizado para brindar apoyo al soporte técnico de la empresa. Este sistema no solo será capaz de manejar consultas y tareas repetitivas con eficiencia, sino que también se integrará de manera fluida con los sistemas existentes de la empresa y sus APIs, como las bases de datos y plataformas de comunicación.

El chatbot estará acompañado de una documentación técnica exhaustiva, que incluirá:

Manuales de usuario: Guías paso a paso diseñadas para que los usuarios finales puedan interactuar con el chatbot de manera intuitiva y resolver sus problemas sin necesidad de asistencia adicional.

Guías de administración: Instrucciones detalladas para los administradores del sistema sobre cómo configurar, personalizar y mantener el chatbot, asegurando que puedan adaptarlo a las necesidades cambiantes de la empresa.

Especificaciones técnicas: Documentación técnica que incluye la arquitectura del sistema, diagramas de flujo, dependencias de software, y procedimientos de mantenimiento, destinada a los equipos de desarrollo y TI para facilitar futuras actualizaciones y mejoras del sistema.

Tutoriales en YouTube: Instrucciones similares a los manuales, esta será más de interacción multimedia en caso de que los manuales físicos no son suficientes.

Además, el chatbot estará diseñado para ser escalable y adaptable, permitiendo su fácil expansión a medida que el proyecto crece, y su rendimiento será continuamente monitoreado para asegurar que cumpla con los estándares de calidad y eficiencia establecidos. Este conjunto de resultados garantiza que tanto los usuarios finales como los administradores puedan utilizar y mantener el sistema de manera efectiva, facilitando su integración y operación continua en el entorno empresarial, contribuyendo así a los objetivos estratégicos de la organización.

Implementación de metodología Scrum

Este capítulo describe cómo se aplicará una de las metodologías ágiles conocidas como Scrum aplicado en el desarrollo del chatbot para el proyecto de Itel de Oracle, abarcando desde la planificación y organización del backlog hasta la implementación de cada componente, con un enfoque en la colaboración, procesos iterativos, el uso de herramientas y tecnologías integradas. Adicionalmente, se describe algunas metodologías existentes en el mercado para su comparativa.

Diseño y aplicación de metodologías agiles en el proyecto

Una metodología puede definirse como: *"La metodología es el conjunto de métodos y técnicas aplicadas durante un proceso de investigación para alcanzar resultados teóricamente válidos. En este sentido, la metodología actúa como un soporte conceptual que guía los procedimientos de la investigación, asegurando precisión y fiabilidad en los resultados obtenidos."* (*Metodología*, n.d.) esto quiere decir que, este tipo de disciplinas son bases fundamentales para ser aplicados en nuestro trabajo con la finalidad de hallar un proyecto más eficiente y ordenado. Existen dos tipos de metodologías globales que puedan ser útiles a la hora de tomar decisiones: **Metodologías Tradicionales** y **Metodologías Ágiles**.

Según en lo informado de la página web Library, "Las metodologías tradicionales son aquellas utilizadas como guía para garantizar la creación de un producto con un nivel de calidad definido. Se caracterizan por tener un mayor énfasis en la planificación y control del proyecto, en especificaciones precisas de requisitos y modelado". (*Metodologías tradicionales - Marco Teórico*, n.d.). Se conoce también como un seguimiento de un estándar, un proceso ya definido por fases dentro de esta disciplina que al igual que las metodologías ágiles pueda ayudar con un resultado final con ciertas limitaciones y variedad de tiempos. Para esta disciplina, se conocen algunas metodologías tradicionales como por ejemplo el Modelo Cascada, Modelo V, Modelo

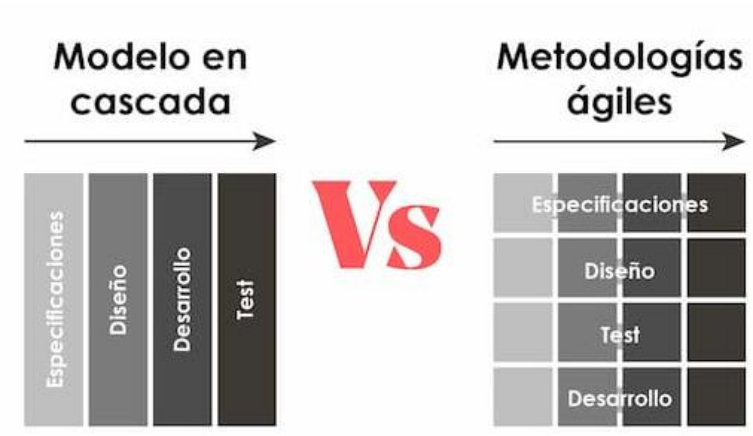
Espiral, entre otros. Cualquiera de ellas y realizando su paso a paso se puede construir un chatbot con las funcionalidades necesarias.

En otros aspectos, según en la documentación tomada de la página página web PlainConcepts, “Las metodologías ágiles son enfoques de gestión de proyectos que enfatizan la flexibilidad, la colaboración y la entrega continua de valor. A diferencia de los métodos tradicionales, que suelen ser lineales y rígidos, las metodologías ágiles dividen los proyectos en ciclos cortos llamados "sprints" o iteraciones.” (*Metodologías ágiles*, 2021) Para el en caso del chatbot, implica dividir el proyecto por ciclos, esto ayuda a la adaptación a cambios y a la entrega contínua del valor. Además, promueve la colaboración entre desarrolladores, diseñadores y otros involucrados, asegurando que el chatbot se ajuste a las necesidades reales de los usuarios. Aquí Scrum juega un papel importante para este desarrollo del chatbot. En este caso, se ha seleccionado esta metodología ágil, ya que se cuenta con las certificaciones correspondientes.

En la figura 2 tomada de la página Positivo Pro, resume de una manera más concisa las diferencias entre las metodologías ágiles vs metodologías tradicionales:

Figura 2

Metodologías de desarrollo de software tradicionales vs ágiles



Nota: Figura que muestra una comparativa entre las metodologías tradicionales y metodologías ágiles.

Fuente: Información tomada de Positivo Pro ("Metodologías Agiles vs Tradicionales," n.d.)

La siguiente tabla se realiza una comparativa entre las metodologías tradicionales vs metodologías ágiles de acuerdo con el criterio del autor de esta tesis:

Tabla 3

Comparativa entre metodologías ágiles vs tradicionales

Aspecto	Metodologías Tradicionales (Cascada)	Metodologías Ágiles (Scrum)
Enfoque	Lineal y secuencial	Iterativo e incremental
Flexibilidad	Baja: Cambios son costosos y difíciles de implementar	Alta: Se adapta fácilmente a cambios en los requisitos
Entrega de Producto	Al final del ciclo de desarrollo	Entregas parciales y funcionales en cada iteración
Participación del Cliente	Limitada al inicio y al final	Constante durante todo el proceso
Control de Calidad	Se realiza al final del desarrollo	Se realiza continuamente en cada iteración
Adaptabilidad	Baja, adecuada para proyectos con requisitos claros y estables	Alta, ideal para proyectos con requisitos cambiantes

Nota: Se hace una descripción comparativa entre las metodologías tradicionales vs metodologías ágiles.

Fuente: Autoría propia

Como se puede observar en la tabla las metodologías ágiles ofrecen múltiples ventajas que se hacen especialmente útiles en la gestión de proyectos, en este caso específico para el desarrollo del chatbot puede ser flexible ya que puede brindar soluciones en la ingeniería con base en su teoría. A continuación, se destacan algunas de las principales razones para utilizar esta metodología elegida en la tabla 4:

Tabla 4

Razones para usar metodologías ágiles en el diseño del chatbot Itel

Razón para usar metodologías ágiles	Descripción
Adaptabilidad al cambio	Permiten ajustar los proyectos en respuesta a requisitos cambiantes, esencial en entornos dinámicos.
Mejora de la calidad del producto	Dividen el trabajo en iteraciones cortas para facilitar la detección y corrección temprana de errores.
Mayor satisfacción del cliente	Entregan incrementos funcionales continuamente y permiten la participación activa del cliente en el proceso.
Incremento de la productividad	Organizan el trabajo en ciclos cortos y priorizan tareas esenciales, optimizando recursos y tiempos.
Fomento de la colaboración	Promueven comunicación efectiva y trabajo en equipo, resultando en soluciones más innovadoras y eficientes.
Transparencia y control	Dividen el proyecto en etapas con entrega de resultados parciales, permitiendo una mejor supervisión y ajuste.

Nota: Se hace una descripción resumida en cuanto a las ventajas y limitaciones de las metodologías ágines en el diseño del chatbot.

Fuente: (Principales ventajas y limitaciones de las metodologías ágiles, 2016), (¿Cuáles son los beneficios de utilizar la metodología ágil?, n.d.) & (programacionpro.com, 2024).

Metodología Scrum aplicada al chatbot de Itel

Descripción de Scrum

Scrum, es un marco de trabajo ágil ampliamente utilizado en la gestión de proyectos, especialmente en el desarrollo de software, debido a su capacidad para gestionar la complejidad y adaptarse rápidamente a los cambios. Basado en ciclos de desarrollo cortos llamados sprints, Scrum facilita la entrega incremental de valor al cliente, lo que es especialmente efectivo en proyectos donde los requisitos cambian con frecuencia.

Componentes clave de Scrum

Roles definidos

Scrum Master

Es el encargado o facilitador del equipo Scrum, responsable de asegurar que el equipo siga las prácticas y valores de Scrum, eliminando impedimentos y facilitando la comunicación entre los miembros del equipo.

Figura 3

Insignia Scrum Master

Nota: De acuerdo con la definición de la figura 3 desde certmind y publicada en insignium.uk, es demostrar que la profesional cuenta buen dominio de la terminología, estructura y prácticas de

Scrum, además de contar con las habilidades para impulsar la adopción de Scrum en equipos de trabajo, lo cual, le asignan este tipo de insignias. Tomado de la página web insignium.uk (*Badge Scrum Master (Arnold Sebastian Beltran Puentes)*, n.d.).

A continuación, en la tabla 5, se relacionan las responsabilidades y limitaciones en cuanto a este rol:

Tabla 5

Responsabilidades y limitaciones Scrum Master

Responsabilidades	Limitaciones
Eliminación de Impedimentos: Resuelve barreras que impiden el progreso del equipo.	Falta de Autoridad Formal: No puede tomar decisiones técnicas o de negocio directamente.
Facilitación de Eventos Scrum: Organiza y modera eventos con eficiencia y enfoque (Sprint Planning, Daily Scrum, etc.).	Dependencia del Compromiso del Equipo: El éxito depende de la alineación del equipo con los principios ágiles.
Promotor de Prácticas Ágiles: Fomenta la autoorganización y la adaptabilidad del equipo y la organización.	Gestión de Conflictos: Limitado en mediar en conflictos graves sin el apoyo de la alta dirección.
Protección del Equipo: Defiende al equipo de interrupciones externas y distracciones que puedan desviar sus prioridades.	

Nota: Se hace una descripción de las responsabilidades y limitaciones en cuanto a Scrum Master.

Fuente: Autoría propia con base a certmind.com

Scrum Product Owner

Representa los intereses del cliente y de los stakeholders, gestionando el backlog del producto y asegurando que el equipo Scrum trabaje en las tareas con el mayor valor para el negocio.

Figura 4

Insignia Scrum Product Owner

Nota: De acuerdo con la definición de la insignia de la figura 4 desde certmind y publicada en insignium.uk, es demostrar que el profesional domina la estructura, consideraciones y prácticas para la adopción de Scrum, además de demostrar las habilidades en gestión de proyectos ágiles basados en Scrum, lo cual, le asignan este tipo de insignias. Tomado de la página web insignium.uk (*Badge Scrum Product Owner (Arnold Sebastian Beltran Puentes)*, n.d.).

En la tabla 6 se relacionan las responsabilidades y limitaciones en cuanto a este rol:

Tabla 6

Responsabilidades y limitaciones Scrum Product Owner

Responsabilidades	Limitaciones
Gestión del Product Backlog: Define y prioriza el backlog, asegurando que las	Conflicto de Prioridades: Dificultad para equilibrar las diversas prioridades de múltiples stakeholders.

funcionalidades más valiosas se aborden primero.	
Definición de Requerimientos: Traduce las necesidades del negocio en historias de usuario claras para el equipo de desarrollo.	Dependencia de Información: Requiere información clara y oportuna de los stakeholders; la falta de esta información puede afectar la priorización.
Toma de Decisiones Estratégicas: Alinea el producto con los objetivos estratégicos de la empresa.	Carga de Trabajo: La gestión del backlog y la satisfacción de todos los stakeholders puede resultar en una carga de trabajo significativa.
Comunicación con Stakeholders: Actúa como enlace principal entre stakeholders y el equipo de desarrollo, ajustando el backlog.	

Nota: Se hace una descripción de las responsabilidades y limitaciones en cuanto a Scrum Product Owner.

Fuente: Autoría propia con base a certmind.com

Scrum Developer

Un grupo multidisciplinario de profesionales que trabajan juntos para entregar incrementos de producto funcionales al final de cada sprint.

Figura 5

Insignia Scrum Developer

Nota: De acuerdo con la definición de la insignia de la figura 5 desde certmind y publicada en insignium.uk, es demostrar que el profesional tiene una comprensión práctica de los conceptos básicos, terminología, estructura y prácticas de Scrum, además de contar con las habilidades técnicas necesarias en un entorno de agilidad, lo cual, le asignan este tipo de insignias. Tomado de la página web insignium.uk (*Badge Scrum Developer (Arnold Sebastian Beltran Puentes)*, n.d.).

En la tabla 7 se relacionan las responsabilidades y limitaciones en cuanto a este rol:

Tabla 7

Responsabilidades y limitaciones Scrum Developer

Responsabilidades	Limitaciones
Autoorganización: El equipo decide cómo abordar las tareas y planifica su trabajo sin necesidad de dirección externa.	Autoorganización vs. Dirección: La falta de experiencia o disciplina puede dificultar la gestión efectiva del tiempo y tareas.
Multidisciplinaridad: El equipo cuenta con miembros que tienen diversas habilidades	Dependencia de Especialistas: Puede haber una dependencia excesiva de ciertos expertos,

necesarias para cubrir todos los aspectos del desarrollo. Entrega de incrementos: El equipo entrega incrementos funcionales al final de cada sprint, cumpliendo con la definición de hecho. Responsabilidad Compartida: Todos los miembros comparten la responsabilidad de alcanzar los objetivos y la calidad del producto.	lo que ralentiza el progreso si estos recursos son limitados. Variedad de Habilidades: La diversidad de habilidades puede dificultar la alineación del equipo hacia un enfoque común.

Nota: Se hace una descripción de las responsabilidades y limitaciones en cuanto a Scrum Developer

Fuente: Autoría propia con base a certmind.com

Diseño, comparación y pruebas de desempeño para seleccionar las herramientas a implementar

En este capítulo, se analiza la elección de la plataforma y lenguaje clave para el desarrollo del chatbot, con el objetivo de seleccionar la opción que ofrezca el mejor rendimiento y asegure su funcionamiento óptimo, aunque ya se tiene cono favorito Python para este fin. Si bien el objetivo principal es determinar la aplicación idónea, también es relevante explorar las diversas soluciones tecnológicas disponibles en el mercado. En este contexto se la importancia de seleccionar un lenguaje de programación adecuado para la implementación del chatbot y cómo influye en factores clave como el rendimiento, la escalabilidad y la facilidad de mantenimiento. Además, se realiza una comparativa entre estas aplicaciones, destacando sus ventajas, desventajas, curvas de rendimiento y simulaciones que ejemplifiquen su potencial.

Criterios de evaluación de lenguajes de programación

Uno de los aspectos clave a considerar al seleccionar cualquier lenguaje de programación para el desarrollo de chatbots es el rendimiento. Tal como sucede al adquirir un nuevo dispositivo, como una laptop, un automóvil o un teléfono celular, nuestras expectativas están centradas en que su rendimiento sea superior para ejecutar tareas cotidianas de manera eficiente. En este contexto, es fundamental analizar cuidadosamente cómo cada lenguaje responde a los desafíos de un chatbot, tales como la capacidad de gestionar múltiples consultas simultáneas, la rapidez en las respuestas y la estabilidad en su funcionamiento. En esta sección, se destacarán los principales factores que deben ser tomados en cuenta para asegurar que el lenguaje seleccionado optimice el rendimiento del chatbot y cumpla con los objetivos del proyecto:

Rendimiento

Este criterio evalúa la capacidad del lenguaje para ejecutar el chatbot de manera eficiente, especialmente en situaciones de alta demanda. Se mide en función del tiempo de respuesta, uso de recursos del sistema (CPU, memoria, hardware) y la capacidad de manejar múltiples solicitudes simultáneas. Un buen lenguaje garantiza tiempos de respuesta rápidos, incluso bajo carga, y optimizar el uso de los recursos del sistema.

Algunos puntos para destacar

Eficiencia en entornos de alta demanda: El chatbot diseñado en cualquier aplicación deberá ser capaz de responder múltiples consultas simultáneamente sin degradar su rendimiento, asegurando estabilidad y tiempos de respuesta óptimos en escenarios críticos.

Optimización del uso de recursos: Debe ser capaz de priorizar el bajo consumo de CPU y memoria, garantizando que el chatbot funcione correctamente sin afectar otros procesos del sistema.

Consideraciones

Tiempo de respuesta promedio a consultas de usuarios.

Eficiencia en el uso de memoria y CPU.

Capacidad de procesamiento de múltiples usuarios de manera concurrente.

Escalabilidad

La escalabilidad se refiere a la facilidad con la que el chatbot puede crecer y adaptarse a un mayor volumen de usuarios y nuevas funcionalidades sin comprometer el rendimiento. El lenguaje debe permitir la integración de nuevos módulos y la expansión a nuevos entornos sin requerir cambios drásticos en la arquitectura del sistema.

Algunos puntos para destacar

Capacidad de expansión modular: El chatbot con el código que se deba implementar debe permitir la integración de nuevas funcionalidades sin afectar su estructura principal, asegurando flexibilidad en futuras actualizaciones.

Rendimiento estable en entornos de alta demanda: Es necesario la priorización de soluciones que mantenga un uso eficiente de los recursos del sistema, garantizando tiempos de respuesta óptimos incluso con un número creciente de usuarios.

Consideraciones

Facilidad para añadir nuevas funcionalidades.

Rendimiento sostenido al incrementar el número de usuarios.

Compatibilidad con sistemas distribuidos y arquitecturas en la nube.

Facilidad de desarrollo

Este criterio analiza la curva de aprendizaje del lenguaje, la disponibilidad de herramientas de desarrollo, librerías y documentación que faciliten el trabajo de los desarrolladores. Un lenguaje que ofrezca una comunidad activa y recursos amplios permite un desarrollo más ágil y eficiente.

Algunos puntos para destacar

Amplia disponibilidad de frameworks y librerías: Debe contar con disponibilidad de lenguajes con bibliotecas especializadas en IA, PLN y redes neuronales (RNN), así como frameworks que agilicen el desarrollo para el chatbot.

Soporte comunitario y documentación accesible: La existencia de una comunidad activa y una documentación extensa facilita la resolución de errores, la optimización del código y la

incorporación de nuevas funcionalidades, son referencias claves para el desarrollo de la herramienta.

Consideraciones

Curva de aprendizaje del lenguaje.

Disponibilidad de frameworks y librerías especializadas (como frameworks de PLN, RNN, IA o personalizables).

Comunidad activa y cantidad de documentación accesible.

Compatibilidad e integración

La compatibilidad y la capacidad de integración con sistemas existentes es fundamental. El lenguaje debe ser flexible y permitir una integración fluida con bases de datos, APIs externas, servicios en la nube y sistemas empresariales. En este caso, se evaluará especialmente la integración con servicios de Oracle y bases de datos SQL.

Algunos puntos para destacar

Interoperabilidad con APIs empresariales: El programa que se seleccione debe ser capaz de conectarse a APIs compatibles de la empresa, que permita la interacción con plataformas como APEX de Oracle, sistemas de gestión interna y herramientas de monitoreo en Claro.

Conexión eficiente con bases de datos: Se debe garantizar la integración con Oracle en Venecia y Triara, asegurando consultas optimizadas, acceso seguro y gestión eficiente de la información.

Escalabilidad y compatibilidad en la nube OCI: El chatbot con el código seleccionado debe estar diseñado para operar en entornos locales y en la nube, facilitando la integración con servicios empresariales sin afectar el rendimiento del sistema y compatibilidad.

Consideraciones

Facilidad de integración con APIs y servicios de terceros.

Compatibilidad con bases de datos como Oracle de Claro ubicados en Venecia y Triara.

Soporte para la comunicación con sistemas empresariales y servicios en la nube.

Seguridad

La seguridad es fundamental y obligatorio en el desarrollo del chatbot, especialmente cuando maneja datos sensibles o interactúa con sistemas empresariales. El lenguaje debe ofrecer herramientas y prácticas que faciliten la implementación de medidas de seguridad robustas, como el cifrado, la autenticación y la protección contra ataques.

Algunos puntos para destacar

Seguridad con SSL/TLS: Todas las comunicaciones entre el chatbot y los usuarios, así como las interacciones con las bases de datos, deben estar protegidas mediante cifrado SSL/TLS, asegurando la confidencialidad e integridad de los datos.

Autenticación y Autorización: Se deben implementar mecanismos robustos de autenticación y autorización para controlar el acceso a las funcionalidades del chatbot y proteger la información sensible de los usuarios como acceso por VPN autorizado, enlace mediante Apex de Oracle y captcha de seguridad.

Consideraciones

Disponibilidad de librerías de seguridad y cifrado (SSL/TLS).

Facilidad de implementación de autenticación y control de acceso.

Prevención de vulnerabilidades comunes (como inyecciones de código o ataques DoS).

Mantenimiento y soporte

El mantenimiento continuo es crucial para asegurar la estabilidad y la evolución del chatbot a lo largo del tiempo. Un lenguaje que facilite la depuración, la actualización de código y la implementación de parches es esencial para garantizar que el sistema pueda mantenerse y escalar sin complicaciones.

Algunos puntos para destacar

Facilidad de actualización y escalabilidad: El código que se seleccione se debe priorizar los lenguajes y arquitecturas que permitan actualizaciones rápidas y eficientes, sin interrumpir el funcionamiento del chatbot.

Disponibilidad de herramientas de depuración: Contar con entornos de desarrollo de fácil comprensión que permita identificar y corregir errores de manera ágil garantizará la estabilidad del sistema a largo plazo.

Consideraciones

Facilidad para realizar mantenimiento y actualizaciones.

Disponibilidad de herramientas de depuración.

Coste y complejidad de mantener el código a largo plazo.

Comparación entre lenguajes de programación para chatbots

A continuación, se resume una tabla comparativa entre los lenguajes de programación más utilizados para el desarrollo de chatbots, evaluados según diferentes criterios clave.

Tabla 8

Resumen comparativa lenguajes de desarrollo para chatbots

Lenguaje	Ventajas	Desventajas	Casos de uso

Python	Fácil de aprender y desarrollar. Gran soporte para inteligencia artificial (IA) y procesamiento de lenguaje natural (PLN) mediante bibliotecas como NLTK, TensorFlow y Keras. Amplia comunidad y soporte.	Menor rendimiento en aplicaciones con alta concurrencia. No es el más eficiente para tareas en tiempo real.	Chatbots inteligentes con IA y PLN. Soluciones de análisis de datos y aprendizaje automático.
JavaScript (Node.js)	Ideal para aplicaciones en tiempo real. Gran rendimiento con múltiples usuarios concurrentes. Uso tanto en el backend como en el frontend.	Menos enfoque en IA y PLN que Python. Puede requerir más código para lograr ciertas tareas comparado con otros lenguajes.	Chatbots integrados en sitios web o plataformas web. Chatbots en aplicaciones de tiempo real como Slack o Facebook Messenger.
Java	Excelente escalabilidad y rendimiento. Muy utilizado en entornos empresariales.	Curva de aprendizaje más pronunciada. Mayor complejidad y verbosidad en el código.	Chatbots en aplicaciones empresariales grandes.

	Gran estabilidad y seguridad.		Soluciones que requieren robustez y alta concurrencia.
C# (C-Sharp)	Gran integración con el ecosistema Microsoft (Azure, .NET). Buen rendimiento en aplicaciones empresariales.	Menor soporte para IA/PLN nativo en comparación con Python. Principalmente usado en entornos de Microsoft.	Chatbots corporativos que necesitan integración con plataformas Microsoft. Aplicaciones empresariales de atención al cliente.
C++	Control total sobre los recursos y excelente rendimiento. Ideal para aplicaciones de rendimiento crítico.	Mayor complejidad en el desarrollo. Menos eficiente para el desarrollo rápido y dinámico como en Python o JavaScript.	Aplicaciones de alto rendimiento, procesamiento intensivo y simulaciones complejas.
MATLAB (UNAD)	Potente en cálculos numéricos y análisis de datos. Toolboxes avanzadas para IA y PLN.	No diseñado para chatbots de producción. Licencia costosa. Rendimiento limitado en tiempo real.	Prototipado y simulaciones de algoritmos de IA. Análisis de datos complejos y pruebas de conceptos.

Ruby	Sintaxis sencilla y fácil de leer. Desarrollo rápido de prototipos con Ruby on Rails.	Menor rendimiento y escalabilidad en comparación con otros lenguajes como Java o Node.js.	Desarrollo rápido de chatbots para aplicaciones web. Proyectos con requerimientos de tiempo limitado.

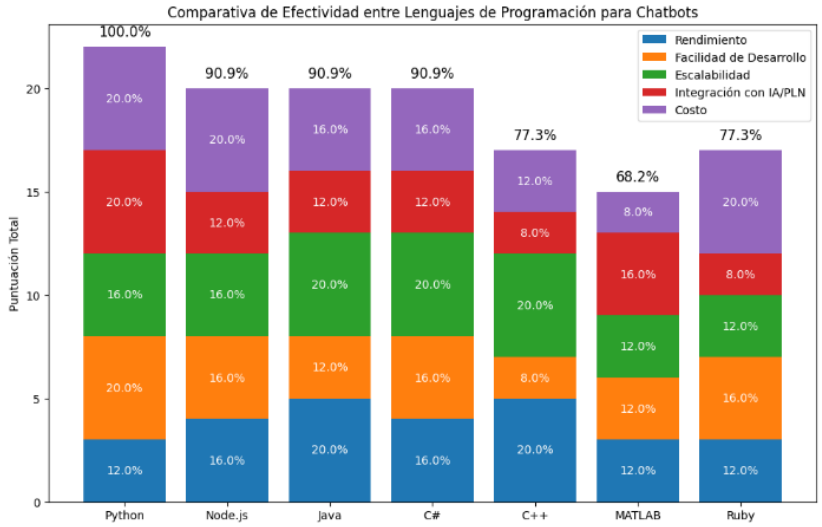
Nota: Se relaciona la comparativa de aplicaciones disponibles en el mercado para la creación del chatbot.

Fuente: Autoría propia

A continuación, se muestra en la figura 6 un análisis realizado de puntuación en Python asignado a modo personal. Esta información es generada a partir de datos configurados en Python utilizando la librería Matplotlib. Dado que esta comparativa fue elaborada bajo el criterio del autor de esta tesis, basada en su investigación, no proviene de una fuente externa. La evaluación se fundamenta en puntuaciones hipotéticas asignadas a los lenguajes de programación más utilizados en el desarrollo de chatbots, con una escala de 1 a 5, donde 1 es la puntuación mínima y 5 la máxima. Tras realizar los cálculos correspondientes en la aplicación de Python, se observa que Python emerge como una de las mejores opciones para el desarrollo de chatbots convencionales, gracias a su versatilidad y amplia adopción. Por otro lado, aunque MATLAB ofrece capacidades avanzadas, como el procesamiento de imágenes y la inteligencia artificial, es menos común y adecuado para el desarrollo directo de chatbots.

Figura 6

Gráfica comparativa entre lenguajes de programación para chatbots



Nota: Gráfica ilustrativa comparativa entre los lenguajes de programación para la elaboración de un chatbot con las funciones compatibles y requeridas por negocio a criterio del autor con base a su investigación.

Fuente: Autoría propia.

Desarrollo de la aplicación con IA – sistema de chatbot

En este capítulo se describe el proceso de desarrollo del sistema de chatbot basado en inteligencia artificial, destacando las tecnologías y herramientas clave empleadas. El objetivo principal es detallar cómo cada componente fue implementado para cumplir con las necesidades del proyecto, optimizando la interacción con los usuarios y automatizando procesos clave dentro del proyecto Oracle de Itel. A lo largo del capítulo, se explorarán las funcionalidades principales del chatbot, desde la gestión de consultas frecuentes y la creación de requerimientos (REQ), hasta la integración con bases de datos, sistemas de seguridad y envío de alertas y notificaciones y estado de los servicios en modo monitoreo para la toma de decisiones, mostrando cómo el chatbot permite mejorar la eficiencia operativa y la experiencia del usuario. Se recuerda que este desarrollo se basará en el código Python. Algunas partes se requiere de intervención investigativa como libros de desarrollo basadas en este lenguaje, fuentes publicadas por internet y publicaciones de la universidad.

Arquitectura del sistema

El diseño arquitectónico del chatbot se basa en una estructura modular que permite una clara separación entre sus componentes principales, facilitando el mantenimiento y la escalabilidad del sistema. La arquitectura del chatbot se divide en tres capas fundamentales: backend, frontend, y la integración con la inteligencia artificial y procesamiento de lenguaje natural (IA/PLN).

Backend

El backend del chatbot se encarga de gestionar la mayoría de la lógica del negocio, el procesamiento de solicitudes de los usuarios y la integración con otras herramientas de la empresa. Son procesos que no verá el usuario, ya que interactuará desde un servidor los servicios

automatizados. Está desarrollado en Python, aprovechando su flexibilidad y robustez. Además, el backend se comunica con las bases de datos ubicado en Claro Soluciones de Venecia y Triara, así como la herramienta de MyIT obteniendo información relevante de los REQ en tiempo real para responder a las consultas más frecuentes. También tienen la posibilidad de enviar las alertas y notificaciones en cuanto al estado de los servicios y monitoreos a través del canal de WhatsApp.

Frontend

La interfaz del chatbot está desarrollada utilizando Flask como principal. El frontend proporciona una experiencia de usuario enriquecida, permitiendo la interacción con el chatbot a través de mensajes de texto en una interfaz web html5. Se ha diseñado pensando en la accesibilidad más amigable, incorporando características como soporte para leer texto de la pantalla y modos claro y oscuro, entre otras funciones.

Integración con IA/PLN

El motor de inteligencia artificial del chatbot utiliza librerías como NLTK y TensorFlow para el procesamiento de lenguaje natural (PLN), lo que permite al sistema entender y generar respuestas en lenguaje natural. Esta integración permite al chatbot interpretar correctamente las consultas de los usuarios y generar respuestas lo más cercanamente precisas, mejorando la experiencia de interacción. Además, el chatbot cuenta con capacidades de aprendizaje automático para mejorar sus respuestas basadas en el feedback recibido con el modelo avanzado o indicando aprendizaje manual.

Bases de datos

El chatbot está integrado con las bases de datos corporativas Venecia y Triara de la empresa Claro Soluciones, que almacenan toda la información necesaria para responder a las consultas de los usuarios y técnicos en cuanto a consulta de paquetes, promociones, planes y

servicios prepagos disponibles. La comunicación con las bases de datos se realiza de forma segura mediante consultas SQL, asegurando la integridad de los datos.

Herramientas y tecnologías utilizadas

A continuación, se describe en la tabla 9 que describe las herramientas implementadas en el desarrollo del chatbot, el cual ya ha sido puesto en producción y continúa en fase de entrenamiento y ajuste para proporcionar respuestas personalizadas según las necesidades del negocio de Oracle. Esta aplicación se ejecuta localmente, lo que garantiza la seguridad y confidencialidad de la información, asegurando que todos los datos y consultas sean procesados de manera eficiente y segura dentro del entorno controlado por la empresa.

Tabla 9

Herramientas utilizadas para el desarrollo del chatbot

Componente	Descripción	Detalles Técnicos
Lenguaje de Programación	Python se utiliza como el lenguaje base debido a su versatilidad, facilidad de integración y amplia adopción.	Versión: Python 3.x. Herramientas utilizadas: Anaconda para la gestión de entornos y dependencias.
Frameworks de IA/PLN	NLTK y TensorFlow son utilizados para implementar el procesamiento de lenguaje natural (PLN).	NLTK: Manejo de procesamiento básico de texto (tokenización, lematización). TensorFlow: Modelos de IA para PLN más avanzados.

Bases de Datos	Integración con las bases de datos Venecia y Triara para almacenar y recuperar las consultas y respuestas del chatbot.	SQL como lenguaje de consulta. Seguridad mediante autenticación y encriptación SSL.
Desarrollo Web	Flask se utiliza para crear la interfaz del chatbot, permitiendo una interacción ligera y eficiente a través de la web.	Framework web ligero de Python. Soporte para REST APIs y conexiones rápidas.
API's o Servicios Externos	Integración con API's externas para obtener datos adicionales o realizar consultas complejas que no maneja el chatbot.	API's de servicios en la nube para consultas externas y mejoras en el procesamiento. Generación y consulta de requerimientos (REQ) con la herramienta de MyIT. Uso de JSON para el intercambio de datos. Módulos para funcionabilidad en servicios de telecomunicaciones básicas asociados a los manejados de WebLogic. Notificaciones a través de canales, grupos o usuarios de WhatsApp.

Nota: Tabla con información detallada acerca de las herramientas utilizadas para el desarrollo del chatbot requeridas por la línea de negocio.

Fuente: Autoría propia.

Flujo de interacción del usuario

Este apartado describe de algunos flujos de interacción que sigue un usuario al interactuar con el chatbot, desde el momento en que inicia la consulta hasta el cierre de la sesión. El flujo está diseñado para ser intuitivo y eficiente, permitiendo al usuario obtener respuestas a consultas comunes, así como generar tickets de soporte cuando sea necesario, monitoreo de servicios del aprovisionador, base de datos, estado en servidores, entre otros servicios base. Se describe algunos flujos del servicio ya implementados en el chatbot y que funcionan a la fecha de hoy en el proyecto.

Descripción del flujo de interacción

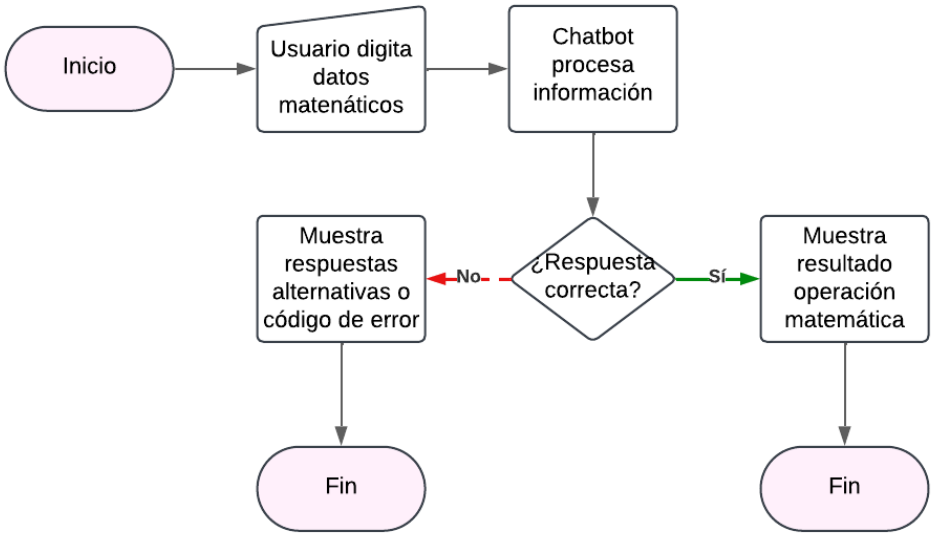
A continuación, se plasman mediante flujogramas las importantes interacciones en cuanto a los servicios que presta el chatbot a la fecha. Cada figura cuenta con su propia descripción del proceso más adelante para su comprensión.

Procesos matemáticos

El chatbot puede procesar operaciones matemáticas básicas desde sumas, restas, multiplicación, división y combinaciones entre estos signos matemáticos. De acuerdo con la figura 7, el usuario puede interactuar directamente con el chatbot para realizar operaciones matemáticas.

Figura 7

Diagrama de flujo proceso matemático



Fuente: Autoría propia

Descripción del flujograma matemáticas

A continuación, se describe cada uno de los elementos del flujograma de la figura 7.

Inicio: Representa el punto de inicio del flujo.

Usuario digita datos matemáticos: El usuario ingresa los datos o la operación matemática que desea resolver.

Chatbot procesa información: El chatbot toma los datos proporcionados por el usuario y realiza el procesamiento para interpretar la operación matemática.

¿Respuesta correcta?: Aquí se verifica si el resultado calculado por el chatbot es correcto o si hay algún error en la interpretación o procesamiento.

Si la respuesta es correcta (camino verde): El chatbot muestra el resultado de la operación matemática y el flujo finaliza.

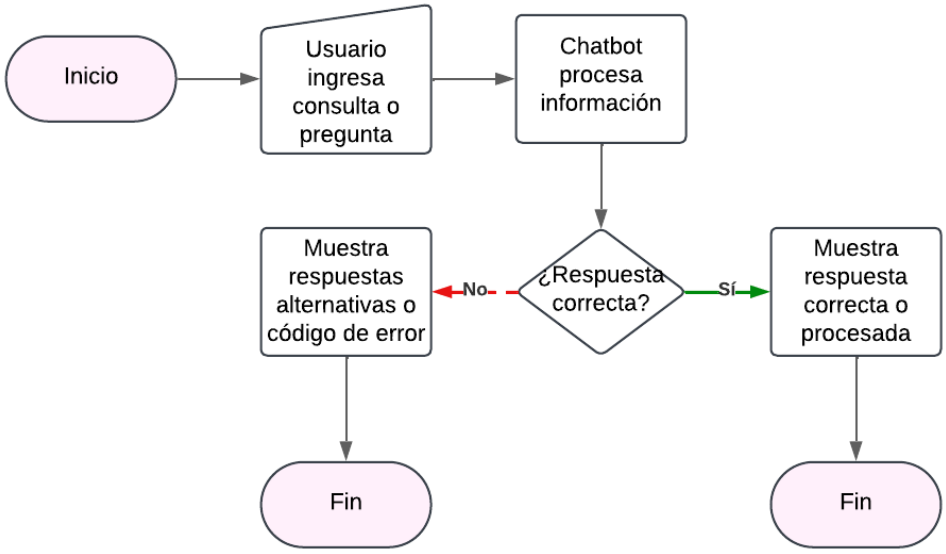
Si la respuesta no es correcta (camino rojo): El chatbot ofrece respuestas alternativas o muestra un código de error y el flujo finaliza.

Consulta de preguntas frecuentes

Es una de las funciones generales más importantes que cuenta el modelo chatbot básico. Esto incluye, respuestas acertadas, opciones en caso de no hallar una respuesta directa y referencias de internet en caso de aplicar. En la figura 8, se muestra un breve proceso de cómo puede interactuar el usuario con el modelo chatbot con preguntas simples o frecuentes.

Figura 8

Diagrama de flujo consultas frecuentes



Nota: Ilustración tipo flujograma realizado a partir de aplicación web Ludichart. La descripción se encuentra más adelante.

Fuente: Autoría propia

Descripción del flujograma preguntas frecuentes

A continuación, se describe cada uno de los elementos del flujograma de la figura 8.

Inicio: Punto de partida del flujo.

Usuario ingresa consulta o pregunta: El usuario introduce una pregunta o solicitud al sistema del chatbot.

Chatbot procesa información: El chatbot analiza la consulta ingresada por el usuario y busca generar una respuesta.

¿Respuesta correcta?: Este es un punto de decisión donde se evalúa si el chatbot puede proporcionar una respuesta adecuada o si surge algún error.

Si la respuesta es correcta (camino verde): El chatbot muestra la respuesta correcta o procesada, y el flujo termina.

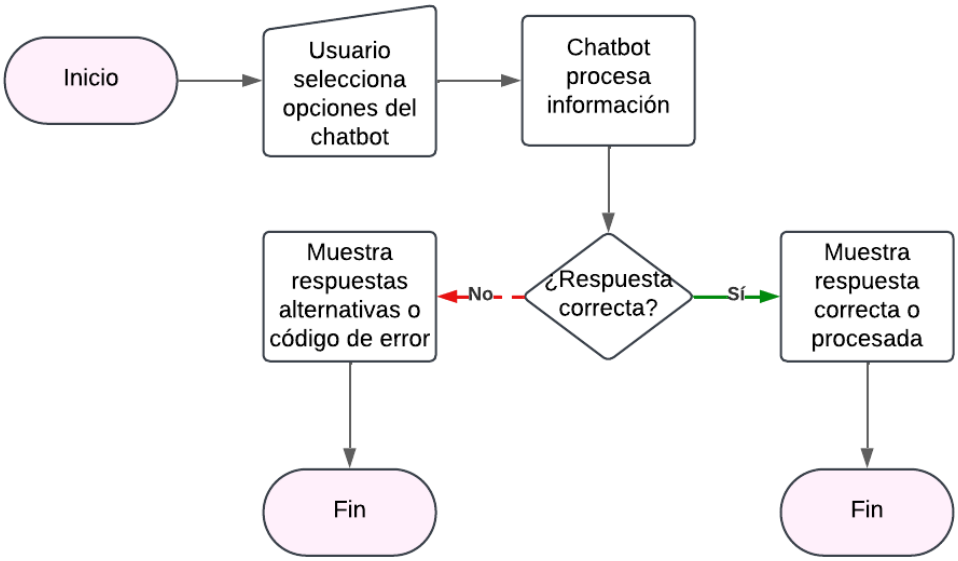
Si la respuesta no es correcta (camino rojo): El chatbot muestra respuestas alternativas o un código de error, y el flujo finaliza.

Selección de opciones y sugerencias

Son opciones de interacción que muestra el chatbot en cada conversación en el cual el usuario selecciona opciones predefinidas para recibir una respuesta adicional. En la figura 9, se evidencia un ejemplo de cómo se puede activar este beneficio para la continuidad de su interacción.

Figura 9

Diagrama de flujo de opciones y sugerencias



Nota: Ilustración tipo flujograma realizado a partir de aplicación web Ludichart. La descripción se encuentra más adelante.

Fuente: Autoría propia

Descripción del flujograma opciones y sugerencias

A continuación, se describe cada uno de los elementos del flujograma de la figura 9.

Inicio: Representa el punto de partida del flujo.

Usuario selecciona opciones del chatbot: El usuario interactúa con el chatbot eligiendo entre las opciones disponibles, como menús, comandos preconfigurados o respuestas con las opciones sugeridas.

Chatbot procesa información: El chatbot analiza la opción seleccionada por el usuario y realiza el procesamiento necesario para generar una respuesta.

¿Respuesta correcta?: Se evalúa si el chatbot puede proporcionar una respuesta adecuada basada en la selección del usuario.

Si la respuesta es correcta (camino verde): El chatbot muestra la respuesta correcta o procesada, y el flujo finaliza.

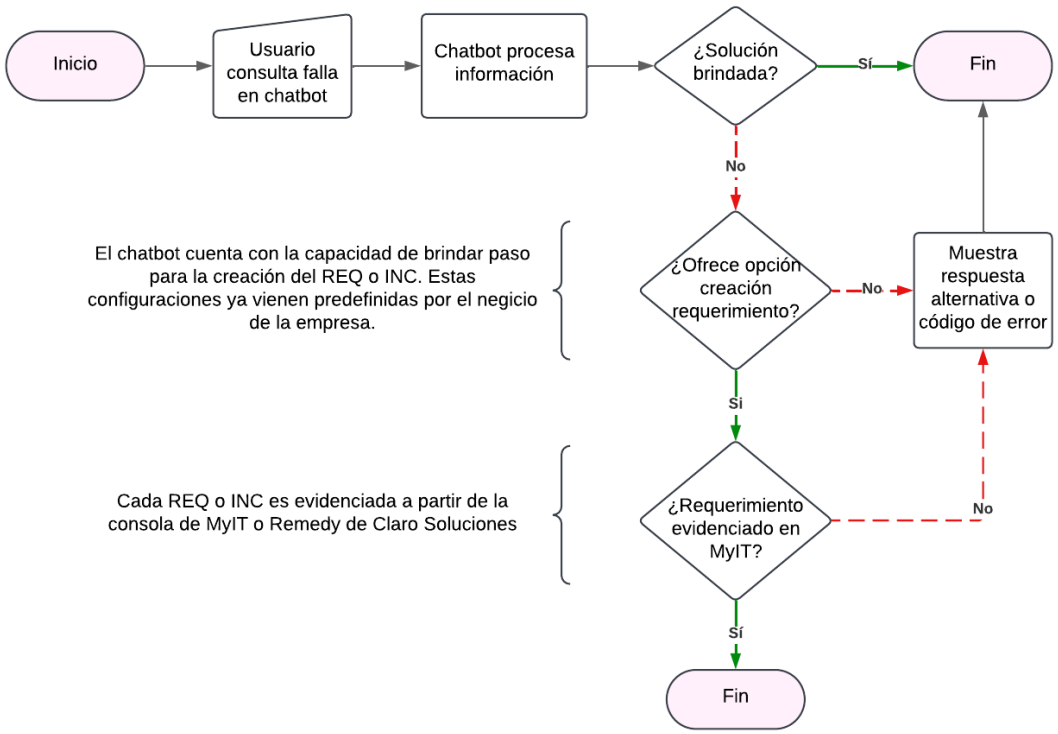
Si la respuesta no es correcta (camino rojo): El chatbot presenta respuestas alternativas o un código de error, y el flujo termina.

Creación de REQ. en caso de una falla

Es un valor agregado que alberga un proceso interno en cuanto a la creación del REQ. interviniendo directamente en la herramienta de MyIT en caso de que el usuario consulte una falla y no logre resolver. En la figura 10, se muestra un ejemplo de cómo el usuario puede interactuar con el chatbot para la creación del REQ en MyIT.

Figura 10

Diagrama de flujo creación de REQ



Nota: Ilustración tipo flujograma realizado a partir de aplicación web Ludichart. La descripción se encuentra más adelante.

Fuente: Autoría propia

Descripción del flujograma para la creación de REQ

A continuación, se describe cada uno de los elementos del flujograma de la figura 10.

Inicio: Marca el comienzo del flujo.

Usuario consulta falla en chatbot: El usuario reporta o consulta sobre un problema o falla que está experimentando con el chatbot.

Chatbot procesa información: El chatbot analiza la consulta y evalúa si puede proporcionar una solución.

¿Solución brindada?: Se verifica si el chatbot puede resolver el problema reportado.

Si la solución es brindada (camino verde): El flujo termina exitosamente.

Si no se brinda una solución (camino rojo): El flujo continúa hacia la siguiente etapa.

¿Ofrece opción creación requerimiento?: Se determina si el chatbot puede sugerir o iniciar un proceso de creación de requerimientos para resolver el problema.

Si se ofrece la opción (camino verde): El proceso avanza al siguiente paso.

Si no se ofrece (camino rojo): El chatbot muestra respuestas alternativas o un código de error, y el flujo termina.

¿Requerimiento evidenciado en MyIT?: Se valida si el requerimiento fue creado o evidenciado en la herramienta MyIT.

Si el requerimiento se crea correctamente (camino verde): El flujo termina satisfactoriamente.

Si no está evidenciado (camino rojo): El chatbot muestra un mensaje de error o información adicional, y el flujo termina.

Fin: El proceso concluye, ya sea con la resolución del problema, la creación de un requerimiento, o la indicación de un error.

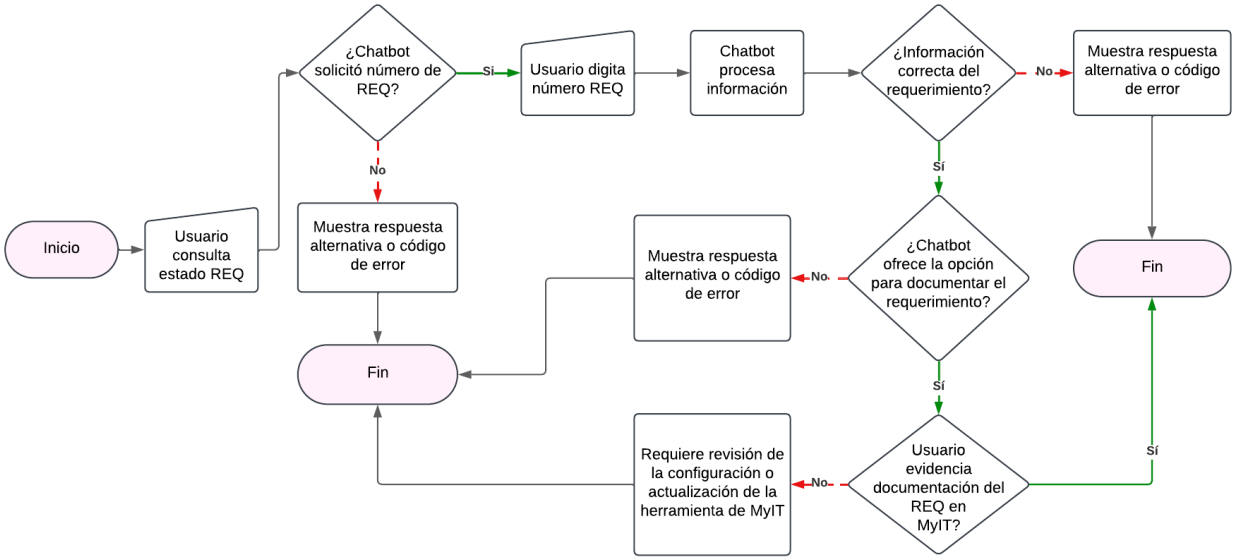
Consultas de REQ

Este proceso consulta el estado de REQ desde la interfaz del chatbot y solamente deben ser con requerimientos creados a partir de la misma cuenta del usuario, dado de que no se cuenta con la base de datos directamente desde MyIT. Chatbot no es compatible con Remedy. Se

muestra un ejemplo de cómo el usuario puede interactuar con el chatbot de acuerdo con el flujograma de la figura 11.

Figura 11

Diagrama de flujo consulta de REQ



Nota: Ilustración tipo flujograma realizado a partir de aplicación web Ludichart. La descripción se encuentra más adelante.

Fuente: Autoría propia.

Descripción del flujograma para las consultas de REQ

A continuación, se describe cada uno de los elementos del flujograma de la figura 11.

Inicio: Marca el inicio del proceso.

Usuario consulta estado REQ: El usuario solicita información sobre el estado de un requerimiento específico.

¿Chatbot solicitó número de REQ?: Se verifica si el chatbot pidió al usuario ingresar el número del requerimiento.

Sí (camino verde): El usuario procede a ingresar el número del requerimiento.

No (camino rojo): El chatbot muestra una respuesta alternativa o un código de error, y el flujo termina.

Usuario digita número REQ: El usuario ingresa el número del requerimiento solicitado.

Chatbot procesa información: El chatbot analiza el número ingresado y busca el estado del requerimiento en su base de datos.

¿Información correcta del requerimiento?: Se evalúa si la información obtenida es válida.

Sí (camino verde): El chatbot muestra la información correcta y el flujo termina.

No (camino rojo): El chatbot muestra una respuesta alternativa o un código de error, o bien pasa al siguiente paso.

¿Chatbot ofrece la opción para documentar el requerimiento?: Se evalúa si el chatbot brinda la posibilidad de registrar información adicional sobre el requerimiento.

Sí (camino verde): El flujo pasa a verificar si el usuario evidencia la documentación en MyIT.

No (camino rojo): El chatbot solicita revisión de la configuración o actualización de la herramienta MyIT, y el flujo termina.

¿Usuario evidencia documentación del REQ en MyIT?: Se verifica si el usuario confirma la documentación del requerimiento en la herramienta MyIT.

Sí (camino verde): El proceso finaliza exitosamente.

No (camino rojo): Se muestra un mensaje indicando la necesidad de revisión o actualización de MyIT, y el flujo concluye.

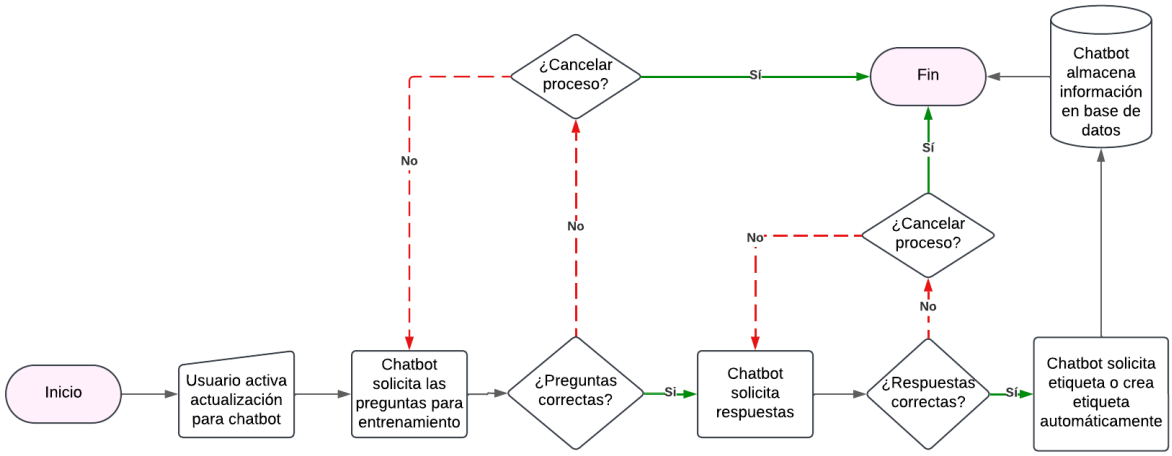
Fin: El flujo termina en varios puntos dependiendo del éxito o fallo en los pasos anteriores.

Actualización y entrenamiento del modelo chatbot

El chatbot cuenta con opción de entrenamiento directamente desde la interfaz para la no intervención del código en cuanto a carga de preguntas y respuestas frecuentes, al menos de que un profesional encargado conozca mejor la variable del código. En la figura 12, se muestra cómo puede el usuario interactuar con el chatbot para iniciar su entrenamiento una vez cargado su base de datos para ese fin.

Figura 12

Diagrama de flujo actualización y entrenamiento del modelo



Nota: Ilustración tipo flujograma realizado a partir de aplicación web Ludichart. La descripción se encuentra más adelante.

Fuente: Autoría propia.

Descripción del flujograma para el entrenamiento del modelo

A continuación, se describe cada uno de los elementos del flujograma de la figura 12.

Inicio: Representa el inicio del flujo.

Usuario activa actualización para chatbot: El usuario inicia el proceso de actualización y entrenamiento del chatbot.

Chatbot solicita las preguntas para entrenamiento: El chatbot pide al usuario las preguntas necesarias para el entrenamiento.

¿Cancelar proceso?: Se evalúa si el usuario desea cancelar el proceso.

Sí (camino verde): El flujo termina.

No (camino rojo): Continúa el proceso de validación.

¿Preguntas correctas?: El chatbot valida si las preguntas proporcionadas son correctas.

Sí (camino verde): Se pasa a la siguiente etapa.

No (camino rojo): Solicita corrección de preguntas o permite cancelar el proceso.

Chatbot solicita respuestas: El chatbot pide las respuestas correspondientes a las preguntas.

¿Cancelar proceso?: Se verifica nuevamente si el usuario desea cancelar.

Sí (camino verde): El flujo termina.

No (camino rojo): El proceso continúa.

¿Respuestas correctas?: Se valida si las respuestas proporcionadas son correctas.

Sí (camino verde): El proceso pasa a la etapa de etiquetado.

No (camino rojo): Solicita corrección de respuestas o permite cancelar.

Chatbot solicita etiqueta o crea etiqueta automáticamente: El chatbot solicita que se asignen etiquetas a las preguntas y respuestas o genera las etiquetas automáticamente a elección del usuario si es necesario.

Chatbot almacena información en base de datos: Finalmente, el chatbot guarda las preguntas, respuestas y etiquetas en la base de datos para su uso futuro.

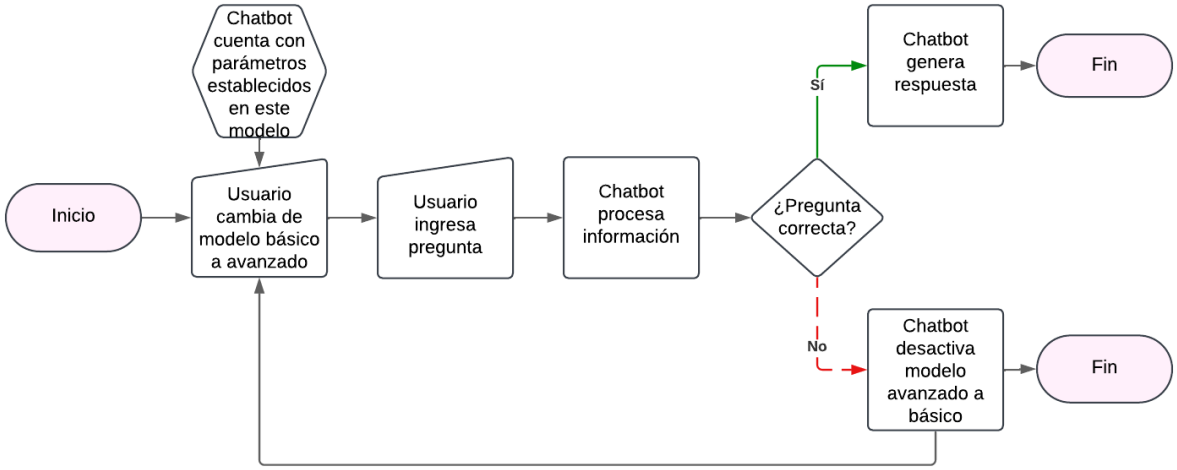
Fin: El flujo concluye dependiendo de las situaciones dentro del escenario.

Selección modelo avanzado

Este modelo mediante GPT, enriquece la experiencia del usuario. Solo debe reconocer ciertos patrones que son prestablecidos de acuerdo con las políticas y negocio establecido por Oracle para el proyecto Itel. A continuación, en la figura 13 se muestra el procedimiento ya definido para este modelo. Este solo se utilizará para casos de alta importancia como reconocimiento de errores avanzados y solvencia de problemas para negocio.

Figura 13

Diagrama de flujo para la selección del modelo avanzado o IA



Nota: Ilustración tipo flujograma realizado a partir de aplicación web Ludichart. La descripción se encuentra más adelante.

Fuente: Autoría propia.

Descripción del flujograma en cuanto a la selección del modelo avanzado

A continuación, se describe cada uno de los elementos del flujograma de la figura 13.

Inicio: Representa el punto de inicio del flujo.

Chatbot cuenta con parámetros establecidos en este modelo: Indica que el chatbot tiene configuraciones predefinidas para sus modelos (básico y avanzado).

Usuario cambia de modelo básico a avanzado: El usuario elige activar el modelo avanzado del chatbot.

Usuario ingresa pregunta: El usuario introduce una consulta o pregunta que el chatbot debe procesar.

Chatbot procesa información: El chatbot analiza la pregunta utilizando los parámetros del modelo avanzado.

¿Pregunta correcta?: Se valida si la pregunta es compatible con el modelo avanzado.

Sí (camino verde): El chatbot genera una respuesta, y el flujo termina exitosamente.

No (camino rojo): El chatbot desactiva el modelo avanzado y regresa al modelo básico, concluyendo el flujo.

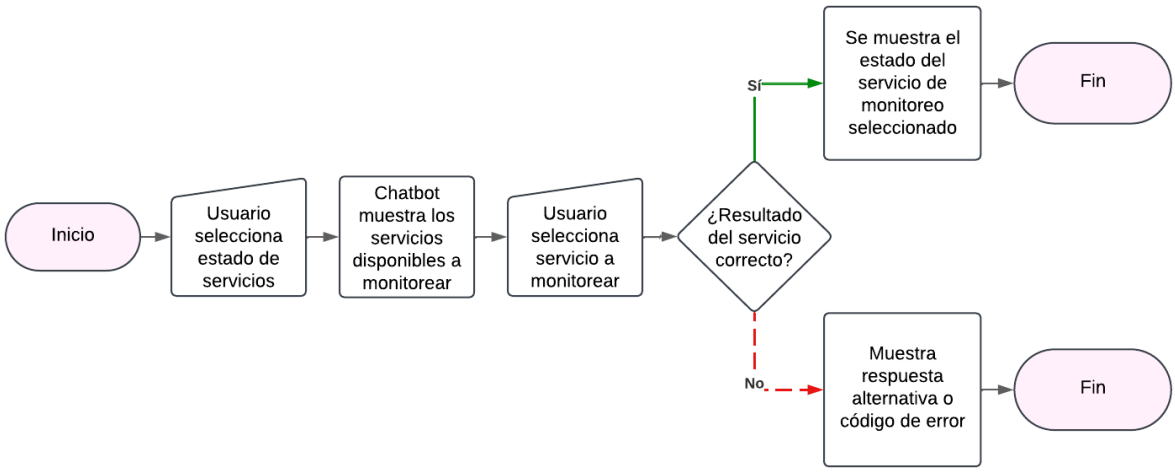
Fin: Representa el cierre del proceso en ambos escenarios.

Monitoreo sistemas de telecomunicaciones

Otro de los valores agregados que cuenta el chatbot es el monitoreo de los servicios de telecomunicaciones. Utiliza módulos personalizados para la intervención de las bases de datos de Venecia y Triara. Adicional, se ha creado otros módulos que monitorean los servidores, incluyendo, servicios de ping, telnet, estado de consumo de memoria RAM, entre otros aspectos para la detección de fallas y creación de tickets REQ. cuando sea necesario y explicado en la figura 14.

Figura 14

Diagrama de flujo monitoreo de servicios de telecomunicaciones



Fuente: Autoría propia.

Descripción del flujograma del flujo de monitoreo de sistemas de telecomunicaciones

A continuación, se describe cada uno de los elementos del flujograma de la figura 14.

Inicio: Representa el inicio del proceso.

Usuario selecciona estado de servicios: El usuario elige consultar el estado de los servicios como servidores y bases de datos.

Chatbot muestra los servicios disponibles a monitorear: El chatbot presenta una lista de servicios que se pueden monitorear.

Usuario selecciona servicio a monitorear: El usuario escoge un servicio específico de la lista para verificar su estado.

¿Resultado del servicio correcto?: Se valida si el resultado del monitoreo del servicio seleccionado es adecuado o no.

Sí (camino verde): Se muestra el estado del servicio monitoreado, y el flujo concluye exitosamente.

No (camino rojo): Se presenta una respuesta alternativa o un código de error, y el flujo termina.

Fin: Marca el final del proceso en ambos escenarios.

Los diagramas de flujo fueron elaborados a partir de la aplicación de Ludichart. (*Lucidchart | Diagramas Creados Con Inteligencia*, n.d.)

Implementación y despliegue

Este apartado describe el proceso de implementación del chatbot en el entorno de producción en la sede Plaza Claro para el proyecto Itel de Oracle, incluyendo las herramientas utilizadas, la configuración del entorno y las prácticas de mantenimiento que asegurarán su funcionamiento óptimo a largo plazo.

Proceso de implementación en el equipo cliente

La implementación del chatbot en el equipo dedicado de la empresa sigue un enfoque estructurado para garantizar su estabilidad y seguridad mientras sea mejorada su versión para el cargue a servidores OCI de Oracle. A continuación, se detallan los pasos principales:

Preparación del entorno de desarrollo y pruebas

Antes de su implementación en producción oficial, se crea entornos virtuales en equipos locales como en el equipo cliente para realizar pruebas exhaustivas. Estos entornos de prueba permiten verificar que el chatbot funcione correctamente en un entorno controlado y simule las condiciones de producción.

Configuración del equipo servidor de producción

El equipo dedicado para el modelo chatbot es el Dell Inspiron 14-3467, máquina suficientemente capaz de suplir las necesidades del modelo del chatbot, cumpliendo con las expecificaciones técnicas mínimas permitidas por el modelo. Esto incluyte la instalación de módulos y dependencias esenciales como Python principalmente, paquetes ofimáticos como LibreOffice para hojas de cálculo (este viene prestablecido en sistemas Linux) y Nginx para la gestión de solicitudes HTTP, lo cual facilita su portabilidad y escalabilidad.

El entorno de producción está optimizado para gestionar múltiples consultas simultáneas, asegurando que el chatbot pueda responder a los usuarios sin tiempos de espera prolongados.

Configuración del entorno de producción

El entorno de producción se configura para maximizar la eficiencia y seguridad del chatbot. Las configuraciones clave incluyen:

Gestión de recursos: Se ha asignado los recursos necesarios en el servidor para garantizar que el chatbot funcione sin interrupciones, asignando suficiente memoria RAM y potencia de procesamiento.

Seguridad: Se ha implementado medidas de seguridad como SSL/TLS para la protección de los datos transmitidos y la autenticación VPN para el acceso administrativo, asegurando que solo personal autorizado pueda realizar cambios en el sistema.

Monitoreo y registro de actividades: Se han empleado herramientas de monitoreo básicos los logs de procesos y monitoreo del sistema como htop para rastrear el rendimiento del chatbot en tiempo real, identificando posibles problemas y ajustando el rendimiento según sea necesario. Además, se mantiene un registro de todas las interacciones y actividades para análisis y auditorías futuras mediane logs de servicios.

Mantenimiento del sistema y actualizaciones futuras

Es fundamental el mantenimiento del modelo para asegurar su longevidad y capacidad de adaptarse a los cambios en el negocio y la tecnología:

Mantenimiento regular: Se programan revisiones periódicas para limpiar registros antiguos, verificar la integridad de la base de datos y realizar copias de seguridad. Estas tareas de mantenimiento permiten que el chatbot funcione de manera fluida y sin interrupciones.

Actualización de modelos de IA: A medida que el modelo de IA del chatbot se entrena con datos nuevos, se implementan actualizaciones en el sistema para mejorar su precisión y adaptabilidad. Esto incluye la actualización de las librerías de procesamiento de lenguaje natural (PLN) y el reentrenamiento de modelos según sea necesario.

Implementación de nuevas funcionalidades: El sistema está diseñado para ser modular, lo que permite agregar nuevas funcionalidades de manera rápida y sencilla sin afectar el funcionamiento actual. Por ejemplo, se pueden integrar nuevos servicios para WhatsApp como procesos de notificaciones o herramientas de análisis de datos y monitoreo mediante gráficos para una mejor toma de decisiones.

Soporte y documentación: Todo el proceso de implementación y mantenimiento está documentado para que el equipo de TI de la empresa pueda gestionar el chatbot de manera autónoma. Se proporciona soporte para la solución de problemas y actualizaciones de acuerdo con las necesidades de la empresa. Además, estas pueden ser consultadas directamente desde el chatbot para su gestión, monitoreo y toma de decisiones.

Filtrado de preguntas y respuestas innecesarias: A veces, los usuarios pueden realizar preguntas y que el modelo genere respuestas altamente innecesarias, estas serán suprimidas para garantizar que las preguntas y respuestas sean únicamente corporativas.

Chatbot puesta en producción

En la actualidad, este servicio de chatbot de código abierto con IA basada en Python que incluye librerías tanto oficiales como no oficiales y librerías de total autonomía se encuentra en la línea de producción en la sede de Plaza Claro en la ciudad de Bogotá en la Avenida 68 con calle 26 cumpliendo con todos los requerimientos puestos en esta tesis y en constantes pruebas piloto para detectar fallos y aplicar mejoras en cuanto a sus entrenamientos constantes y mejoras del código propuesto, a la fecha se encuentra funcionando de manera correcta sobre más del 90% operativo. Una vez aprobado el modelo propuesto, será implementado mediante servidores OCI de Oracle mediante enlaces oficiales. En el anexo A se encuentra el código fuente del chatbot, el cuál se encuentra en proceso de registro software para temas de derechos de autor.

Enlaces repositorio del código y video

A continuación, se comparte código de GitHub y enlace de video de YouTube donde se encuentran las librerías y módulos que hacen realidad el funcionamiento del código, incluyendo, las pruebas de funcionamiento de la herramienta, lo cual, se brinda soporte técnico, mantenimiento del código y actualización de versiones.

Tabla 10

Enlace código del desarrollo y video tutorial funcionamiento

Nombre producto	Chatbot en base Python con funciones IA que brinda asesoramiento y servicio técnico para el proyecto Itel - Oracle
Enlace YouTube	https://www.youtube.com/watch?v=6-gBeSCOIhg
Repositorio GitHub	https://github.com/sebastianbeltran142/chatbot_web

Nota: La información propuesta en la tabla 10, más que todo en el repositorio de Github se encuentra en constante actualización el modelo propuesto en esta tesis, dado de que se crean nuevos módulos para el chatbot propuesto de acuerdo con la línea y recomendaciones del negocio. En el enlace de YouTube se muestra de manera detallada la funcionalidad del chatbot en el equipo de producción ubicado en la sede cliente de Plaza Claro.

Fuente: Autoría propia

Algunas pruebas iniciales

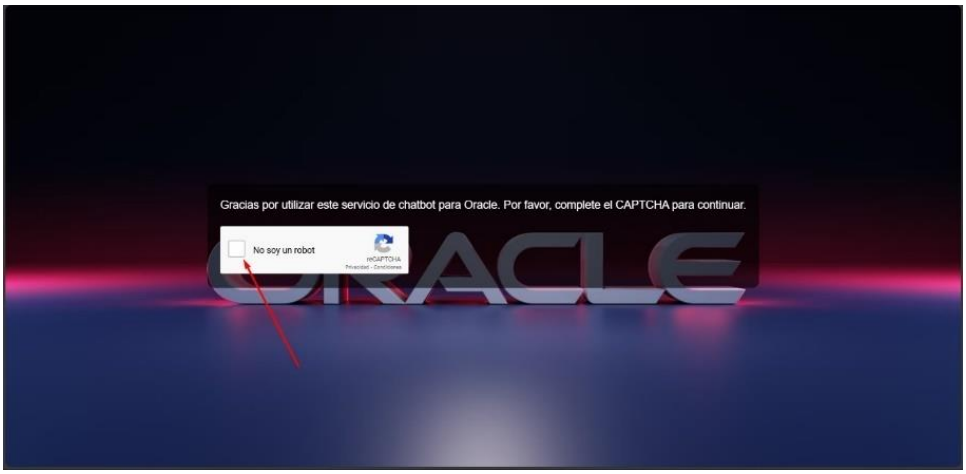
Se evidencia carga de la plataforma de chatbot en su versión moderna y profesional, adaptándose mejor para el negocio del cliente en cuanto al requerimiento.

Evidencia del reCAPTCHA de Google

Una vez se cargue la plataforma web del chatbot se muestra la presentación de bienvenida. El usuario debe ser capaz de aprobar el reCAPTCHA versión 2 de Google para su acceso al servicio del chatbot como en el ejemplo de la figura 15.

Figura 15

Presentación de bienvenida del chatbot con reCAPTCHA



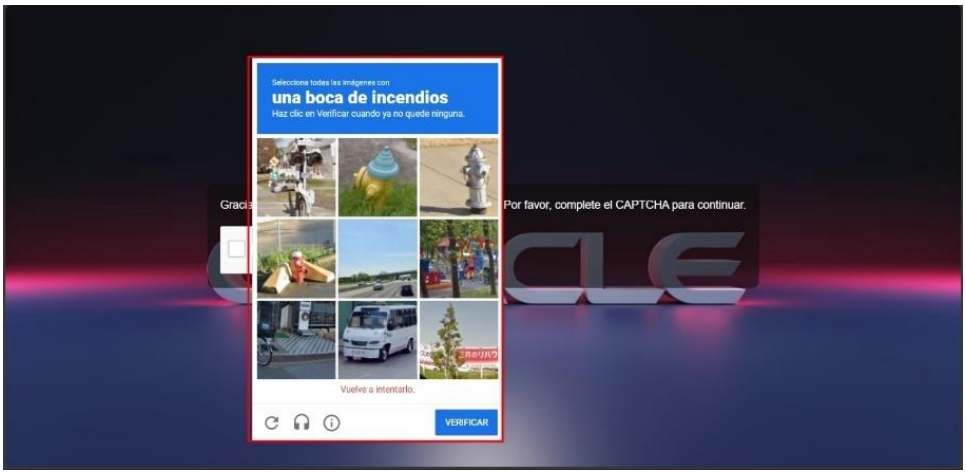
Nota: Ilustración que muestra la presentación inicial del chatbot web, el cual, se evidencia el recaptcha de Google versión 2 antes de la interacción con la herramienta.

Fuente: Autoría propia.

En algunas ocasiones, como se evidencia en la figura 16, solicitará realizar ciertas pruebas adicionales que propone la misma herramienta del reCAPTCHA. Es importante haber iniciado sesión dentro de una cuenta verídica de Google en el mismo navegador donde tendrá acceso al chatbot en caso de aplicar.

Figura 16

Prueba reCAPTCHA



Nota: Ilustración que muestra la presentación inicial del chatbot web, el cual, se evidencia el recaptcha de Google versión 2 realizando comprobación de seguridad adicional.

Fuente: Autoría propia.

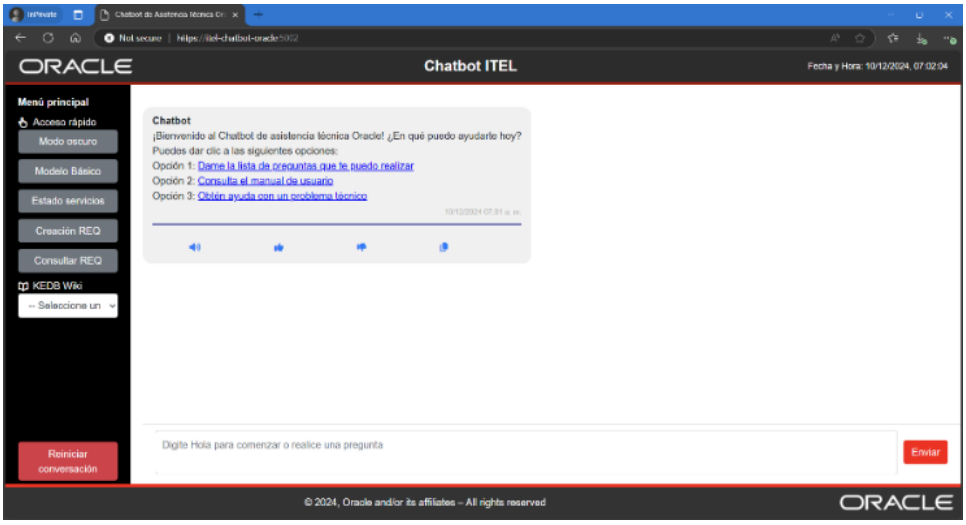
Una vez aprobado el acceso por parte de reCAPTCHA de Google, se evidencia la interfaz del clatbot como en el ejemplo de la figura 17. Al acceder a el, puede evidenciar varios servicios implementados y en construcción que harán parte de la herramienta propuesta para Oracle.

Presentación inicial de la interfaz

La interfaz de presentación y bienvenida se ha ido gradualmente modernizando con el fin de presentar al usuario una plataforma de vanguardia más amigable y profesional. Esta es la cara del chatbot actúal.

Figura 17

Presentación de bienvenida o interfaz del chatbot



Nota: Ilustración que muestra la presentación de bienvenida del chatbot.

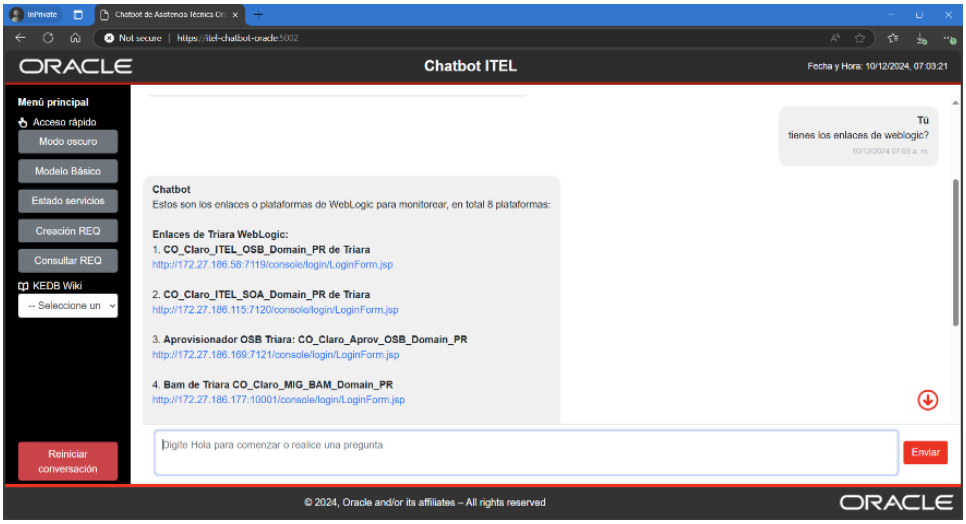
Fuente: Autoría propia.

Interacciones básicas simples

La herramienta cuenta con una base de datos base para la lograr cierta interacción del chatbot con el usuario, también cuenta con otros módulos implementados para una mejor experiencia con el usuario final. Este es un ejemplo de ello como se evidencia en la figura 18 cuando se solicita ciertos enlaces web como WebLogic mostrando una respuesta de interacción del chatbot.

Figura 18

Interacciones básicas simples



Nota: Ilustración que muestra la interacción con procesos de preguntas simples.

Fuente: Autoría propia.

Aunque se conoce que faltan algunas afinidades, dado de que a veces, las respuestas que brinda este primer modelo no son totalmente precisas, más es un buen comienzo, lo cual, mientras se afina el modelo a su vez entrenando el chatbot puede ir concatenando respuestas más cercanas dependiendo del requerimiento del negocio.

Interacciones matemáticas

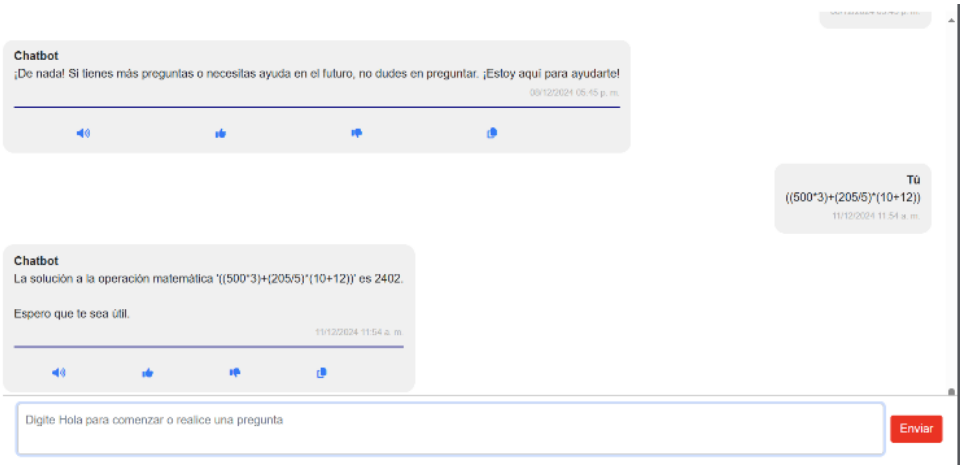
Con el chatbot, puede realizar operaciones matemáticas básicas con solo números y combinación de signos como suma, resta, multiplicación y división. Se puede aplicar esta información de manera directa en el chatbot sin necesidad de llamar otra librería. Es bueno realizar un proceso de registro de manera correcta en el chatbot con la finalidad la operación sea una experiencia interesante, de lo contrario puede generar errores comunes por compatibilidad o falta de comprensión del chatbot. En el ejemplo de la figura 19, se solicitó al chat realizar la siguiente operación básica:

$$\left((500 * 3) + \left(\frac{205}{5}\right) * (10 + 12)\right)$$

Este fue el resultado evidenciado en el chatbot.

Figura 19

Operaciones matemáticas básicas en chatbot



Nota: Ilustración que muestra la interacción con procesos de matemáticas básicas.

Fuente: Autoría propia.

Se realizó para esta última operación una prueba directamente con la página de Mathway que se halla con internet con fines comparativos evidenciando el siguiente resultado mostrado en la figura 20 como ejemplo.

Figura 20

Comprobación de cálculo matemático en Mathway

Nota: Ilustración que muestra la interacción con procesos de matemáticas básicas en la página web de Mathway con la finalidad de realizar comparativas del resultado con el chatbot.

Fuente: Pruebas realizadas a partir de la página web Mathway.

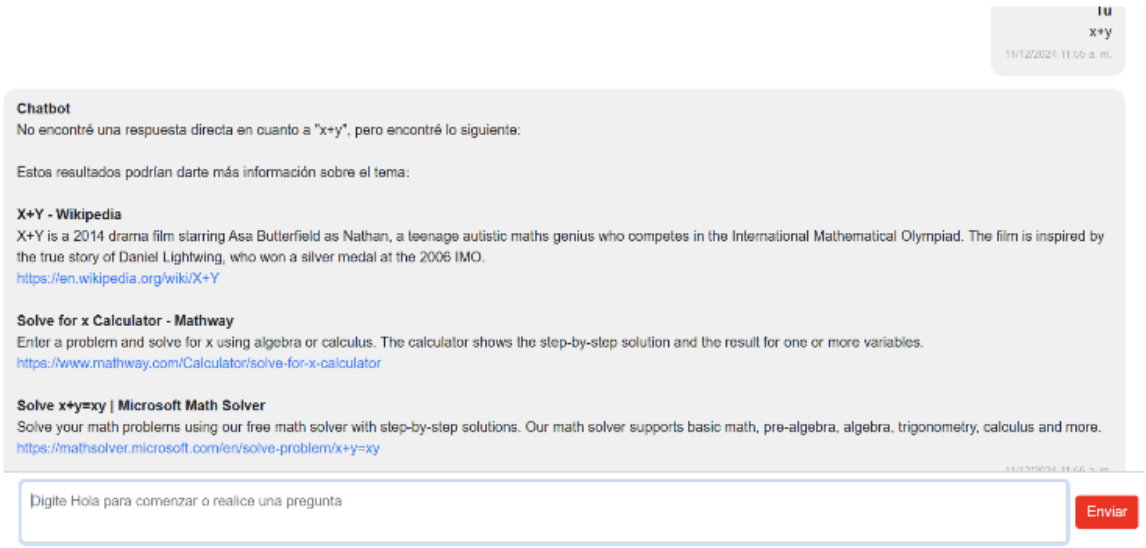
A pesar de que el modelo de chatbot no cuenta con una interfaz dinámica para mostrar las fracciones matemáticas similares a la página de Mathway, este tiene la capacidad de realizar operaciones diversas con números matemáticos. El chatbot no realiza operaciones con símbolos como letras u otro tipo de fracciones y no está diseñado para este fin, ya que no cuenta con la capacidad para realizar este tipo de operaciones avanzadas, más cuenta con la ventaja de proporcionar información sobre enlaces o páginas web que brindan operaciones matemáticas más complejas con la ayuda del buscador integrado DuckDuckGo mediante módulo no oficial. Por ejemplo, se realizó la siguiente suma algebraica básica. En el ejemplo de la figura 21, se evidencia el siguiente resultado en cuanto se requiere lanzar la operación propuesta:

$$x + y$$

Lo cual obtuvo el siguiente resultado:

Figura 21

Operaciones algebraicas x + y



Nota: Ilustración que muestra la interacción con opciones de respuesta propuestas por chatbot en caso de no saber resolver matemáticas complejas.

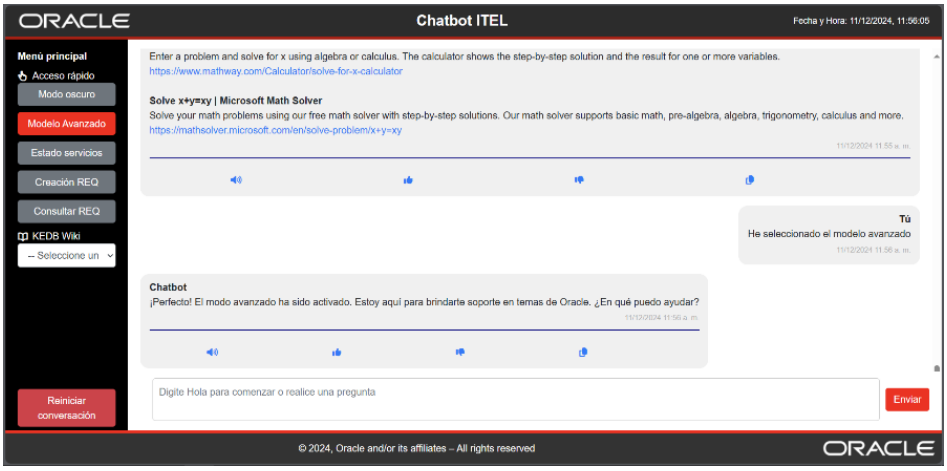
Fuente: Autoría propia.

Selección de modelo del chatbot para tareas más complejas y solución de errores de la mesa de servicios

El chatbot propuesto para el proyecto Itel de Oracle cuenta con tecnología GPT para el procesamiento de palabras y solvencia de problemas de manera segura, con la finalidad de realizar investigaciones en cuanto a errores de los servicios de Oracle. Al activar el servicio, puede interactuar únicamente en cuanto a relación con los servicios Oracle. El chatbot inicialmente informa al técnico o al usuario que el servicio de IA habilitado es únicamente para servicios relacionados con la empresa Oracle y con el negocio como se evidencia en la figura 22.

Figura 22

Cambio de modelo básico a modelo avanzado con IA



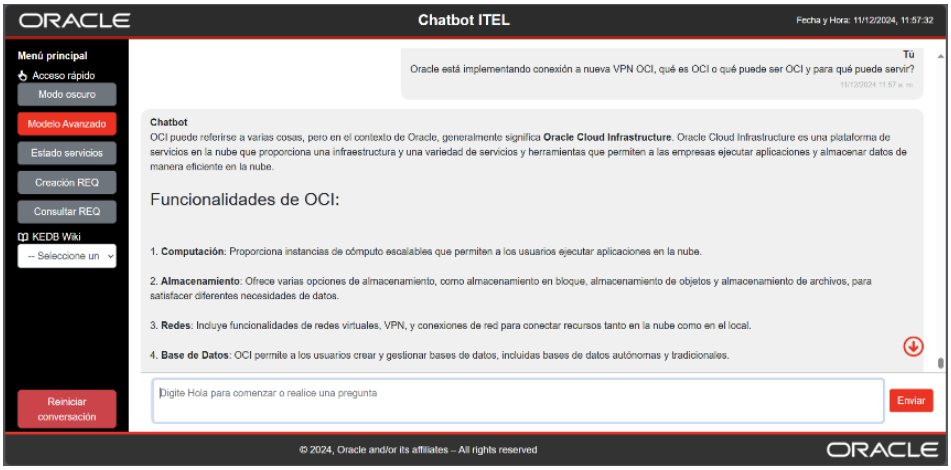
Nota: Ilustración que muestra el cambio de interacción de chatbot básico a chatbot avanzado o con IA.

Fuente: Autoría propia.

Una vez activado el servicio, se puede realizar preguntas mucho más complejas para el procesamiento interno mediante módulos ya definidos para GPT con IA. En el ejemplo de la figura 23 se simuló preguntando “Oracle está implementando conexión a VPN OCI, ¿Qué es OCI… y para qué puede servir?”, lo cual ha brindado una respuesta de ejemplo.

Figura 23

Respuesta generada por modelo avanzado IA



Nota: Ilustración que muestra respuesta generada por chatbot a partir del modelo avanzado o con IA.

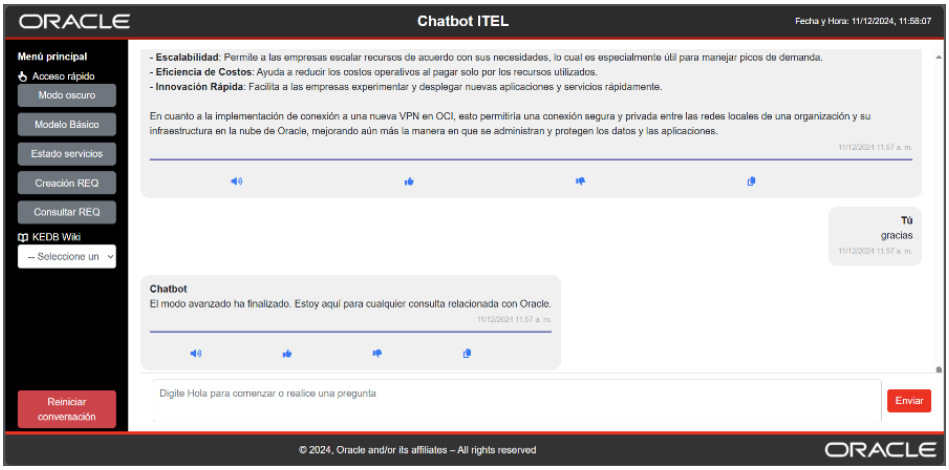
Fuente: Autoría propia.

Lectura en voz alta, me gusta, no me gusta y copiar respuesta.

El chatbot cuenta con opción de dictado o lectura de voz alta, lo que ayuda a las personas con discapacidad visual escuchar las respuestas generadas por la herramienta. Adicional a ello, cuenta con otras funcionalidades como me gusta, no me gusta y copiar, fundamentales para el seguimiento en cuanto a la funcionalidad del chatbot. Se evidencian los íconos en el ejemplo de la figura 24.

Figura 24

Íconos opción "Me gusta", "No me gusta", leer respuesta en voz alta



Nota: Ilustración que muestra los iconos de me gusta, no me gusta y lectura de respuesta en voz alta en chatbot.

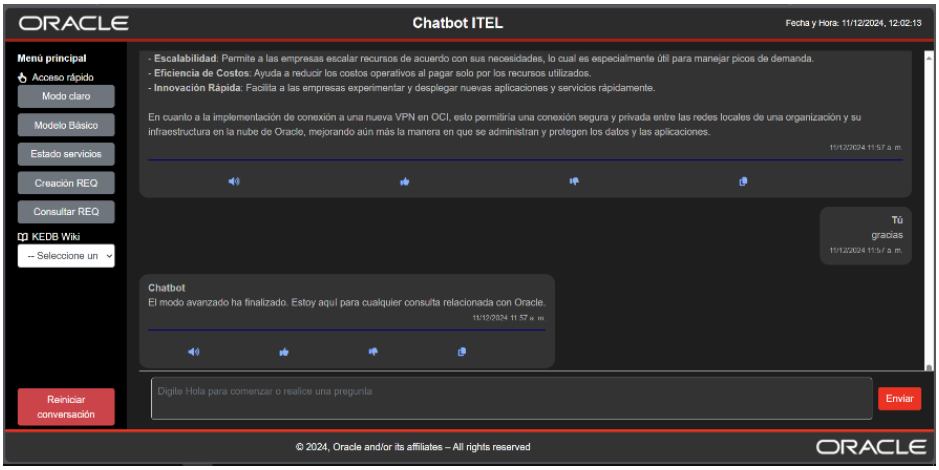
Fuente: Autoría propia.

Cambio de interfaz a modo claro / modo oscuro

La interfaz del chatbot cuenta con la opción modo claro y modo oscuro. Este es fundamental más que todo para los que cuentan con computadoras que tienen mucho brillo. También permite descansar la vista. En el ejemplo de la figura 25 se muestra la opción de modo oscuro.

Figura 25

Cambio modo claro y oscuro



Nota: Ilustración que muestra el cambio de interfaz del chatbot en modo claro a oscuro.

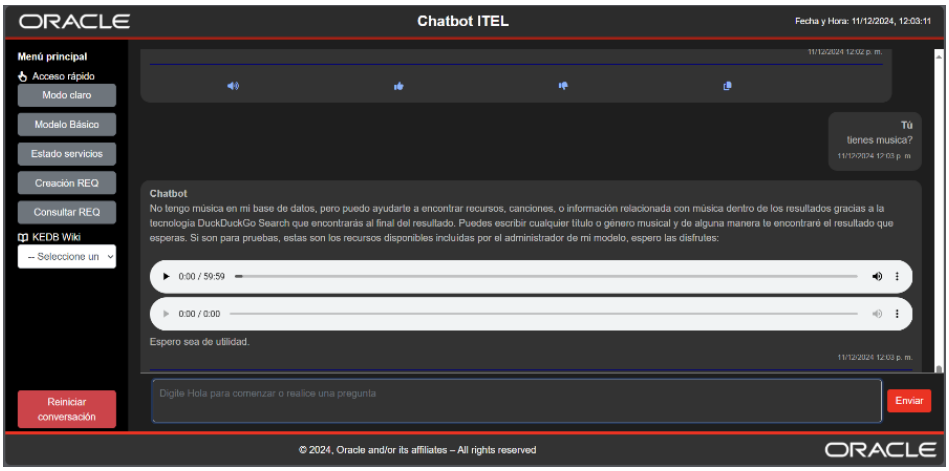
Fuente: Autoría propia.

Reproducción multimedia

El chatbot tiene la capacidad de realizar reproducciones multimedia como MP3, MP4 que cuente dentro de su base de datos como se evidencia a continuación en la figura 26.

Figura 26

Reproducción multimedia en chatbot



Nota: Ilustración que muestra la reproducción multimedia en chatbot.

Fuente: Autoría propia.

También cuenta con la capacidad para la reproducción de videos enlazados en línea de YouTube. Esto es ideal para escuchar música y ver tutoriales en línea como en el ejemplo que se muestra en la figura 27. A continuación, se anexa un ejemplo solicitado al chatbot en cuanto a música de Mozart.

Figura 27

Reproducción de videos YouTube

Nota: Ilustración que muestra la reproducción de contenidos de YouTube en chatbot.

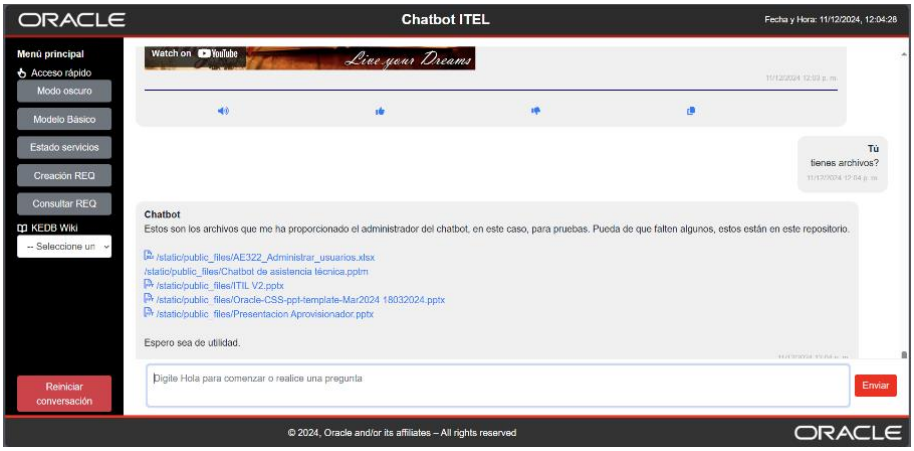
Fuente: Autoría propia.

Acceso a archivos digitales

El chatbot cuenta con la opción de mostrar qué archivos cuenta y lo pueda descargar directamente desde el navegador, lo cual es útil para la muestra de manuales y tutoriales en archivos y libros del proyecto Itel de Oracle. En la figura 28, se muestra un ejemplo de la visualización de archivos desde su interfaz del chatbot.

Figura 28

Visualización de archivos digitales como manuales y tutoriales



Nota: Ilustración que muestra los archivos digitales disponibles en chatbot.

Fuente: Autoría propia.

Las demás pruebas iniciales se ven reflejadas en el clip de video publicado en YouTube para su mejor comprensión.

Pruebas funcionales al chatbot

Este capítulo aborda los avances más significativos en el desarrollo del chatbot, abarcando las pruebas piloto y pruebas funcionales en su implementación y despliegue. Este proceso permite verificar que el sistema cumpla con las especificaciones establecidas y funcione conforme a lo propuesto en esta tesis, así como a las expectativas en el entorno proporcionado por la empresa Claro Soluciones mediante el proyecto Itel de Oracle. El propósito principal de estas pruebas es asegurar que el chatbot gestione adecuadamente las consultas y solicitudes de los usuarios, manteniendo un rendimiento eficiente y ofreciendo respuestas que se perciban lo más naturales y humanas posibles. Esto es particularmente relevante en el contexto empresarial de Claro Soluciones, donde el chatbot debe integrarse con sistemas internos, gestionar tickets REQ. y responder a consultas complejas vinculadas al proyecto.

Los objetivos específicos de las pruebas funcionales en Claro Soluciones incluyen:

Detección de fallas: Identificar y registrar errores o comportamientos no deseados en el chatbot para corregirlos antes de su despliegue completo. Esto abarca fallos en la generación de respuestas, errores en la gestión de tickets REQ. o problemas de conexión con las bases de datos de la empresa.

Identificación de áreas de mejora: Evaluar el desempeño del chatbot en distintos escenarios de uso para determinar posibles mejoras en áreas como la precisión de las respuestas, la rapidez de procesamiento y la capacidad de comprender consultas complejas.

Optimización de la experiencia del usuario: Asegurar de que la interfaz del chatbot sea intuitiva y fácil de usar, además de garantizar que las respuestas sean claras, relevantes y útiles para los técnicos y usuarios de la mesa de servicios.

En lo posible, el chatbot pueda generar errores en cuanto a la entrega de respuestas simples e intermedias, estas pruebas son importantes ya que con ello nos ayuda a realizar los ajustes pertinentes al modelo propuesto en el chatbot. Adicional a ello, estas pruebas nos ayudan a garantizar que el chatbot sea una herramienta más confiable y robusta, lista para ser implementada en un entorno de producción y capaz de agregar valor a las operaciones diarias de la empresa.

Pruebas de desempeño

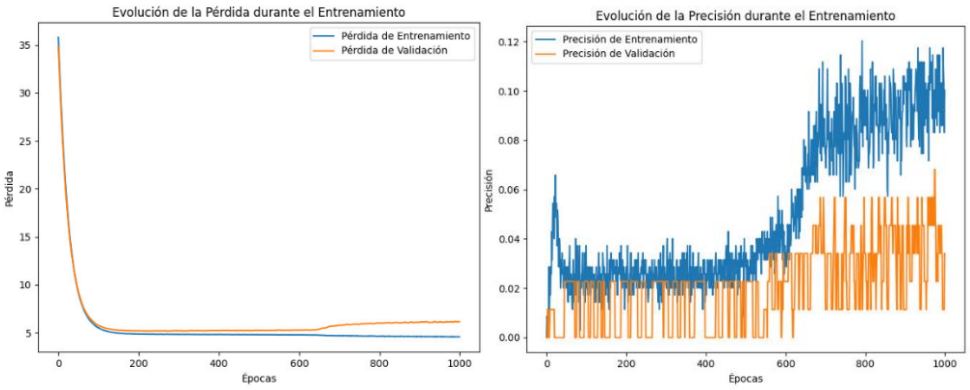
Este apartado detalla las pruebas de desempeño realizadas para evaluar la eficacia del chatbot en términos entrenamiento, tiempo de respuesta, uso de recursos, creación de requerimientos y escalabilidad entre otros servicios. Estas pruebas permiten validar que el sistema cumple con los requisitos de rendimiento necesarios y ofrece una experiencia óptima para los técnicos y usuarios.

Resultado pruebas de entrenamiento

A continuación, se grafica los entrenamientos realizados en el chatbot en cuanto a las épocas, evolución de pérdidas durante el entrenamiento y evolución de la precisión. Estas gráficas son generadas directamente desde el modelo chatbot dentro de sus herramientas ya implementadas para mostrar el resultado de estos entrenamientos.

Figura 29

Evolución de la pérdida y precisión durante el entrenamiento etapa 1



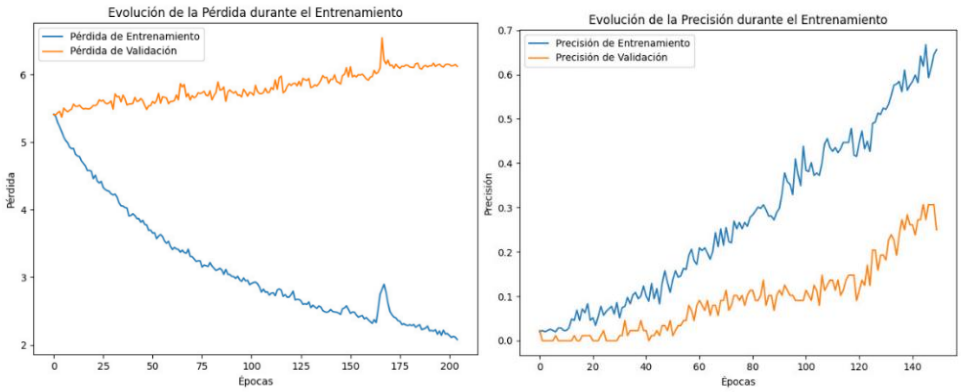
Nota: Se muestra una ilustración gráfica acerca de los comportamientos de entrenamiento del chatbot en su primera etapa. Información obtenida utilizando el modelo básico.

Fuente: Autoría propia.

Se evidencia en la primera y segunda imagen de la figura 29 en cuanto al entrenamiento del modelo. La pérdida de entrenamiento disminuye rápidamente, mostrando un aprendizaje efectivo, pero la pérdida de validación se estabiliza a un nivel superior, sugiriendo un posible sobreajuste. Esto es debido a que en las primeras pruebas se evidencia mejor aprendizaje, aunque, en cuanto al sobreajuste genera doble procesamiento del modelo. Eso es ajustable desde el código.

Figura 30

Evolución de la pérdida y precisión durante el entrenamiento etapa 2



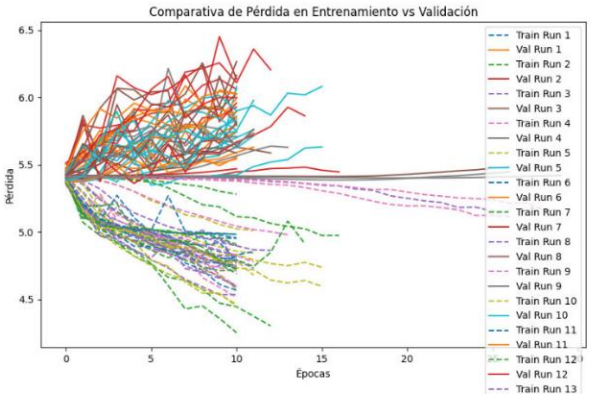
Nota: Se muestra una ilustración gráfica acerca de los comportamientos de entrenamiento del chatbot en su segunda etapa. Información obtenida utilizando el modelo básico.

Fuente: Autoría propia.

En las siguientes imágenes comparativas de la figura 30, la precisión en entrenamiento aumenta, mientras que la precisión en validación fluctúa y se mantiene más baja, indicando que el modelo no generaliza bien y podría necesitar ajustes para mejorar en datos no vistos. Aunque en este ajuste, se dio cuenta de que la base de datos es escasa, lo cual genera mucho ruido. Para mejorar esto, es necesario implementar la base de conocimientos en el modelo. En la figura 31 se muestra una evaluación mediante épocas del entrenamiento del chatbot.

Figura 31

Evaluación de entrenamiento mediante épocas en chatbot



Nota: Se muestra una ilustración gráfica acerca de los comportamientos de entrenamiento del chatbot mediante épocas. Información obtenida utilizando el modelo básico.

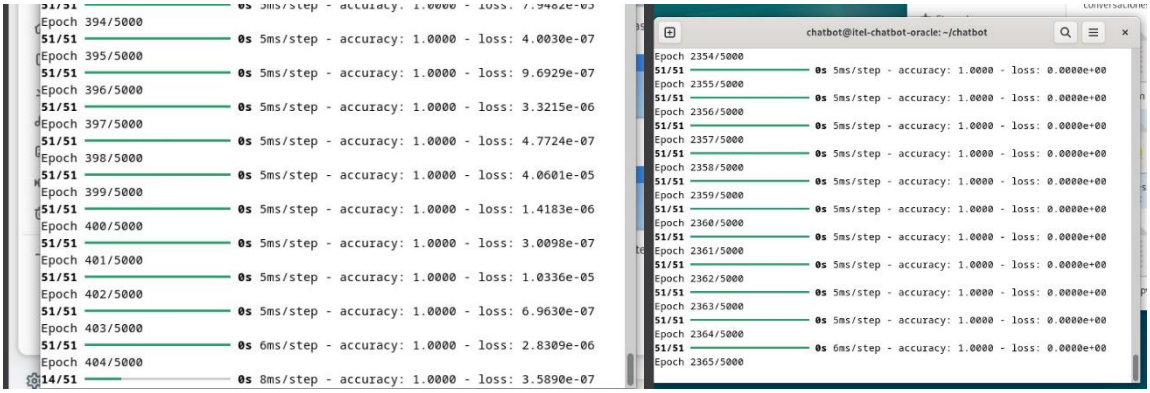
Fuente: Autoría propia.

La figura 31 ofrece una visualización más clara de la comparativa de pérdida entre el entrenamiento y la validación a lo largo de múltiples épocas. Cada línea representa una "ejecución" o "run" del modelo, permitiendo observar cómo la pérdida varía en las primeras épocas y luego se estabiliza en diferentes niveles. Una vez que el modelo identifica el mejor camino de entrenamiento, mantiene un flujo constante que favorece el ajuste en cada época, logrando así una precisión en las respuestas de más del 90% en el chatbot.

A pesar del avance logrado, se considera que el modelo puede ser aún más robusto con una mayor cantidad de datos en su base de conocimiento. Cuanta más información se proporcione, mayor será su capacidad para afinar el entrenamiento y ofrecer respuestas aún más precisas. Sin embargo, con el ajuste actual, el modelo es completamente funcional y adecuado para operar en su estado actual, brindando una interacción eficaz para el usuario.

Figura 32

Muestra entre entrenamiento en cuanto a precisión y pérdida



Nota: Ilustración que muestra el entrenamiento del modelo del chatbot con IA en producción.

Fuente: Autoría propia.

En la figura 32 muestra un entrenamiento del modelo de chatbot en el cual la precisión (accuracy) se mantiene en 1.0000 y la pérdida (loss) es de 0.0000 en múltiples épocas, lo que sugiere que el modelo ha alcanzado un sobreajuste perfecto en el conjunto de entrenamiento. Aunque esta precisión puede parecer ideal, es probable que el modelo no generalice bien a datos nuevos. Aun con este sobreajuste y con varias pruebas realizadas, es ideal para garantizar el entrenamiento y funcionamiento del chatbot.

Resultados pruebas de rendimiento

Se evaluó el tiempo que tarda el chatbot en responder a consultas de diferentes niveles de complejidad (simples, intermedias y complejas). Al principio, mientras el chatbot inicia con su consulta en cuanto al encontrar la respuesta tiende a tardar alrededor de 5 segundos. Una vez realizada de nuevo la consulta el rendimiento fue mejor. A continuación, se muestra en la tabla 11 estos resultados en cuanto a tiempo y complejidad:

Tabla 11

Evaluación del nivel de complejidad en chatbot

Nivel complejidad	Prueba 1	Prueba 2	Prueba 3
Simple	5 segundos	4 segundo	2 segundos
Intermedias	5 segundos	4 segundos	3 segundos
Complejas	8 segundos	6 segundos	5 segundos

Nota: Tabla que muestra la comparativa en cuanto a tiempos de respuesta y complejidad del chatbot.

Fuente: Autoría propia

Análisis acerca de los resultados

Consultas simples: Se refiere a consultas frecuentes. Tienen tiempos de respuesta rápidos, mejorando progresivamente entre pruebas (de 4 a 2 segundos), lo que indica eficiencia en tareas básicas.

Consultas intermedias: Se refiere a nuevos entrenamientos, registros para las notificaciones y muestra de resultados en cuanto al servicio de monitoreo y bases de datos. Mantienen una respuesta estable en las primeras dos pruebas (4 segundos), con ligera mejora en la tercera prueba (3 segundos). Estos datos pueden variar de acuerdo con la disponibilidad de las bases de datos, servicios de monitoreo y notificaciones y cálculos binarios, así como pruebas en cuanto a servicios de telecomunicaciones.

Consultas complejas: Se refiere a la generación de texto avanzado con librerías IA y respuestas consultadas por internet. Los tiempos se mantienen alrededor de los 5 y 8 segundos, ya que estas tareas dependen exclusivamente desde la consulta de las bases de datos, hasta servicios avanzados. En caso de que estos servicios superen el tiempo, el chatbot genera una

alerta si algunos de los servidores se presentaron afectación o falta de actualización de la información como caducidad de contraseñas o falla en los servicios.

En general, los tiempos de respuesta del chatbot se mantienen y van mejorando con cada prueba, especialmente en tareas de consultas básicas e intermedias. Hay que tener en cuenta que estos servicios dependen de manera exclusiva para acceder a la información a bases de datos, herramientas de MyIT, WhatsApp o afines una vez sea solicitado por el técnico y el usuario. Aun así, ofrece buenos resultados en cuanto a esta capacidad y rendimiento del servidor local. En la tabla 12 se observa resultados globales de acuerdo con el desempeño en el equipo de experiencia.

Tabla 12

Resultados en pruebas de desempeño

Aspecto Evaluado	Resultado	Observación
Tiempo de Respuesta	2 a 4 segundos (Puede variar de acuerdo con la complejidad)	Cumple con el estándar para interacciones en tiempo real; mejora con optimización adicional.
Uso de CPU	30% en carga normal, 70% alta	Uso eficiente; posible ajuste para cargas extremas. Probado con 2 hilos.
Uso de RAM	2 GB en carga normal, 8 GB alta	Suficiente para producción estándar; escalar en casos de alta demanda. La máquina actualmente cuenta con 16 GB de memoria RAM asignada.

Escalabilidad	Estable hasta 15 usuarios simultáneos de manera local (Ya que se utiliza flask de python)	Requiere escalado horizontal mediante servidores optimizados a partir de 100 usuarios en adelante.

Nota: Tabla que contiene información detallada en cuanto apruebas de desempeño del chatbot en producción.

Fuente: Autoría propia

Resultado de las pruebas del modelo chatbot para Itel Oracle

Para esta actividad, se realizaron simulaciones básicas del modelo del chatbot para Oracle en código Python en cuanto a la simulación de rendimiento y uso de los recursos estabilizados. No se tuvo pruebas con los demás lenguajes de programación debido a la falta de experiencia por parte del autor de esta tesis y no va aplicado un desarrollo sofisticado en cuanto a su operación.

Propiedades de los equipos para el laboratorio del chatbot

Estas son las propiedades básicas de tres equipos que se realizaron pruebas con el modelo de chatbot propuesto.

Propiedades del equipo 1

A continuación, se presentan las características del Equipo 1 en la Tabla 13. Este equipo forma parte del conjunto de herramientas base utilizadas para el desarrollo del proyecto final. Actualmente, se encuentra en proceso de actualización para garantizar su óptimo rendimiento.

Tabla 13

Información características equipo 1

Device name	Asus Vivobook – Personal

Processor	11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz 3.11 GHz
Installed RAM	8,00 GB
System type	64-bit operating system, x64-based processor
Operative System	Windows 10

Nota: Se detalla información sobre las propiedades del equipo 1.

Fuente: Autoría propia

Propiedades del equipo 2

A continuación, se presentan las características del Equipo 2 en la Tabla 14. Este equipo pertenece a la línea corporativa y está destinado al servicio de chatbot en modo maqueta o QA. En él se llevaron a cabo las pruebas iniciales del chatbot antes de su implementación en producción.

Tabla 14

Información características equipo 2

Device name	AsusPro P1440FA
Processor	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
Installed RAM	16,00 GB
System type	64-bit operating system, x64-based processor
Operative System	Windows 11

Nota: Se detalla información sobre las propiedades del equipo 2.

Fuente: Autoría propia

Propiedades del equipo 3

A continuación, se presentan las características del equipo servidor-cliente, donde el chatbot se encuentra en producción, como se detalla en la Tabla 15.

Tabla 15

Información características equipo sede cliente

Device name	Dell Inspiron 14-3467
Processor	Xeon E3-1200 v6/7th Gen Core Processor Host Bridge/DRAM Registers
Installed RAM	16,00 GB
System type	64-bit operating system, x64-based processor
Operative System	Linux Ubuntu

Nota: Se detalla información sobre las propiedades del equipo 3.

Fuente: Autoría propia

Requerimientos mínimos del modelo chatbot

De acuerdo con el análisis en cuanto al consumo de memoria RAM, procesamiento y hardware actual, estas son las que se evidenciaron en el modelo que se ha implementado en una de las máquinas:

Tabla 16

Requerimientos mínimos del chatbot en cuanto a hardware - software

Propiedades	Mínimo requerido	Observaciones

Processor	1.5 GHz	Velocidad mínima para garantizar procesamiento de consultas rápidas y efectivas.
RAM	8 GB	Memoria suficiente para el funcionamiento básico; los módulos de entrenamiento pueden requerir más.
Operative System	Windows 10, Windows 11, Linux Ubuntu, Debian	Compatibilidad probada con sistemas operativos comunes para entornos de desarrollo y producción.
System type	64-bit operating system, x64-based processor	Recomendado para optimizar el rendimiento y compatibilidad con librerías de IA/PLN.
Storage	256 GB SSD	Para almacenamiento de datos temporales y permanentes, preferiblemente en formato SSD para mayor rapidez
Software Base	Python 3.x, Matplotlib, NLTK, TensorFlow, Flask	Herramientas y frameworks necesarios para el desarrollo e implementación de las funcionalidades de IA.

Nota: Se describe la información mínimo requerido para la instalación del chatbot en equipos de prueba y producción.

Fuente: Autoría propia

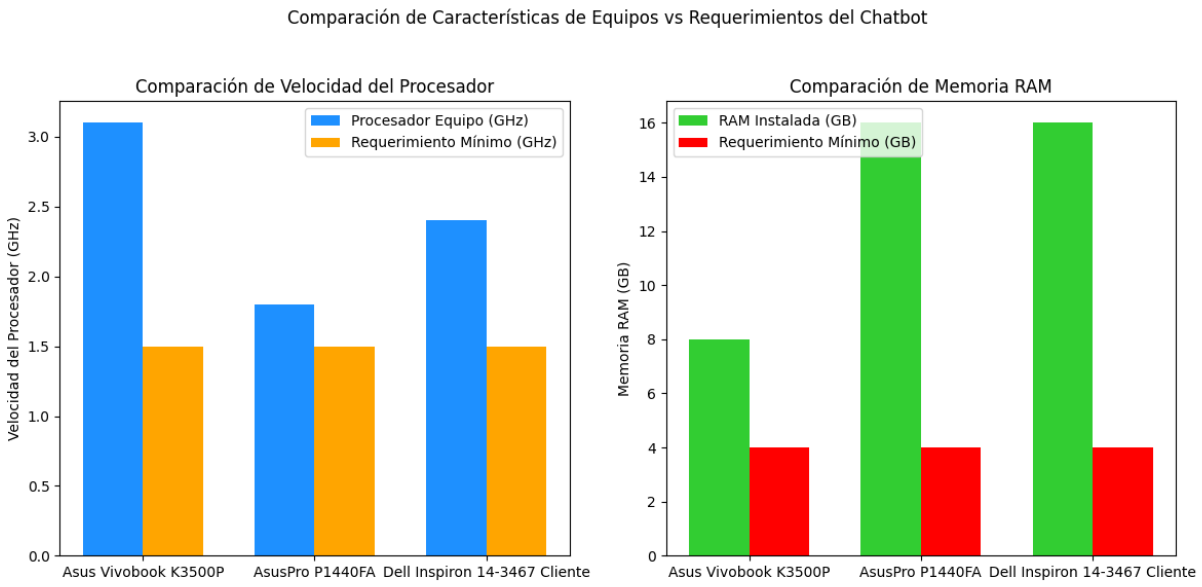
Este cuadro resume las especificaciones mínimas que aseguran un rendimiento estable del chatbot en entornos de producción y entrenamiento en Python.

Comparativa de rendimiento en equipos locales y sede cliente

A continuación, se muestra una gráfica visual en cuanto al comportamiento del chatbot frente a los equipos relacionados anteriormente:

Figura 33

Gráfica comparativa entre hardware de los equipos y requisitos del chatbot



Nota: Ilustración gráfica que contiene la comparativa de los requisitos mínimos en cuanto al consumo de recursos del chatbot frente a las máquinas propuestas en esta tesis.

Fuente: Autoría propia

La gráfica de la figura 33 muestra que todos los equipos evaluados superan los requerimientos mínimos de velocidad de procesador y memoria RAM necesarios para el funcionamiento del chatbot, destacándose el Asus Vivobook en rendimiento del procesador y el AsusPro y Dell Inspiron en capacidad de memoria RAM, lo cual asegura un rendimiento óptimo en el uso y entrenamiento del modelo. Para en este caso, no se evaluó con tarjetas avanzadas

como CUDA (Compute Unified Device Architecture), lo que en este caso en algunas ocasiones puede afectar el rendimiento de la PC.

Muestra de rendimiento del sistema y consumo de chatbot en producción

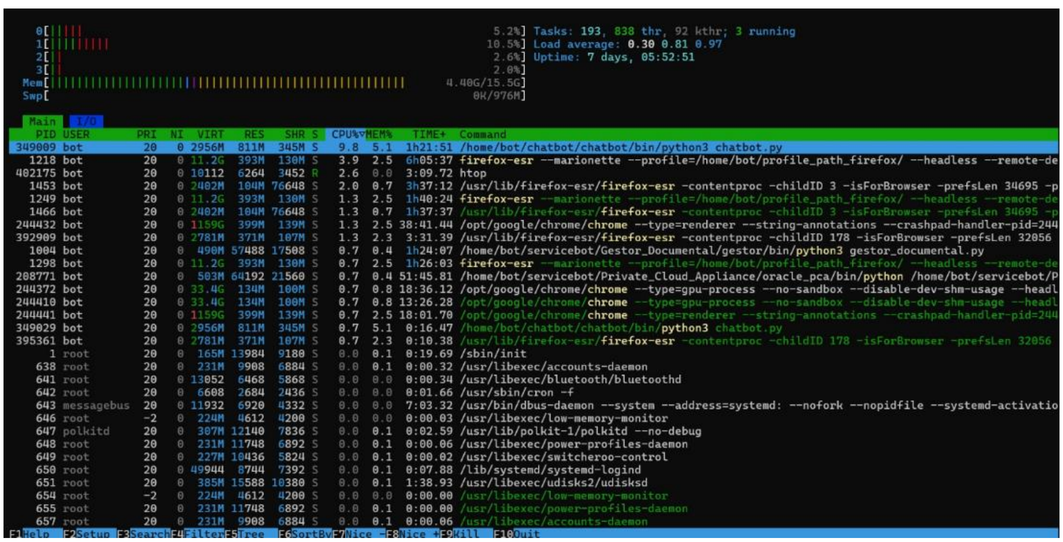
En este apartado, se muestra un análisis detallado en cuanto al rendimiento de la PC y el consumo del chatbotpuesto en funcionamiento:

Pruebas de rendimiento del sistema con htop

A continuación, se presenta el análisis del consumo de recursos de hardware y software del sistema, utilizando la herramienta htop, como se muestra en la Figura 34.

Figura 34

Htop con procesos activos del sistema



Nota: Ilustración que muestra el consumo de recursos del sistema en el servidor de producción con htop.

Fuente: Autoría propia

Se comparte información detallada de la figura 34 en la tabla 17 para su mejor comprensión:

Tabla 17

Informe de monitoreo con htop con más procesos activos

Concepto	Detalle
Memoria RAM	15.5 GB total, 4.40 GB en uso
CPU	Aumento de carga debido a procesos adicionales como Firefox ESR y Chrome
Hilos	Se observan procesos con alto consumo de memoria (firefox-esr, chrome)
Consumo del Chatbot	Sigue usando ~800 MB por instancia
Conclusión parcial	Hay más procesos activos en comparación con la primera imagen, lo que puede impactar el rendimiento

Nota: Se muestra resumen general en cuanto al comportamiento del chatbot frente a otros programas en el equipo cliente ubicado en la sede de Plaza Claro.

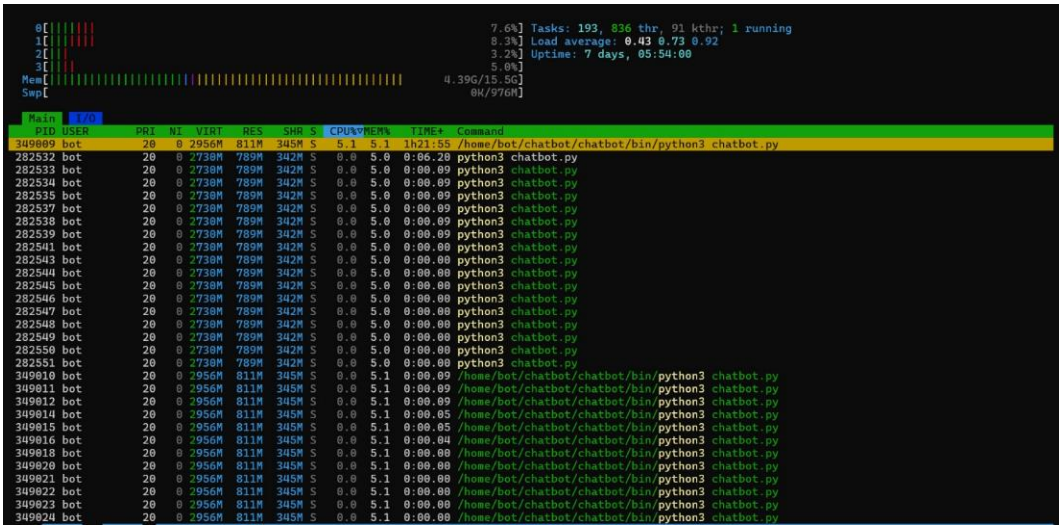
Fuente: Autoría propia

Pruebas de rendimiento del chatbot con htop

A continuación, se presenta el análisis del consumo de recursos de hardware y software del chatbot, utilizando la herramienta htop, como se muestra en la Figura 35.

Figura 35

Htop con múltiples procesos del chatbot



Nota: Ilustración que muestra el consumo de recursos del chatbot en el servidor de producción con htop.

Fuente: Autoría propia

Se comparte información detallada de la figura 35 en la tabla 18 para su mejor comprensión:

Tabla 18

Informe del consumo de PC con chatbot

Concepto	Detalle
Memoria RAM	15.5 GB total, 4.39 GB en uso
CPU	5.1% de uso en el proceso principal del chatbot
Hilos	Múltiples instancias de chatbot.py ejecutándose simultáneamente
Consumo del Chatbot	Cada proceso de python3 chatbot.py usa 789 MB de RAM

Conclusión parcial	Hay varios procesos ejecutando la misma tarea, lo que puede afectar el rendimiento

Nota: Se muestra resumen general en cuanto al comportamiento del consumo de recursos del chatbot en el equipo cliente ubicado en la sede de Plaza Claro.

Fuente: Autoría propia

Pruebas de rendimiento de hardare y software y recursos consumidos del chatbot con bashtop

A continuación, se presenta el análisis del consumo de recursos de hardware y software del chatbot y otras aplicaciones que requiere el sistema para su funcionamiento óptimo, utilizando la herramienta bashtop, como se muestra en la Figura 36.

Figura 36

Bashtop para análisis gráfico general

Nota: Ilustración que muestra el consumo de recursos del sistema y chatbot en el servidor de producción con bahtop.

Fuente: Autoría propia

Se comparte información detallada de la figura 36 en la tabla 19 para su mejor comprensión:

Tabla 19

Informe consumo general de proceso en equipo de producción

Concepto	Detalle
Memoria RAM	15.8 GB total, 4.88 GB en uso (31%)
CPU	Uso promedio de 25-30% en los núcleos
Hilos y Procesos	Varios procesos activos, python3 chatbot.py sigue siendo el más relevante
Consumo del Chatbot	Cada instancia sigue ocupando entre 789 MB y 811 MB de RAM
Conclusión parcial	El sistema tiene margen de trabajo, pero es recomendable reducir el número de procesos del chatbot

Nota: Se muestra resumen general en cuanto al comportamiento del consumo de recursos del chatbot y otros recursos necesarios para su funcionamiento en el equipo cliente ubicado en la sede de Plaza Claro.

Fuente: Autoría propia

Resumen final del estado de CPU

La memoria RAM del equipo se evidencia con más del 60% libre

La CPU en general se evidencia buen funcionamiento de manera moderada entre un 25% y 30%, lo que significa que el sistema responde de manera acertada.

En cuanto al consumo del chatbot, está usando un aproximado de memoria de 700 MB y 800 MB de manera consecutiva, sin tener en cuenta el caché necesario para la estabilidad neuronal recurrente.

Los navegadores actuales (Firefox – Chrome…) se mantienen activos, estos son parte fundamental en cuanto al funcionamiento con WhatsApp para el envío de las notificaciones y alertas en tiempo real gracias a la pronta respuesta de monitoreo general del chatbot.

Recomendaciones y mejoras futuras

El presente capítulo aborda las recomendaciones y posibles mejoras para optimizar el desempeño y funcionalidad del chatbot desarrollado, así como su adaptabilidad a las necesidades futuras de la empresa Claro Soluciones y otras organizaciones que deseen implementar sistemas similares. Estas sugerencias están basadas en los hallazgos obtenidos durante el desarrollo, pruebas y evaluación del proyecto. Este proyecto aplica las mejoras por recomendación del cliente, profesionales de TI y personal experimental que se encuentra activo en el proyecto y del autor de esta tesis:

Ampliación del entrenamiento del sistema

El modelo actual puede beneficiarse de un entrenamiento más amplio basado en datos obtenidos de:

Interacciones reales desde el chatbot: Incorporar feedback directo de los usuarios para ajustar y mejorar las respuestas en caso de que las respuestas sean erróneas por el chatbot.

Actualizaciones constantes: Incluir nuevas consultas y escenarios relevantes a medida que evolucionen las necesidades del negocio. Gracias a su desarrollo en Python, permite conectar nuevos módulos personalizados.

Ampliación de la base de conocimientos: Incrementar la información relacionada con la empresa Oracle como tramas, promociones, configuraciones, paquetes, tutoriales, wikis, videos y otros datos relevantes para brindar un soporte integral y completa.

Optimización de la interfaz de usuario

Para aumentar la accesibilidad y usabilidad del chatbot, se sugiere:

Opciones de Personalización: Permitir que los usuarios configuren preferencias como mantener tema oscuro posterior a una actualización o refresco del navegador, tamaño de fuente y ampliar los atajos de acceso para respuestas aún más precisas.

Compatibilidad con Dispositivos Móviles: Similar al anterior, optimizar la interfaz para un uso eficiente en smartphones y tablets, aunque esta parte debe ser aprobada por negocio.

Integración de Accesibilidad: Implementar herramientas para usuarios con discapacidades, como soporte para comandos por voz.

Ampliación de la capacidad de hardware

Para garantizar una mayor eficiencia en el rendimiento del sistema, se sugiere:

Implementación de CUDA: Utilizar capacidades de procesamiento paralelo por recomendación de algunas librerías para acelerar las operaciones de entrenamiento y respuesta.

Actualización, implementación de procesadores: Incorporar hardware avanzado que permita mayor rapidez en las operaciones del chatbot.

Escalabilidad del sistema

Finalmente, se recomienda preparar el chatbot para futuros escenarios empresariales mediante:

Integración multicanal: Conexión con plataformas como WhatsApp o Microsoft Teams para ampliar su alcance.

Soporte para interacciones simultáneas: Aumentar la capacidad de manejar un mayor número de consultas concurrentes.

Adaptación a nuevas tecnologías: Incorporar avances como computación en la nube o integraciones con IoT para ampliar sus aplicaciones.

Migración a la infraestructura OCI de Oracle

Se recomienda migrar el chatbot a la infraestructura OCI para aprovechar sus beneficios, entre ellos:

Mayor escalabilidad: OCI proporciona una arquitectura cloud que facilita la adaptación del sistema a crecientes volúmenes de interacciones.

Optimización de costos: La gestión de recursos en la nube permite un uso más eficiente del presupuesto tecnológico.

Integración con otras tecnologías de Oracle: Aprovechar servicios como bases de datos y soluciones de analítica integradas dentro de OCI.

Promoción y uso del chatbot en nuevos proyectos

Es fundamental dar a conocer el chatbot a nuevos usuarios y técnicos involucrados en otros proyectos asociados, como PCA, SPA y otras iniciativas de Claro Soluciones y Oracle. Esto podría lograrse mediante:

Capacitaciones dirigidas: Organizar sesiones de formación para familiarizar a los equipos con las funcionalidades del chatbot.

Demostraciones en vivo: Mostrar el potencial del chatbot en eventos o reuniones de proyecto.

Compatibilidad para carga de archivos

Se recomienda habilitar la compatibilidad del chatbot para gestionar archivos como Word, PDF, ejecutables (.exe), archivos comprimidos (.zip) y otros formatos relevantes para mostrar disponibilidad de archivos y formatos al alcance del técnico. Esto facilitará:

Contar con disponibilidad de la información 24/7: Permite que el chatbot cuente con archivos y herramientas en su unidad local para el acceso de la información de los técnicos y usuarios del proyecto Itel.

Gráficas de monitoreo de servicios

Para mejorar el análisis del desempeño, se sugiere que el chatbot incluya la capacidad de generar y mostrar gráficas dinámicas sobre el estado de los servicios. Esto sería particularmente útil para monitorear:

Estado de los servicios de Triara, Venecia y Aprovisionador: Presentar indicadores de disponibilidad, tiempos de respuesta y rendimiento general en el contexto de los servicios del proyecto Itel de Oracle.

Alertas visuales: Generar gráficos que muestren de manera clara cualquier anomalía o degradación en el servicio.

Conclusiones

El desarrollo del chatbot básico en código de Python ha permitido proporcionar respuestas frecuentes, procesar automatizaciones dirigidas y realizar el seguimiento de requerimientos en soporte técnico y mesa de servicios, cumpliendo con el objetivo general del proyecto. Su implementación ha demostrado ser una solución eficiente para optimizar la atención al cliente en la empresa Claro Soluciones dentro del marco del proyecto Itel de Oracle, reduciendo significativamente los tiempos de respuesta y mejorando la gestión de soporte técnico.

En relación con el levantamiento de información de requerimientos, se logró recopilar y documentar de manera efectiva los requerimientos y datos clave sobre los procesos técnicos de la empresa, lo que permitió entrenar al chatbot con información relevante y estructurada, asegurando su precisión en la atención a los usuarios. La recolección de datos incluyó aspectos como procesos internos, configuraciones, y estructuras de código, lo que facilitó una integración sólida con los sistemas existentes.

El diseño del software basado en metodologías ágiles tipo Scrum permitió una gestión estructurada del desarrollo, mejorando la iteración y adaptación a los requerimientos del proyecto. La arquitectura del chatbot se diseñó con una estructura modular, lo que asegura su escalabilidad para futuras mejoras y la integración de nuevos módulos de acuerdo con las necesidades del negocio. Esta flexibilidad facilita la implementación de nuevas funcionalidades sin afectar la estabilidad del sistema.

El análisis comparativo de lenguajes de programación permitió validar que Python era la mejor opción para el desarrollo del chatbot, debido a su flexibilidad, facilidad de integración con bibliotecas de procesamiento de lenguaje natural y su compatibilidad con bases de datos

empresariales. Las pruebas de desempeño confirmaron que el chatbot es capaz de manejar múltiples solicitudes simultáneamente sin comprometer su rendimiento.

El desarrollo e implementación del prototipo del chatbot incorporó técnicas avanzadas de inteligencia artificial para mejorar la precisión de las respuestas, logrando automatizar el 80% de las consultas frecuentes y reducir los tiempos de respuesta en un 60%. Estas mejoras han permitido optimizar los recursos del equipo de soporte técnico, liberando a los técnicos para atender incidentes de mayor complejidad.

Las pruebas y validación del chatbot demostraron su estabilidad y efectividad en entornos reales, identificando posibles mejoras para optimizar su desempeño. Entre los principales hallazgos se destacó la necesidad de fortalecer la seguridad del sistema, implementar medidas de cifrado robustas y reforzar los protocolos de autenticación para garantizar la protección de datos sensibles.

Para maximizar el impacto del chatbot, se recomienda desarrollar funcionalidades predictivas para anticiparse a problemas frecuentes, ampliar su entrenamiento con datos de escenarios reales, optimizar la interfaz de usuario con opciones de personalización y mejorar la capacidad de procesamiento con tecnologías como CUDA. Asimismo, se sugiere expandir la información en su base de datos interna con contenido relevante sobre la empresa, como tramas, promociones, configuraciones y tutoriales paso a paso, garantizando así una experiencia más completa y precisa para los usuarios y técnicos de nivel 1.

Finalmente, el sistema de chatbot no solo optimiza la operación interna de Claro Soluciones, sino que también representa un modelo escalable y replicable para otros sectores empresariales que buscan implementar soluciones basadas en su modelo básico. Su integración con los sistemas existentes, su capacidad de adaptación a nuevas necesidades e interacción con

servicios de mensajería como WhatsApp lo convierten en una herramienta estratégica para la innovación y transformación digital en el proyecto Itel Oracle.

Referencias bibliográficas

Badge Scrum Developer (Arnold Sebastian Beltran Puentes). (n.d.). Retrieved January 12, 2025, from https://insignium.uk/public/index.php/badgeUser/2aa1a553-75e4-4f4f-9a3d-7eac85bab4d3

Badge Scrum Master (Arnold Sebastian Beltran Puentes). (n.d.). Retrieved January 12, 2025, from https://insignium.uk/public/index.php/badgeUser/58008721-af3e-4c07-b95a-e9b1b59db457

Badge Scrum Product Owner (Arnold Sebastian Beltran Puentes). (n.d.). Retrieved January 12, 2025, from https://insignium.uk/public/index.php/badgeUser/22a572b5-5b3e-4ca0-bc3c-97d0c3582868

ChatGPT. (2023). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=ChatGPT&oldid=1164922098

¿Cómo usar Pickle para guardar y cargar variables en Python? – Barcelona Geeks. (n.d.). Retrieved April 29, 2024, from https://barcelonageeks.com/como-usar-pickle-para-guardar-y-cargar-variables-en-python/

¿Cuáles son los beneficios de utilizar la metodología ágil? | AppMaster. (n.d.). Retrieved December 23, 2024, from https://appmaster.io/es/blog/cuales-son-los-beneficios-de-utilizar-la-metodologia-agil

DuckDuckGo. (2025). In *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=DuckDuckGo&oldid=165255767

DuckDuckGo—Your protection, our priority. (n.d.). DuckDuckGo. Retrieved February 8, 2025, from https://duckduckgo.com

El tutorial de Python. (n.d.). Python documentation. Retrieved July 6, 2023, from https://docs.python.org/3/tutorial/index.html

GitHub. (2024). In *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=GitHub&oldid=163856193

GitHub · Build and ship software on a single, collaborative platform. (2024). GitHub. https://github.com/

IA generativa en las empresas: 7 usos en la actualidad. (n.d.). Retrieved December 1, 2024, from https://www.claro.com.co/empresas/noticias-interes/ia-generativa-usos/

Inteligencia artificial. (2024). In *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Inteligencia_artificial&oldid=160718139

Inteligencia Artificial. Casos prácticos con Aprendizaje Profundo—Grupo Editorial RA-MA. (n.d.). Retrieved June 11, 2024, from https://www.ra-ma.es/libro/inteligencia-artificial_139032/, https://www.ra-ma.es/libro/inteligencia-artificial_139032/

Introduction to Python. (n.d.). Retrieved July 6, 2023, from https://www.w3schools.com/python/python_intro.asp

Lasky, J. (20230301). Chatbot. *Salem Press Encyclopedia of Science*. https://research.ebsco.com/c/qcagk4/viewer/html/poa5zs3mib

Lucidchart | Diagramas creados con inteligencia. (n.d.). Retrieved November 11, 2024, from https://www.lucidchart.com/pages/es

Metodología: Qué es, Tipos y Ejemplos. (n.d.). Enciclopedia Significados. Retrieved December 24, 2024, from https://www.significados.com/metodologia/

Metodologías ágiles: ¿Qué son y cuáles son las más utilizadas? - ADEN. (2021, September 21). ADEN Business Magazine. https://www.aden.org/business-magazine/metodologias-agiles/

Metodologías Agiles vs Tradicionales. (n.d.). *POSITIVO*. Retrieved December 24, 2024, from https://www.positivo.pro/blog/metodologias-agiles-vs-tradicionales/

Metodologías tradicionales—Marco Teórico. (n.d.). Retrieved December 24, 2024, from https://1library.co/article/metodolog%C3%ADas-tradicionales-marco-te%C3%B3rico.yjd85kky

MINTIC Colombia—Inicio. (n.d.). MINTIC Colombia. Retrieved June 14, 2023, from http://www.mintic.gov.co/portal/715/w3-channel.html

Openai/openai-python. (2024). [Python]. OpenAI. https://github.com/openai/openai-python (Original work published 2020)

pandas—Python Data Analysis Library. (n.d.). Retrieved July 6, 2023, from https://pandas.pydata.org/

Principales ventajas y limitaciones de las metodologías ágiles. (2016, May 24). OBS Business School. https://www.obsbusiness.school/blog/principales-ventajas-y-limitaciones-de-las-metodologias-agiles

programacionpro.com. (2024, September 16). *Metodologías Ágiles: Beneficios y Ventajas para Mejorar tus Procesos*. ProgramaciónPro. https://programacionpro.com/metodologias-agiles-beneficios-y-ventajas-para-mejorar-tus-procesos/

Python. (n.d.). /clases/1301-expresiones-regulares/11879-python9865/. Retrieved July 6, 2023, from https://platzi.com/clases/1301-expresiones-regulares/11879-python9865/

¿Qué es el procesamiento del lenguaje natural (PLN)? | IBM. (2021, September 23). https://www.ibm.com/mx-es/topics/natural-language-processing

¿Qué es Flask Python? Un breve tutorial sobre este microframework. (2023, March 1). IONOS Digital Guide. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/flask/

¿Qué es un chatbot? | IBM. (n.d.). Retrieved July 5, 2023, from https://www.ibm.com/es-es/topics/chatbots

Qué es Visual Studio Code y qué ventajas ofrece | OpenWebinars. (n.d.). OpenWebinars.net. Retrieved June 11, 2024, from https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/

¿Qué es y para qué sirve el Soporte Técnico Nivel 1? (2020, November 4). Zendesk. https://www.zendesk.com.mx/blog/soporte-tecnico-nivel-1/

Qué significa Random. (n.d.). Significados. Retrieved July 12, 2023, from https://www.significados.com/random/

¿Qué son las redes neuronales convolucionales? | IBM. (2024, May 29). https://www.ibm.com/es-es/topics/convolutional-neural-networks

¿Qué son las redes neuronales recurrentes? | IBM. (2023, December 14). https://www.ibm.com/es-es/topics/recurrent-neural-networks

Ramos De Santis, P. (20240401). Satisfacción del cliente en la logística: Un análisis de chatbots en las empresas líderes de Colombia, Perú y Ecuador. *Retos, Revista de Ciencias Administrativas y Económicas*, *14*(27), 115–130.

Sebastianbel01 (Director). (2022, September 5). *Chatbot para TEAMS con Python Selenium y Pyautogui* [Video recording]. https://www.youtube.com/watch?v=UhQJjcD_sGg

Semana. (2020, April 7). *La startup colombiana que con chatbots busca ayudar a las empresas en medio de la crisis*. Semana.com Últimas Noticias de Colombia y el Mundo. https://www.semana.com/economia/emprendimiento/articulo/la-startup-colombiana-que-con-chatbots-busca-ayudar-a-las-empresas-en-medio-de-la-crisis/202000/

sistemasgeniales. (2020, September 27). *¿Para qué sirve TensorFlow? - SistemasGeniales.com - Páginas web, software y redes sociales*. https://sistemasgeniales.com/2020/09/tensorflow/

Team, T. P. D. (n.d.). *pandas: Powerful data structures for data analysis, time series, and statistics* (Version 2.0.3) [Cython, Python; OS Independent].

Trabajando con JSON - Aprende desarrollo web | MDN. (2022, December 5). https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON

Tutorial NLTK: ¿Qué es la biblioteca NLTK en Python? (2024, March 2). https://www.guru99.com/es/nltk-tutorial.html

Uso de Chatbot en Colombia Creció un 380 % en los Últimos Meses. (n.d.). WebFindYou Colombia. Retrieved June 17, 2024, from https://www.webfindyou.com.co/blog/uso-de-chatbot-en-colombia-crecio-un-380-en-los-ultimos-meses/

Visual Studio Code—Code Editing. Redefined. (n.d.). Retrieved October 16, 2022, from https://code.visualstudio.com/

YouTube. (n.d.). Retrieved April 29, 2024, from https://www.youtube.com/

YouTube. (2024). In *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=YouTube&oldid=159734601