

UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES
“UNIANDES”



FACULTAD	Sistemas Mercantiles
CARRERA	Ingeniería de Software
MODALIDAD	En Línea
TEMA	Aplicación web empresarial para la gestión de Help Desk del área de sistemas de la empresa Imbauto S.A. con inteligencia de negocios
AUTORES	Sebastián Alexander Cedillo Herrera Mario Roberto Farinango Torres Erick Jhonier Guamán Pereira Adrián Adolfo Merlo Arcos Cristian Joel Riofrio Medina
DOCENTES	Ing. Rita Azucena Diaz Vásquez Ing. Diego Paul Palma Rivera Dr. Luis Javier Molina Chalacan Ing. Silvio Amable Machuca Vivar

AMBATO – ECUADOR

Mayo – septiembre 2024

2. PROBLEMA

2.1 Planteamiento del Problema

La empresa Imbauto S.A., dedicada a la comercialización de vehículos, repuestos y accesorios multimarca, así como a la prestación de servicios de taller mecánico y colisiones especializados, enfrenta serios desafíos en la gestión de sus solicitudes de soporte técnico. Actualmente, estas solicitudes se reciben a través de diversos canales como correo electrónico, llamadas telefónicas y WhatsApp, pero no se registran en una herramienta centralizada. Esto ha generado múltiples problemas operativos y de gestión de los soportes.

La ausencia de un registro centralizado dificulta la organización y priorización de las solicitudes. Esto incrementa el riesgo de que algunas solicitudes se pierdan o sean pasadas por alto, afectando negativamente la eficiencia del soporte técnico y la satisfacción de los usuarios. Sin un registro adecuado de los soportes atendidos, resulta complicado verificar si las solicitudes están siendo gestionadas dentro de los tiempos adecuados y con la eficiencia necesaria. Esto impide una correcta evaluación y seguimiento del desempeño del equipo de soporte técnico.

La incapacidad para establecer prioridades claras y gestionar Acuerdos de Nivel de Servicio (SLAs) afecta la capacidad de respuesta ante solicitudes urgentes. Esto puede llevar a retrasos en la resolución de problemas críticos, comprometiendo la operatividad de la empresa y la satisfacción de los usuarios. Además, no existe un sistema formal para notificar a los usuarios sobre el estado y la resolución de sus solicitudes. Esto genera incertidumbre y frustración entre los usuarios, además de dificultar la transparencia y rendición de cuentas en el proceso de soporte.

La empresa carece de datos consolidados sobre el volumen de tickets, tiempos de respuesta inicial, tiempos de resolución, categorías más solicitadas, y otros indicadores clave. Esto impide realizar análisis detallados y tomar decisiones informadas para mejorar el servicio de soporte. Además, no hay una base de conocimiento en línea donde los usuarios puedan consultar soluciones a problemas comunes. Esto incrementa la carga de trabajo del equipo de soporte al tener que atender consultas básicas que podrían ser resueltas por los usuarios de manera autónoma.

Estos problemas afectan directamente la eficiencia operativa del departamento de sistemas de Imbauto S.A., disminuyen la calidad del servicio brindado y aumentan los costos operativos. Por lo tanto, es fundamental desarrollar una aplicación web empresarial para la gestión de Help Desk que integre inteligencia de negocios. Esta solución permitirá centralizar y automatizar la gestión de solicitudes de soporte técnico, mejorar la priorización y seguimiento de tickets, gestionar SLAs de manera efectiva, y generar reportes analíticos para la toma de decisiones estratégicas.

2.2 Situación Conflicto

Imbauto S.A. se encuentra en una situación crítica en lo que respecta a la gestión de su soporte técnico. La gestión de las solicitudes de soporte sin una herramienta centralizada ha llevado a una serie de problemas que afectan tanto al departamento de sistemas como a los usuarios finales. La empresa recibe solicitudes de soporte a través de múltiples canales como correo electrónico, llamadas telefónicas y WhatsApp, lo que dificulta la organización y el seguimiento adecuado de cada caso.

Esta dispersión de canales y la ausencia de un sistema automatizado ha resultado en una gestión desorganizada y propensa a errores. Los técnicos de soporte se ven abrumados por la dificultad de priorización clara y la incapacidad de dar seguimiento a las solicitudes de manera eficiente. Como consecuencia, se experimentan frecuentes retrasos en la atención de problemas críticos, lo que compromete la operatividad de las agencias de Imbauto S.A. en varias ciudades.

La situación se agrava por la imposibilidad de gestionar efectivamente los Acuerdos de Nivel de Servicio (SLAs), lo que lleva a incumplimientos en los tiempos de respuesta y resolución. Sin un registro formal y consolidado, es difícil verificar si los soportes están siendo atendidos dentro de los plazos establecidos, generando desconfianza e insatisfacción entre los usuarios. La carencia de notificaciones formales sobre el estado y la resolución de las solicitudes incrementa la frustración de los usuarios, quienes permanecen en la incertidumbre respecto al progreso de sus casos.

Además, la empresa carece de datos consolidados y métricas clave que permitan

evaluar el rendimiento del equipo de soporte. Sin información sobre el volumen de tickets, tiempos de respuesta, categorías más solicitadas, y otros indicadores, es imposible realizar un análisis profundo y tomar decisiones estratégicas para mejorar el servicio. Esta carencia de visibilidad también impide identificar problemas recurrentes y optimizar los procesos de soporte.

La inexistencia de una base de conocimiento accesible en línea representa otro conflicto significativo. Los usuarios no pueden consultar soluciones a problemas comunes, lo que aumenta innecesariamente la carga de trabajo del equipo de soporte. Los técnicos deben atender consultas básicas que podrían ser resueltas por los usuarios mismos, si tuvieran acceso a una biblioteca de soluciones y guías.

En este contexto, la implementación de una aplicación web empresarial para la gestión de Help Desk se presenta como una solución imprescindible. Esta herramienta permitirá centralizar la gestión de solicitudes, establecer prioridades claras, gestionar SLAs de manera efectiva, y proporcionar un seguimiento detallado y transparente del estado de cada caso. La integración de inteligencia de negocios facilitará la generación de reportes analíticos, permitiendo una toma de decisiones informada y estratégica. Además, la creación de una base de conocimiento accesible y la implementación de un chat bot apoyado con Inteligencia artificial reducirá la carga de trabajo del equipo de soporte, mejorando la eficiencia operativa y la satisfacción de los usuarios.

2.3 Formulación del Problema

¿Cómo implementar un sistema de gestión de tickets que centralice, organice y optimice el proceso de atención de solicitudes de soporte técnico, permitiendo un seguimiento eficaz, la priorización de casos urgentes, el cumplimiento de SLAs, la generación de métricas útiles y el acceso a una base de conocimiento para los usuarios, con el fin de mejorar la calidad del servicio y la satisfacción tanto de los usuarios como del equipo de soporte?

2.4 Objetivos

2.4.1 General

Desarrollar una aplicación web empresarial para la gestión de Help Desk del

área de sistemas de la empresa Imbauto S.A. integrando inteligencia de negocios para la generación de reportes.

2.4.2 Específicos

- Realizar una revisión bibliográfica relaciona con los conceptos clave del proyecto, para establecer un marco teórico que respalde el desarrollo de la aplicación web empresarial para la gestión de Help Desk.
- Utilizar técnicas de recolección de datos como entrevistas y encuestas dirigidas a los principales involucrados, incluyendo al jefe del departamento de sistemas de Imbauto S.A. y los usuarios del sistema de soporte técnico, para identificar necesidades y problemas específicos, así como obtener expectativas y sugerencias para el nuevo sistema de Help Desk.
- Integrar una herramienta de inteligencia de negocios para el análisis de datos y la generación de reportes detallados que permitan monitorear métricas clave como tiempo de respuesta y satisfacción del usuario para facilitar la toma de decisiones estratégicas .
- Implementar un chatbot basado en un modelo de lenguaje de inteligencia artificial (LLM) que proporcione soporte automatizado e instantáneo a los usuarios, facilitando la resolución rápida de consultas comunes y permitiendo al equipo de soporte técnico centrarse en incidencias más complejas y que requieran atención personalizada.

2.5 Justificación e importancia

La implementación de una aplicación web empresarial para la gestión de Help Desk del área de sistemas de Imbauto S.A. es esencial para abordar los problemas actuales, mejorar la operatividad y eficiencia del soporte técnico. La investigación y desarrollo de este sistema permite centralizar y automatizar la gestión de solicitudes, facilitando así la toma de decisiones informadas y oportunas. La integración de inteligencia de negocios permitirá el análisis detallado de datos y la generación de reportes precisos, optimizando la toma de decisiones.

Los resultados de esta investigación contribuyen a mejorar la realidad operativa de Imbauto S.A., permitiendo establecer un control más riguroso sobre los tiempos de

respuesta, la satisfacción del usuario y la eficiencia en la resolución de problemas. Al proporcionar una plataforma centralizada, el departamento de sistemas de Imbauto puede identificar y abordar de manera más eficaz las solicitudes urgentes, mejorando la calidad del servicio brindado.

La utilidad de esta investigación se demuestra en el contexto de Imbauto S.A., donde la implementación del sistema de Help Desk mejora la comunicación con los usuarios, reduce la incertidumbre y aumenta la satisfacción de los usuarios que solicitan soporte, al mismo tiempo que permite al equipo de soporte técnico concentrarse en incidencias más complejas. En resumen, la investigación propuesta ayuda a solventar problemas operativos inmediatos y establece una base sólida para la mejora continua del servicio de soporte técnico del departamento de sistemas de Imbauto S.A.

3. MARCO TEÓRICO

3.1 Antecedentes del Estudio

Los sistemas de Help Desk han emergido como una herramienta crucial para optimizar la gestión de incidencias en el ámbito de las tecnologías de la información (TI). Diversos estudios han explorado la implementación y los beneficios de estas plataformas en diferentes contextos, desde instituciones educativas hasta empresas de servicios y entidades gubernamentales.

Colmenares (2021) resalta la importancia de contar con un sistema Help Desk para gestionar eficientemente las incidencias de TI en microempresas desarrolladoras de software, un aspecto clave para garantizar la continuidad de sus operaciones y la satisfacción de sus clientes. Su investigación se enfoca en la aplicación de la metodología ITIL v3, un marco de referencia ampliamente reconocido en la gestión de servicios de TI, y la automatización de procesos mediante BPM (Business Process Management). A través de este enfoque, se busca reducir significativamente el tiempo de resolución de incidentes, optimizar la asignación de recursos y mejorar la comunicación entre el equipo de soporte y los usuarios, lo que se traduce en una mayor eficiencia y calidad en la atención al cliente.

Vergara (2021) aborda la implementación de un sistema Help Desk bajo plataforma web en la empresa DATASUM S.R.L., con el objetivo de mejorar el proceso de servicio técnico, un aspecto fundamental para garantizar la satisfacción del cliente y la continuidad del negocio. Su estudio detalla los diferentes módulos implementados en el sistema, como el módulo de tickets para registrar los requerimientos de los clientes, el módulo de carga de trabajo para asignar tareas al personal de soporte, el módulo de seguimiento de tareas y el módulo de gestión de soluciones. Además, destaca la importancia de generar reportes para la explotación de la base de datos de casos de éxito, lo que permite un análisis detallado de la información y facilita la toma de decisiones para la mejora continua del servicio.

Stoilova (2021) explora el uso de chatbots de inteligencia artificial (IA) como herramientas innovadoras de servicio al cliente y soporte en diversas industrias. A través de tres estudios de caso en los sectores de turismo, belleza y bibliotecas, la autora demuestra cómo los chatbots pueden automatizar tareas rutinarias, como responder preguntas frecuentes, programar citas y realizar búsquedas en bases de datos. Además, destaca el potencial de los chatbots para proporcionar asistencia instantánea y personalizada las 24 horas del día, los 7 días de la semana, mejorando significativamente la experiencia del cliente y liberando al personal para enfocarse en tareas más complejas y de mayor valor agregado.

Espinoza y Suarez (2021) examinan la influencia del Help Desk en la gestión de incidencias en una entidad estatal del sector educativo, un contexto en el que la continuidad operativa de los servicios digitales es fundamental para el cumplimiento de la misión institucional. Su investigación se basa en el marco de referencia ITIL y utiliza un enfoque cuantitativo para analizar la relación entre el uso del sistema Help Desk y la eficiencia en la gestión de incidencias. Los resultados de su estudio demuestran una influencia positiva y significativa del Help Desk en todas las etapas del proceso de gestión de incidencias, desde el registro y clasificación hasta la investigación, diagnóstico y cierre de las solicitudes.

Pacasia Torres y Cardenas Molina (2021) presentan la implementación de un sistema Help Desk basado en ITIL e ISO/IEC 20000 en la Contraloría Departamental de Boyacá, una entidad gubernamental en la que la gestión eficiente de los recursos

de TI es esencial para garantizar la transparencia y la rendición de cuentas. Su estudio se enfoca en la sistematización del inventario de hardware, software y muebles, así como en la gestión y solución de problemas informáticos a través de la creación y asignación de incidencias. La implementación de estas herramientas y la adopción de las mejores prácticas de ITIL e ISO/IEC 20000 permiten a la Contraloría mejorar el control y la eficiencia en la gestión de sus recursos de TI, garantizando un servicio de calidad a sus usuarios internos y externos.

3.2 Definiciones conceptuales

- **Gestión de Tickets.-** es una herramienta integral para gestionar solicitudes e incidentes de clientes de manera eficiente y organizada. Permite recibir consultas de diferentes canales, distribuirlas entre agentes, establecer prioridades y automatizar respuestas, agilizando así el proceso de atención al cliente. (Zendesk, Zendesk, 2024)
- **SLA (Service Level Agreement).-** Un acuerdo de nivel de servicio (SLA) es un contrato de proveedor de subcontratación y tecnología que define el nivel de servicio que dicho proveedor se compromete a brindar al cliente. Describe métricas como tiempo de actividad, tiempo de entrega, tiempo de respuesta y tiempo de resolución. (aws, aws, 2023)

*Ilustración 1
Imagen de SLA*



Nota: Service Level Agreements: Definition, Levels & Importance to VoIP. Corey McCraw (2023). Fit Small Business <https://fitsmallbusiness.com/service-level-agreements/>

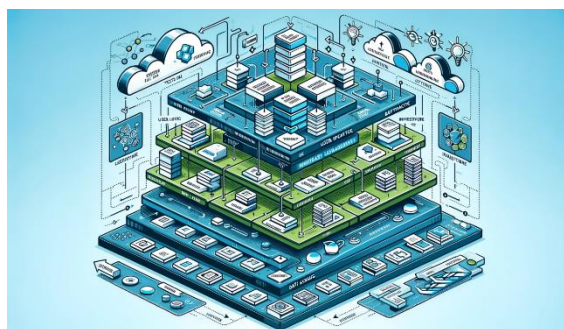
- **Métricas de Desempeño.-** Indicadores de ejecución de acciones que pueden conducir a un resultado. Las métricas de desempeño se suelen utilizar para realizar el seguimiento y la evaluación del estado de un proceso. (Angloamerica,

2020)

- **Base de Conocimientos.-** La base de conocimiento es una biblioteca en línea que cuenta con varios formatos de contenido. Su objetivo es informar y compartir conocimiento entre clientes y empleados de una empresa. Al mismo tiempo, el autoservicio también está diseñado para agilizar la atención al cliente y dinamizar el acceso al conocimiento interno. (Zendesk, Zendesk, 2024)
- **Diagramas UML.-** Lenguaje de modelado estándar que se utiliza para visualizar un plan arquitectónico para elementos como actividades, procesos de negocio y esquemas de base de datos (IBM, 2021)
- **Patrones de Diseño.-** son elementos reutilizables creados para resolver problemas comunes. Es decir que con su aplicación y utilización podremos corregir diferentes problemas que presenta nuestro código de una manera segura, estable y testeada por cientos de programadores de todo el mundo. (CrafCode, 2021)
- **Arquitectura en Capas. -** La arquitectura en capas es un modelo de diseño de software, cuya base es la separación de las diferentes funcionalidades del sistema en capas o niveles, donde cada capa se encarga de un conjunto de tareas específicas y se comunica con los niveles adyacentes mediante interfaces bien definidas. (Durán, 2023)

Ilustración 2

Imagen de Arquitectura en Capas



Nota: Ventajas de la Arquitectura en Capas. Emmanuel Palacio Gaviria(2023).
Linkedin <https://www.linkedin.com/pulse/ventajas-de-la-arquitectura-en-capas-creando-sistemas-emmanuel-gybne/>

- **Microservicios. -** Los microservicios son tanto un estilo de arquitectura como un

modo de programar software. Con los microservicios, las aplicaciones se dividen en sus elementos más pequeños e independientes entre sí. A diferencia del enfoque tradicional y monolítico de las aplicaciones, en el que todo se compila en una sola pieza, los microservicios son elementos independientes que funcionan en conjunto para llevar a cabo las mismas tareas. (Hat, 2023)

- **API (Interfaz de Programación de Aplicaciones).**- es un conjunto de funciones y procedimientos que permite integrar sistemas, permitiendo que sus funcionalidades puedan ser reutilizadas por otras aplicaciones o software. (SYDLE, SUDLE, 2023)
- **Dashboard.**- Es una herramienta de gestión de la información que monitoriza, analiza y muestra de manera visual los indicadores clave de desempeño (KPI), métricas y datos fundamentales para hacer un seguimiento del estado de una empresa, un departamento, una campaña o un proceso específico. (Ortiz, 2023)
- **ETL (Extract, Transform, Load).**- Es un sistema que tiene la capacidad de leer diferentes formatos de archivo y tipos de datos, y transportarlos de un entorno a otro. El ETL sirve para consolidar datos de diferentes fuentes y entregarlos a un nuevo entorno. Además, se puede. (Silva, 2021)
- **Almacén de Datos (Data Warehouse).**- Almacén de datos, como también se le conoce, es un sistema de gestión de datos diseñado para habilitar y soportar actividades empresariales. Su capacidad analítica permite a las organizaciones obtener valiosos datos de su negocio en tiempo real y mejorar la toma de decisiones. (Esan, 2020)
- **OLAP (Online Analytical Processing).**- Es un método informático que permite a los usuarios extraer y consultar datos de manera fácil y selectiva para analizarlos desde diferentes puntos de vista. Las consultas de inteligencia empresarial OLAP a menudo ayudan en el análisis de tendencias, informes financieros, previsión de ventas, presupuestos y otros propósitos de planificación. (TechTarget, 2019)
- **Big Data.**- formado por conjuntos de datos de mayor tamaño y más complejos, especialmente procedentes de nuevas fuentes de datos. Estos conjuntos de datos son tan voluminosos que el software de procesamiento de datos convencional sencillamente no puede gestionarlos. Sin embargo, estos volúmenes masivos de datos pueden utilizarse para abordar problemas

empresariales que antes no hubiera sido posible solucionar. (Tiao, 2024)

Ilustración 3
Representación gráfica de Big Data



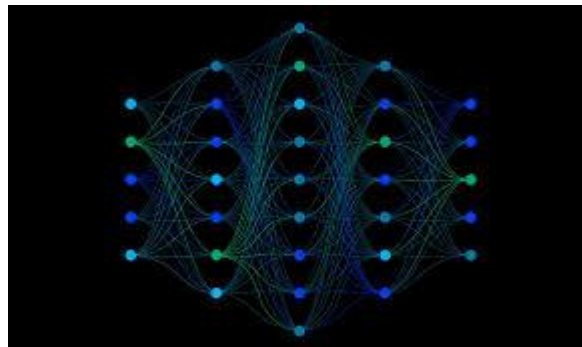
Nota: Big Data BBVA (2011). bbva.com/es/innovacion/el-mundo-laboral-del-big-data-que-esperan-los-expertos-y-que-necesitan-las-empresas/

- **Planificación Estratégica.-** es un proceso que consiste en determinar cuáles son los objetivos de una empresa o una institución y qué plan que se debe implementar para alcanzarlos. Es un tipo de planificación que busca indicar cuál es el mejor camino para lograr las metas que una organización o una persona se han propuesto. (Etecé, 2024)
- **Gestión del Cambio.-** es un área de estudio de gestión que tiene como objetivo ayudar a las organizaciones a adaptarse constantemente a los cambios del mercado. Tenemos que hablar de ella. Después de todo, el cambio es necesario, aunque difícil. (SYDLE, SYDLE, 2022)
- **Cadena de Suministro.-** cadena de abastecimiento o supply chain es un conjunto de elementos que se suceden de forma lógica para que una empresa desarrolle un producto o servicio. Esto incluye todas las actividades y recursos, desde la obtención de la materia prima hasta la entrega del producto final al consumidor. (Element, 2023)
- **Indicadores Clave de Desempeño (KPI).-** Un KPI, sigla que proviene de la frase en inglés key performance indicator (indicador clave de rendimiento), es una métrica cuantitativa que muestra cómo tu equipo o empresa progresa hacia tus objetivos empresariales más importantes. (Martins, 2023)
- **Análisis FODA (SWOT Analysis).-** El análisis SWOT es una poderosa herramienta de planificación que ayuda en la toma de decisiones. La sigla SWOT

es una abreviatura de las palabras Strengths, Weaknesses, Opportunities y Threats, que significan Fortalezas, Debilidades, Oportunidades y Amenazas (de ahí la sigla DAFO que se utiliza en español). (HostGator, 2023)

- **Aprendizaje Automático (Machine Learning).**- El machine learning es la ciencia de desarrollo de algoritmos y modelos estadísticos que utilizan los sistemas de computación con el fin de llevar a cabo tareas sin instrucciones explícitas, en vez de basarse en patrones e inferencias. (aws, aws, 2020)
- **Redes Neuronales Artificiales.**- Están basadas en el funcionamiento de las redes de neuronas biológicas. Las neuronas que todos tenemos en nuestro cerebro están compuestas de dendritas, el soma y el axón: Las dendritas se encargan de captar los impulsos nerviosos que emiten otras neuronas. Estos impulsos, se procesan en el soma y se transmiten a través del axón que emite un impulso nervioso hacia las neuronas contiguas. (xerida, 2019)

Ilustración 4
Imagen de Redes neuronales



Nota: RN. Ana Rivera(2018) Futuro Eléctrico. <https://futuroelectrico.com/redes-neuronales-inteligencia-artificial/>

- **Procesamiento de Lenguaje Natural (NLP).**- Es el campo de conocimiento de la Inteligencia Artificial que se ocupa de la investigar la manera de comunicar las máquinas con las personas mediante el uso de lenguas naturales, como el español, el inglés o el chino. (MORENO, 2020)
- **Chatbot.**- Un chatbot es un software diseñado para simular conversaciones humanas. Puede interactuar con usuarios a través de texto o voz en diversas plataformas digitales. Su complejidad varía desde respuestas predefinidas hasta sistemas avanzados que utilizan inteligencia artificial para entender y responder a una amplia gama de preguntas (IBM, n.d.).

- **Chatbots basado en reglas.-** Estos chatbots siguen un conjunto predefinido de reglas y flujos de conversación. Son ideales para responder preguntas frecuentes y realizar tareas simples. Sin embargo, su capacidad de respuesta es limitada y no pueden manejar consultas complejas o fuera de su programación.
- **Chatbots de aprendizaje automático o inteligentes.-** Estos chatbots utilizan inteligencia artificial y procesamiento del lenguaje natural para comprender y responder a una amplia gama de consultas. Aprenden de las interacciones con los usuarios y mejoran su capacidad de respuesta con el tiempo. Son capaces de manejar consultas complejas, mantener conversaciones naturales y realizar tareas más sofisticadas (Aunoa, 2020).
- **ChatGPT.-** Es un modelo de lenguaje de inteligencia artificial desarrollado por OpenAI. Está entrenado en una gran cantidad de datos de texto para generar respuestas coherentes y relevantes a una amplia gama de preguntas y solicitudes. Puede realizar tareas como responder preguntas, traducir idiomas, generar texto creativo y resumir información (OpenAI, 2022).
- **LLM.-** Un modelo de lenguaje grande (LLM) es un tipo de modelo de aprendizaje profundo que ha sido entrenado en cantidades masivas de datos de texto para comprender y generar lenguaje humano de manera similar a como lo haría una persona. Estos modelos pueden realizar diversas tareas de procesamiento del lenguaje natural (NLP), como traducción de idiomas, resumen de textos, respuesta a preguntas y generación de contenido creativo (IBM, n.d.).
- **Base de datos vectorial.-** Una base de datos vectorial es un tipo de base de datos diseñada para almacenar, gestionar y consultar datos en forma de vectores. A diferencia de las bases de datos tradicionales que almacenan datos en tablas y filas, las bases de datos vectoriales utilizan representaciones matemáticas de alta dimensión para almacenar información, lo que permite realizar búsquedas eficientes basadas en la similitud entre vectores (Amazon Web Services (AWS), n.d.).
- **Embeddings.-** En el contexto del procesamiento del lenguaje natural (PLN) y la inteligencia artificial, un embedding es una representación matemática de palabras, frases o documentos en forma de vectores numéricos densos. Estos vectores capturan el significado semántico y las relaciones entre las palabras, permitiendo que los modelos de aprendizaje automático comprendan y procesen

el lenguaje de manera más efectiva (Jurafsky & Martin, 2020).

- **Búsqueda semántica.-** La búsqueda semántica es una técnica utilizada en los motores de búsqueda que va más allá de la simple coincidencia de palabras clave. Se enfoca en comprender el significado y la intención detrás de una consulta para proporcionar resultados más relevantes y precisos. Para lograrlo, utiliza el procesamiento del lenguaje natural (PLN) y el análisis del contexto para interpretar las palabras y frases en relación con otros términos y conceptos (Pecánek, 2020).
- **Algoritmos Genéticos.-** En informática, se llaman algoritmos el conjunto de instrucciones sistemáticas y previamente definidas que se utilizan para realizar una determinada tarea. Estas instrucciones están ordenadas y acotadas a manera de pasos a seguir para alcanzar un objetivo. (Farrobbial, 2020)
- **Visión por Computadora.-** La visión por computadora o visión computarizada hace referencia a un grupo de tecnologías o herramientas que permiten a los equipos captar imágenes del mundo real, procesarlas y generar información a través de ellas (análisis). (EDSRobotics, 2022)
- **Diagrama caso de uso.-** Los diagramas son diseños geométricos que se realizan con el objetivo de representar gráficamente ideas, procesos, soluciones, mecanismos o fenómenos para facilitar su comprensión. (internet, 2019)
- **Diagrama de secuencia.-** Un diagrama de secuencia se define como un tipo de diagrama UML (Lenguaje Unificado de Modelado) utilizado en ingeniería de software y diseño de sistemas para visualizar las interacciones y la comunicación entre varios componentes u objetos dentro de un sistema. (Vanzant, 2023)
- **Diagrama de clases. -** Un diagrama de clases es una herramienta para comunicar el diseño de un programa que se creó para orientar objetos y que permite modelar relaciones entre diferentes entidades. (Okdiario, 2018)
- **Base de datos transaccional. -** Son bases de datos que tiene como fin el envío y recepción de datos a gran velocidad. Están destinadas generalmente al entorno de análisis de calidad, datos de producción e industrial, y su objetivo principal es asegurar las transacciones dentro de una base de datos relacional o, en caso de que no se puedan asegurar, revertirlas, de manera que evitan que las transacciones queden incompletas, es decir, o se realiza la transacción o no pasa nada (vuelve al estado original). (Ayudaley, 2022)

Ilustración 9
Imagen de BDT



Nota: Bases de Datos transaccionales. Tecnologías de la información (2020)
<https://www.tecnologias-informacion.com/transaccionales.html>

3.3 Preguntas directrices o generativas

1. ¿Qué conceptos clave deben ser revisados bibliográficamente para establecer un marco teórico sólido que respalde el desarrollo de la aplicación web empresarial para la gestión de Help Desk?
2. ¿Cuáles son los problemas específicos más críticos identificados a través de entrevistas y encuestas realizadas al jefe del departamento de sistemas y a los usuarios del sistema de soporte técnico?
3. ¿Cómo se puede integrar una herramienta de inteligencia de negocios para el análisis de datos y la generación de reportes detallados que permitan monitorear métricas clave como tiempo de respuesta y satisfacción del usuario, facilitando la toma de decisiones estratégicas?
4. ¿De qué manera un chatbot basado en un modelo de lenguaje de inteligencia artificial (LLM) puede proporcionar soporte automatizado e instantáneo a los usuarios, facilitando la resolución rápida de consultas comunes y permitiendo al equipo de soporte técnico centrarse en incidencias más complejas?

4. METODOLOGÍA

4.1 Diseño de la investigación

La investigación se diseñó como un estudio mixto, combinando métodos cualitativos y cuantitativos para obtener una comprensión integral del problema y desarrollar una

solución efectiva. La metodología cualitativa incluyó la realización de entrevistas semiestructuradas al jefe del departamento de sistemas de Imbauto S.A., con el objetivo de identificar en profundidad las necesidades, problemas y expectativas relacionadas con el soporte técnico.

Por otro lado, la metodología cuantitativa se basó en la aplicación de cuestionarios tipo Likert a los 110 usuarios del sistema de soporte técnico, con el propósito de obtener datos estadísticos sobre sus experiencias, niveles de satisfacción y sugerencias para el nuevo sistema de Help Desk. Estos datos cuantitativos permitieron complementar la información obtenida en las entrevistas y proporcionar una base sólida para el análisis y desarrollo del sistema.

Además, se realizó una revisión bibliográfica exhaustiva de la literatura y documentos académicos relacionados con los conceptos clave del proyecto, como inteligencia de negocios, inteligencia artificial y gestión de Help Desk. Esta revisión bibliográfica permitió fundamentar teóricamente el desarrollo de la aplicación web y las tecnologías a utilizar.

4.2 Modalidad y tipo de la investigación

La modalidad de la investigación fue aplicada, ya que se enfocó en desarrollar una solución práctica para un problema específico dentro de Imbauto S.A. La investigación aplicada se caracterizó por la implementación de una aplicación web empresarial destinada a mejorar la gestión del Help Desk del área de sistemas de la empresa.

El tipo de investigación se clasificó como de campo y descriptiva. La investigación de campo se utilizó para identificar y comprender los problemas actuales en la gestión del soporte técnico, así como las necesidades y expectativas de los usuarios y del jefe del departamento de sistemas. A través de entrevistas y encuestas, se recolectaron datos que proporcionaron una visión clara y detallada de los desafíos existentes y las áreas que requerían mejoras.

Por otro lado, la investigación descriptiva se empleó para detallar las características y necesidades del sistema de Help Desk propuesto. Esta fase de la investigación

permitió describir con precisión los procesos actuales, los flujos de trabajo, y las métricas clave necesarias para el monitoreo y la mejora continua del servicio de soporte técnico. La integración de inteligencia de negocios y la implementación de un chatbot basado en inteligencia artificial fueron descritas de manera detallada para asegurar que la solución desarrollada cumpliera con los requisitos y expectativas identificados en la fase exploratoria.

Método Inductivo

El enfoque inductivo se basa en el análisis de observaciones y evidencias específicas para formular conclusiones generales o principios más amplios. Este método es fundamental para partir de situaciones particulares o casos individuales y deducir patrones o tendencias aplicables a un contexto más amplio (Hernández, Fernandez, & Baptista, 2014). Este método permitió derivar conclusiones generales a partir de observaciones específicas obtenidas a través de entrevistas y encuestas. Al analizar los datos recolectados de manera inductiva, se identificaron patrones y tendencias que informaron el diseño y desarrollo de la aplicación web y sus funcionalidades. El método inductivo facilitó la comprensión de las necesidades y problemas específicos de los usuarios y del departamento de sistemas, lo que permitió desarrollar una solución más precisa y efectiva.

4.3 Técnicas e instrumentos de la investigación

Para llevar a cabo esta investigación se utilizaron técnicas e instrumentos específicos de recolección de datos, los cuales permitieron obtener información detallada y relevante sobre las necesidades y problemas en la gestión de soporte técnico en Imbauto S.A. A continuación, se describen las técnicas e instrumentos empleados:

Entrevista Semiestructurada

Se realizó una entrevista semiestructurada al jefe del departamento de sistemas de Imbauto S.A. Esta entrevista tuvo como objetivo identificar las necesidades específicas, problemas recurrentes y expectativas relacionadas con el soporte técnico.

Cuestionario Tipo Likert

Se aplicó un cuestionario tipo Likert a los 110 usuarios del sistema de soporte

técnico. Este cuestionario fue diseñado para recolectar datos cuantitativos sobre las experiencias, niveles de satisfacción y sugerencias de los usuarios respecto al soporte técnico recibido.

4.4 Población y muestra

Población:

La población de estudio para esta investigación está compuesta por dos grupos principales dentro de Imbauto S.A.: el jefe del departamento de sistemas y los usuarios del sistema de soporte técnico.

Integrantes de la investigación	Cantidad	Herramienta de investigación
Jefe del departamento de Sistemas	1	Entrevista semiestructurada
Usuarios del sistema de soporte técnico	110	Cuestionario tipo Likert

Muestra:

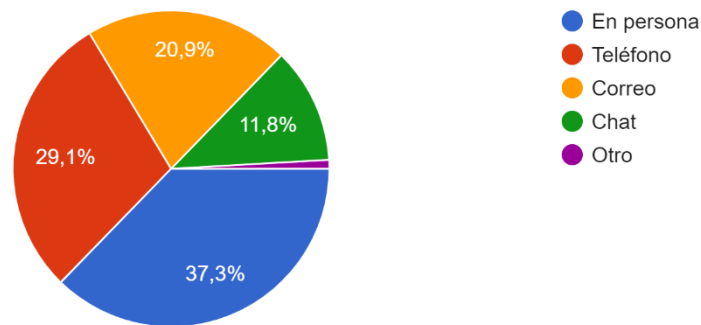
Debido a que la diferencia entre la población total y el tamaño de la muestra calculado es pequeña, se decidió trabajar con la población completa de 110 usuarios. Esta decisión permitió obtener una cobertura completa y precisa en la recolección de datos, mejorando la calidad y validez de los resultados del estudio.

5. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

5.1 Presentación y discusión de resultados

¿Cómo contactaste con el personal del departamento de Sistemas para solicitar soporte?

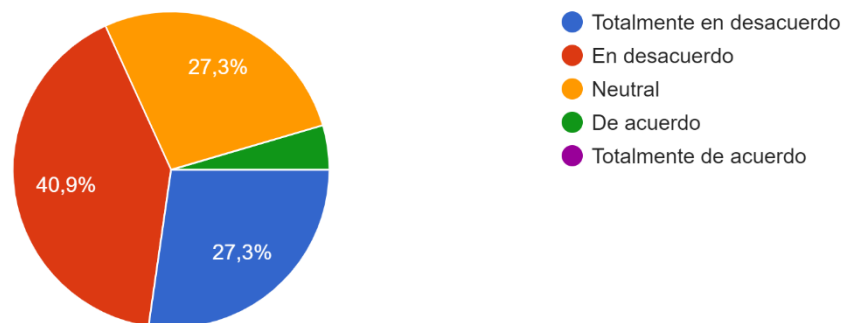
110 respuestas



Los canales de contacto más utilizados para solicitar soporte técnico en Imbauto S.A. son el correo electrónico y el teléfono. Es crucial mantener y mejorar estos canales, mientras que se analiza la necesidad de fortalecer los canales menos utilizados. La implementación de una base de conocimiento en línea podría ser una herramienta valiosa para reducir la carga de trabajo del soporte técnico y mejorar la satisfacción general de los usuarios.

Estoy satisfecho con el tiempo de respuesta del equipo de soporte técnico a mis solicitudes.

110 respuestas

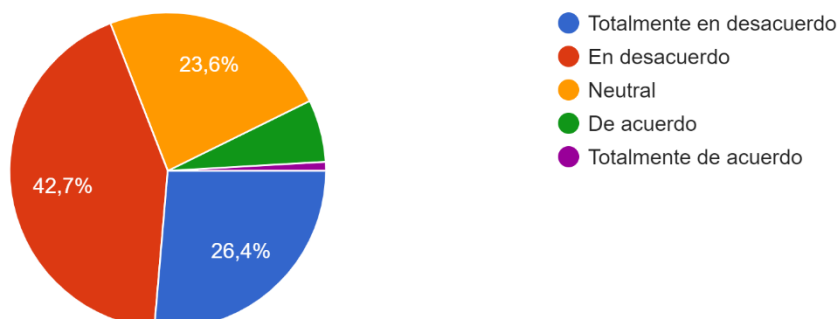


Los resultados de la encuesta muestran que la mayoría de los participantes no está satisfecha con el tiempo de respuesta del equipo de soporte técnico. Solo una minoría está de acuerdo con el tiempo de respuesta, lo que indica una posible área de mejora significativa para el equipo de soporte.

reflejan que la mayoría de los participantes considera que el equipo de soporte técnico no

El equipo de soporte técnico maneja adecuadamente la priorización de solicitudes urgentes.

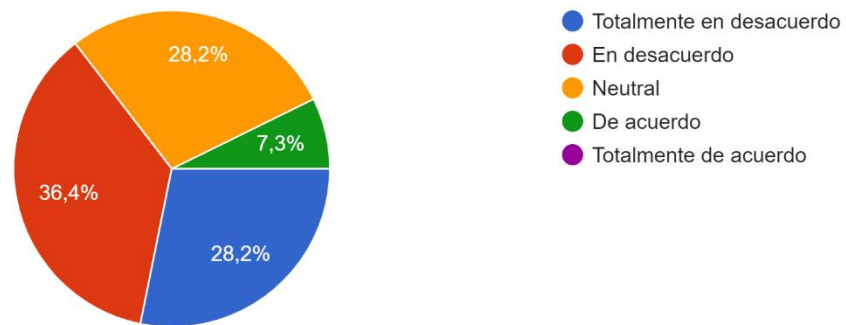
110 respuestas



Reflejan que la mayoría de los participantes considera que el equipo de soporte técnico no maneja adecuadamente la priorización de solicitudes urgentes. La proporción de encuestados satisfechos es considerablemente menor, lo que señala una posible área de mejora en la gestión de prioridades.

Los problemas que reporto al soporte técnico son resueltos de manera efectiva.

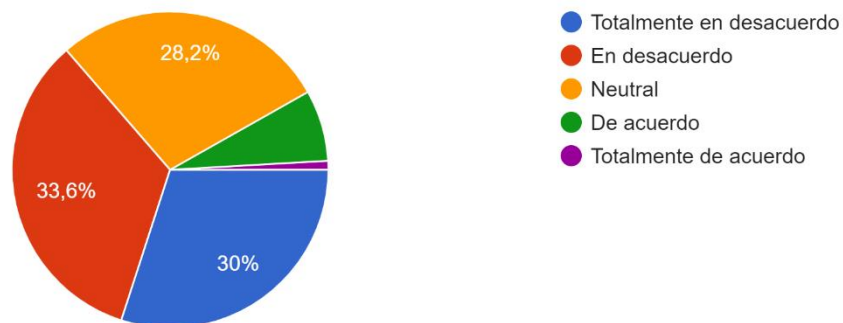
110 respuestas



Muestran que una mayoría significativa de los participantes no está satisfecha con la efectividad del soporte técnico en la resolución de problemas reportados. Una proporción menor de encuestados se siente satisfecha, indicando que hay margen para mejorar en este aspecto.

Recibo actualizaciones periódicas sobre el estado de mis solicitudes de soporte.

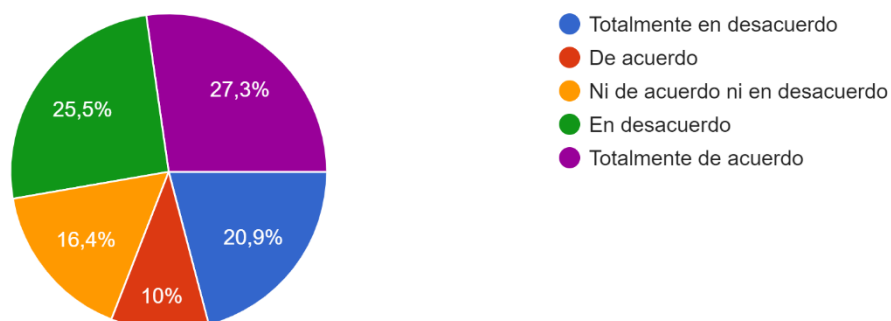
110 respuestas



La mayoría de los usuarios no están satisfechos con la eficacia del soporte técnico para resolver problemas. Es necesario mejorar la comunicación, el tiempo de respuesta y la capacidad de resolución de problemas para aumentar la satisfacción del cliente.

En general, estoy satisfecho con el servicio de soporte técnico que recibo

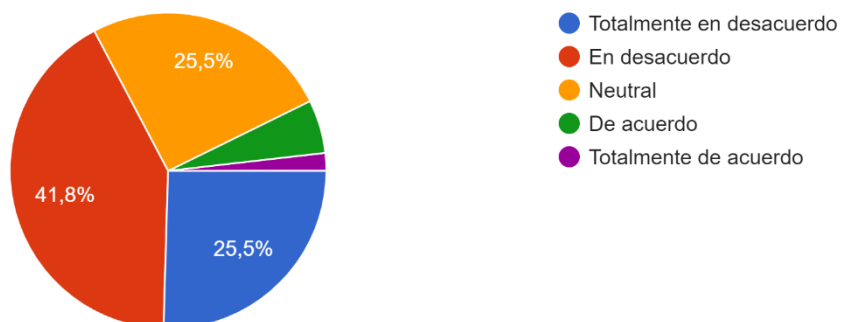
110 respuestas



La mayoría de usuarios (58,2%) no están satisfechos con el soporte técnico. Se necesita mejorar la comunicación, tiempo de respuesta y resolución de problemas.

Considero que el soporte técnico tiene suficiente información y métricas para mejorar continuamente sus servicios.

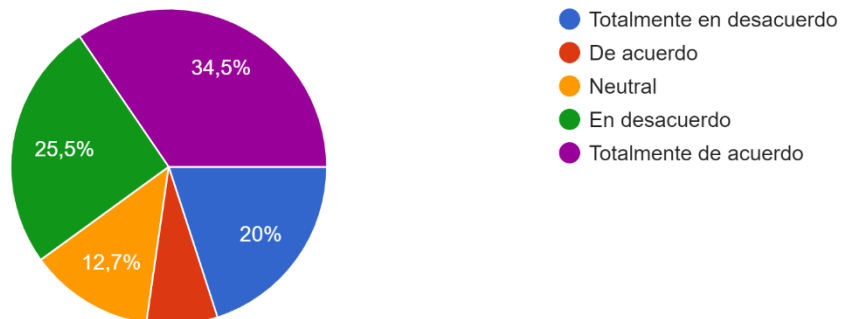
110 respuestas



Soporte técnico insatisfactorio: tenemos una gran mayoría de usuarios insatisfechos. Mejorar comunicación, tiempo de respuesta y resolución de problemas.

Me sería útil tener acceso a una base de conocimiento en línea para resolver problemas comunes por mi cuenta.

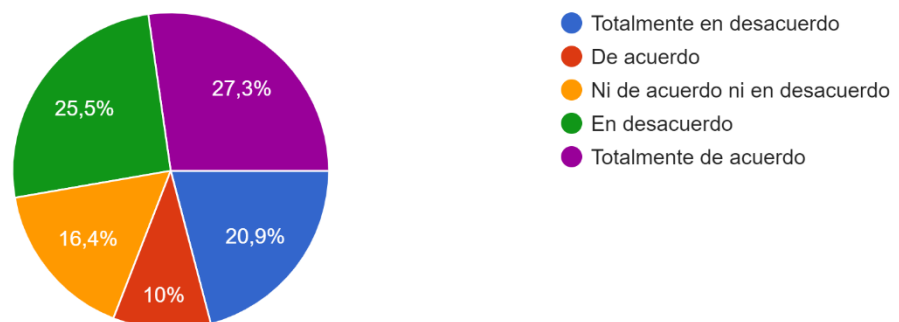
110 respuestas



El dibujo muestra que la mayoría de los usuarios no están satisfechos con la eficacia del soporte técnico en la resolución de problemas.

En general, estoy satisfecho con el servicio de soporte técnico que recibo

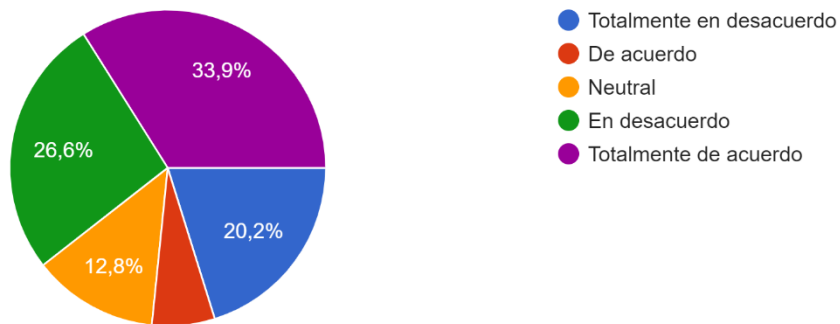
110 respuestas



El cuadro muestra que la mayoría de los usuarios no están satisfechos con la eficacia del soporte técnico en la resolución de problemas. Esto es un indicador preocupante de que la empresa necesita mejorar su servicio de atención al cliente.

Creo que la implementación de una aplicación web para la gestión de soporte técnico mejoraría significativamente la eficiencia del servicio.

109 respuestas



La mayoría de los usuarios no están satisfechos con la eficacia del soporte técnico en la resolución de problemas. Esto es un indicador preocupante de que la empresa necesita mejorar su servicio de atención al cliente.

5.2 Respuestas a las preguntas directrices

1. ¿Qué conceptos clave deben ser revisados bibliográficamente para establecer un marco teórico sólido que respalde el desarrollo de la aplicación web empresarial para la gestión de Help Desk?
 - Para establecer un marco teórico sólido, se revisaron bibliográficamente los conceptos clave relacionados con la gestión de Help Desk, inteligencia de negocios, y modelos de lenguaje de inteligencia artificial (LLM). La revisión incluyó estudios sobre mejores prácticas en la gestión de soporte técnico, tecnologías de inteligencia artificial aplicadas en servicios de soporte, y herramientas de inteligencia de negocios para el análisis y generación de reportes.
2. ¿Cuáles son los problemas específicos más críticos identificados a través de entrevistas y encuestas realizadas al jefe del departamento de sistemas y a los usuarios del sistema de soporte técnico?

- Las entrevistas y encuestas revelaron varios problemas críticos en la gestión actual del soporte técnico. Entre los principales problemas identificados se encuentra la dispersión de las solicitudes de soporte técnico a través de múltiples canales (correo electrónico, llamadas telefónicas y WhatsApp), lo que dificulta la priorización y el seguimiento efectivo de las solicitudes.
3. ¿Cómo se puede integrar una herramienta de inteligencia de negocios para el análisis de datos y la generación de reportes detallados que permitan monitorear métricas clave como tiempo de respuesta y satisfacción del usuario, facilitando la toma de decisiones estratégicas?
- Para integrar una herramienta de inteligencia de negocios, se propuso el desarrollo de dashboards que permitan monitorear métricas clave como el tiempo de respuesta, la satisfacción del usuario, y la eficiencia en la resolución de problemas. Estos dashboards se basarán en los datos recolectados a través de la nueva aplicación de Help Desk, permitiendo visualizar en tiempo real el desempeño del soporte técnico. La generación de reportes detallados facilitará la identificación de áreas de mejora y la toma de decisiones estratégicas, contribuyendo a la optimización continua del servicio de soporte técnico.
4. ¿De qué manera un chatbot basado en un modelo de lenguaje de inteligencia artificial (LLM) puede proporcionar soporte automatizado e instantáneo a los usuarios, facilitando la resolución rápida de consultas comunes y permitiendo al equipo de soporte técnico centrarse en incidencias más complejas?
- Un chatbot basado en un modelo de lenguaje de inteligencia artificial (LLM) puede proporcionar soporte automatizado e instantáneo respondiendo a consultas comunes de los usuarios en tiempo real. Este chatbot será capaz de entender y procesar preguntas frecuentes, proporcionando soluciones inmediatas sin necesidad de intervención

humana. Esto no solo mejora la eficiencia del soporte técnico al reducir el volumen de consultas manejadas por los técnicos, sino que también permite al equipo de soporte centrarse en incidencias más complejas que requieren atención personalizada. Además, el chatbot puede ser integrado en la aplicación web, ofreciendo una experiencia de usuario coherente y accesible en cualquier momento.

6. PROPUESTA

7. CONCLUSIONES-RECOMENDACIONES

7.1. Conclusiones

- La ausencia de un sistema centralizado ha dificultado la gestión eficiente de las solicitudes de soporte técnico, lo que ha resultado en retrasos y una gestión ineficiente.
- La integración de herramientas de inteligencia de negocios permitirá un análisis detallado de los datos de soporte técnico, mejorando la toma de decisiones y la eficiencia operativa.
- La implementación de un chatbot inteligente reducirá la carga de trabajo del equipo de soporte y mejorará la satisfacción del usuario al proporcionar respuestas rápidas a consultas comunes.

7.2. Recomendaciones

- Se recomienda implementar un sistema de gestión de tickets centralizado y basado en la web para mejorar la eficiencia del soporte técnico en Imbauto S.A.
- Es fundamental proporcionar capacitación continua al personal en el uso del nuevo sistema y en mejores prácticas de soporte técnico.
- Establecer un proceso continuo de monitoreo y evaluación del rendimiento del sistema de HelpDesk para identificar áreas de mejora y realizar ajustes necesarios.
- Considerar la ampliación de las capacidades del chatbot para incluir soporte en más áreas y temas técnicos, mejorando aún más la eficiencia y la satisfacción del usuario.

8. REFERENCIAS BIBLIOGRAFICAS

- Amazon Web Services (AWS). (s.f.). ¿Qué es una base de datos vectorial?: <https://aws.amazon.com/es/what-is/vector-databases/>
- Angloamerica. (2020). *Angloamerica*. <https://socialway.angloamerican.com/es-es/toolkit/governance/governance-inner/social-performance-resourcing/social-performance-team-performance-metrics>
- Aunoa. (4 de Diciembre de 2020). ¿Qué tipos de chatbot existen?: <https://aunoa.ai/blog/que-tipos-de-chatbot-existen/>
- aws. (2020). aws. <https://aws.amazon.com/es/what-is/machine-learning/#:~:text=El%20machine%20learning%20es%20la,basarse%20en%20patrones%20e%20inferencias.>
- aws. (2023). aws. [https://aws.amazon.com/es/what-is/service-level-agreement/#:~:text=para%20sus%20servicios%3F,%C2%BFQu%C3%A9%20es%20un%20acuerdo%20de%20nivel%20de%20servicio%20\(SLA%2C,compromete%20a%20brindar%20al%20cliente.](https://aws.amazon.com/es/what-is/service-level-agreement/#:~:text=para%20sus%20servicios%3F,%C2%BFQu%C3%A9%20es%20un%20acuerdo%20de%20nivel%20de%20servicio%20(SLA%2C,compromete%20a%20brindar%20al%20cliente.)
- Ayudaley. (2 de Noviembre de 2022). *Ayudaley*. <https://ayudaleyprotecciondatos.es/bases-de-datos/transaccionales/>
- Colmenares, N. (2021). *Modelo de gestión de incidencias aplicando herramientas de software en el área de Help Desk basado en BPM en una microempresa desarrolladora de Software*. Universidad Señor de Sipán, Pimentel. <https://hdl.handle.net/20.500.12802/9209>
- CrafCode. (2 de Julio de 2021). *CrafCode*. [https://craft-code.com/que-son-los-patrones-de-diseno/#:~:text=Los%20patrones%20de%20dise%C3%B1o%20\(design,programadores%20de%20todo%20el%20mundo.](https://craft-code.com/que-son-los-patrones-de-diseno/#:~:text=Los%20patrones%20de%20dise%C3%B1o%20(design,programadores%20de%20todo%20el%20mundo.)
- Durán, M. (12 de abril de 2023). *Huspot*. <https://blog.hubspot.es/website/que-es-arquitectura-en-capas#:~:text=La%20arquitectura%20en%20capas%20es,adyacentes%20mediante%20interfaces%20bien%20definidas.>
- EDSRobotics. (31 de enero de 2022). *EDSRobotics*. [https://www.edsrobotics.com/blog/vision-computador-que-es/#:~:text=visi%C3%B3n%20por%20computador,%C2%BFQu%C3%A9%20es%20la%20visi%C3%B3n%20por%20computador%3F,trav%C3%A9s%20de%20ellas%20\(an%C3%A1lisis\).](https://www.edsrobotics.com/blog/vision-computador-que-es/#:~:text=visi%C3%B3n%20por%20computador,%C2%BFQu%C3%A9%20es%20la%20visi%C3%B3n%20por%20computador%3F,trav%C3%A9s%20de%20ellas%20(an%C3%A1lisis).)
- Element. (25 de Abril de 2023). *Element*. <https://www.elementfleet.com.mx/recursos/blog/cadena-de-suministro-que-es-que-tipos-existen-y-por-que-es-importante-para-las-empresas#:~:text=La%20cadena%20de%20suministro%2C%20cadena,del%20producto%20final%20al%20consumidor.>
- Esan, C. (01 de Julio de 2020). *Cesan*. <https://www.esan.edu.pe/conexion-esan/data-warehouse-pasos-para-construirlo-con-exito#:~:text=El%20data%20warehouse%20o%20almac%C3%A9n,mejorar%20la%20toma%20de%20decisiones.>
- Espinoza, H., & Suarez, R. (2021). *Help Desk y la gestión e incidencias en una entidad del estado en el sector educativo 2021 [Tesis para optar el título profesional de Ingeniero de Sistemas Empresariales]*. Universidad Científica del Sur, Lima. <https://doi.org/https://doi.org/10.21142/tl.2021.2182>
- Etecé, E. (6 de Abril de 2024). *Concepto*. <https://concepto.de/planeacion-estrategica/>
- Farrobia. (2020). *Farrobia*. <https://www.ferrovial.com/es/stem/algoritmos/#:~:text=En%20inform%C3%A1tica%2C%20se%20llaman%20algoritmos,seguir%20para%20alcanzar%20un%20objetivo.>
- Hat, R. (7 de febrero de 2023). *Red Hat*. <https://www.redhat.com/es/topics/microservices>
- Hernández, R., Fernandez, C., & Baptista, P. (2014). *Metodología de la investigación*. McGraw-Hill Education.
- HostGator. (05 de 07 de 2023). *HostGator*. <https://hostgator.mx/blog/analisis-swot-para-tu-negocio/>
- IBM. (28 de 02 de 2021). *IBM*. <https://www.ibm.com/docs/es/iis/11.5?topic=types-unified-modeling-language-uml-model>
- IBM. (s.f.). ¿Qué es un chatbot? <https://www.ibm.com/es-es/topics/chatbots>
- IBM. (s.f.). ¿Qué son los grandes modelos de lenguaje (LLM)? <https://www.ibm.com/mx->

es/topics/large-language-models

internet, L. u. (23 de Julio de 2019). *La universidad de internet*.
<https://colombia.unir.net/actualidad-unir/tipos-de-diagramas/#:~:text=Los%20diagramas%20son%20dise%C3%B1os%20geom%C3%A9tricos,se%20traduce%20como%20'esquema'>.

Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing* (Tercera ed.). Pearson.

Martins, J. (16 de Agosto de 2023). *asana*. <https://asana.com/es/resources/key-performance-indicator-kpi>

MORENO, A. (2020). *Instituto de ingeniería del conocimiento*. :
<https://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/>

Okdiario. (9 de Noviembre de 2018). *OKDIARIO*. <https://okdiario.com/curiosidades/que-diagrama-clases-3323710#:~:text=Un%20diagrama%20de%20clases%20es,tiene%20y%20mensajes%20que%20entiende>.

OpenAI. (30 de Noviembre de 2022). *ChatGPT*. <https://openai.com/index/chatgpt/>

Ortiz, D. (24 de noviembre de 2023). *Cyberclick*. <https://www.cyberclick.es/numerical-blog/que-es-un-dashboard#:~:text=Un%20dashboard%20es%20una%20herramienta,campa%C3%B1a%20o%20un%20proceso%20espec%C3%ADfico>.

Pacasia Torres, K., & Cardenas Molina, J. (2021). *Implementación de un sistema Help Desk basado en Itil e ISO/IEC 20000 para sistematizar inventario de la contraloría departamental de Boyacá, mediante el uso del software libre GLPI Y OCS INVENTORY*. Universidad Santo Tomás. <http://hdl.handle.net/11634/37788>

Pecánek, M. (24 de Julio de 2020). *Ahrefs Blog*. ¿Qué es la búsqueda semántica? Cómo impacta el SEO: <https://ahrefs.com/blog/es/busqueda-semantica/>

Silva, D. d. (11 de febrero de 2021). *Zendesk*. <https://www.zendesk.com.mx/blog/etl-que-es/>

Stoilova, E. (2021). AI chatbots as a customer service and support tool. *ROBONOMICS: The Journal of the Automated Economy*, 2(21).
<https://www.journal.robonomics.science/index.php/rj/article/view/21>

SYDLE. (19 de 12 de 2022). *SYDLE*. <https://www.sydle.com/es/blog/gestion-del-cambio-60364298da4d0968095ad321>

SYDLE. (26 de 09 de 2023). *SUDLE*. <https://www.sydle.com/es/blog/api-6214f68876950e47761c40e7>

TechTarget. (2019). *ComputerWeekly*. <https://www.computerweekly.com/es/definicion/OLAP-o-procesamiento-analitico-en-linea>

Tiao, S. (11 de MARZO de 2024). *OCI*. <https://www.oracle.com/mx/big-data/what-is-big-data/#:~:text=El%20t%C3%A9rmino%20%E2%80%99Cbig%20data%E2%80%9D%20abarc%20a,como%20%E2%80%99Clas%20tres%20V%E2%80%9D>.

Vanzant, P. (3 de noviembre de 2023). *IDEAS CALE*. <https://ideascale.com/es/blogs/que-es-el-diagrama-secuencia/#:~:text=Un%20diagrama%20de%20secuencia%20se,objetos%20dentro%20de%20un%20sistema>.

Vergara, Z. (2021). *Sistema HelpDesk bajo plataforma web para la mejora del proceso de servicio técnico de la empresa Datasum S. R. L. [Trabajo de suficiencia profesional, Universidad Privada del Norte]*. Repositorio de la Universidad Privada del Norte.
<https://hdl.handle.net/11537/30159>

xerida. (16 de Septiembre de 2019). *xerida*. <https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i>

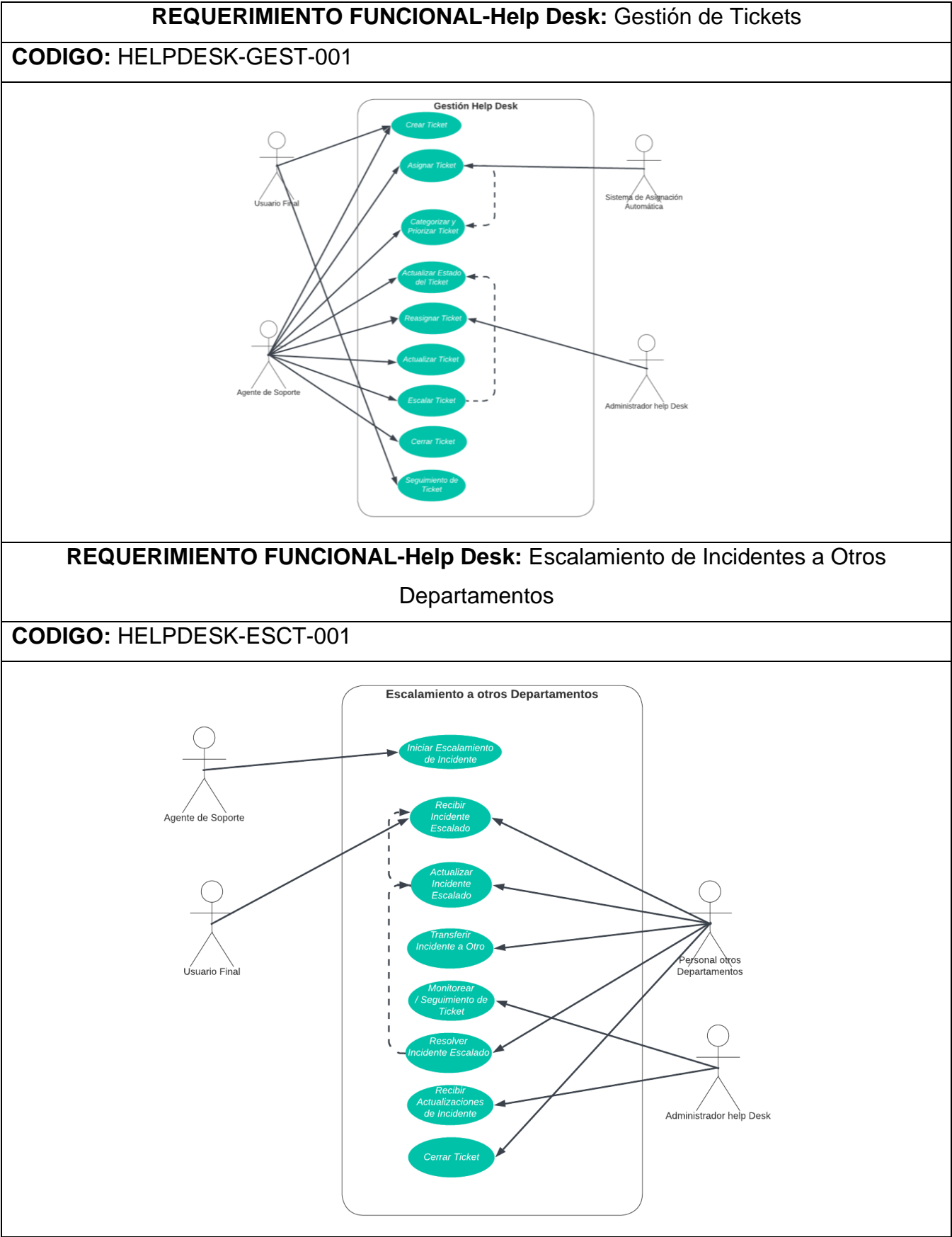
Zendesk. (23 de febrero de 2024). *Zendesk*. <https://www.zendesk.com.mx/blog/sistema-de-gestion-de-tickets/>

Zendesk. (de enero de 2024). *Zendesk*. <https://www.zendesk.com.mx/blog/que-es-base-de-conocimiento/>

9.1. DIAGRAMA UML – CLASES

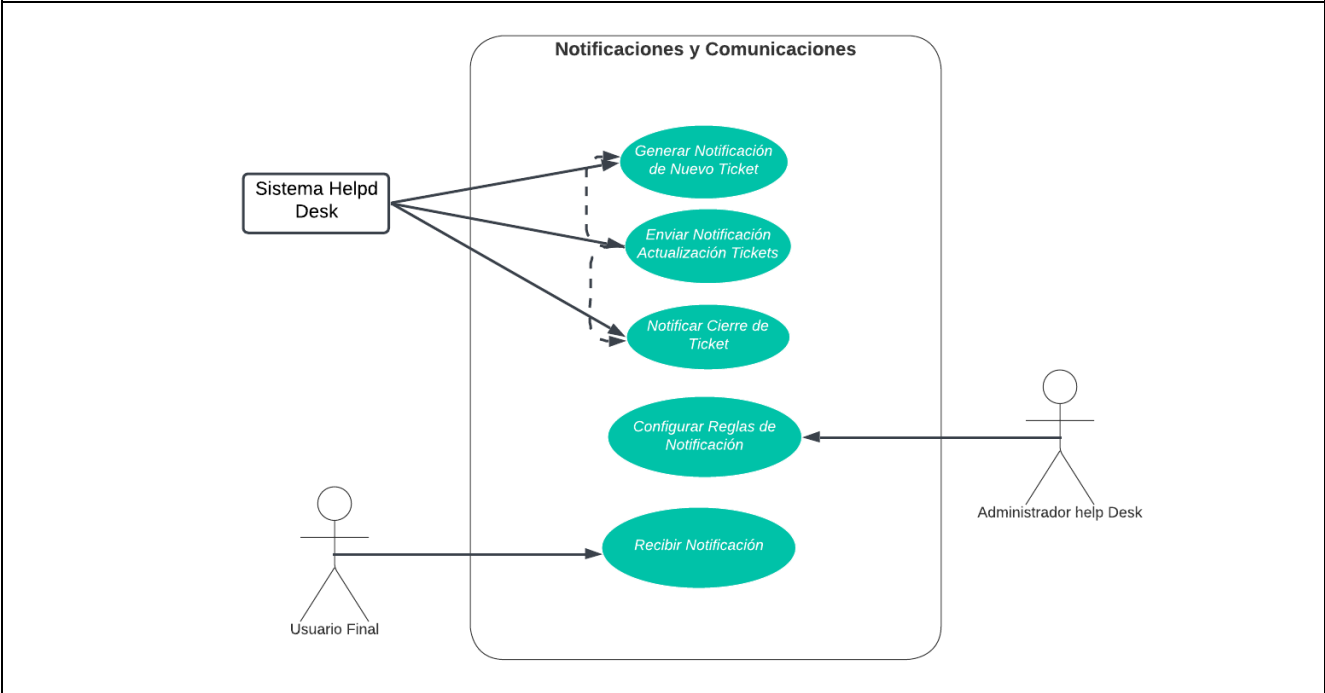


9.2. DIGRAMA UML – CASOS DE USO



REQUERIMIENTO FUNCIONAL - Help Desk: Notificaciones y Comunicaciones

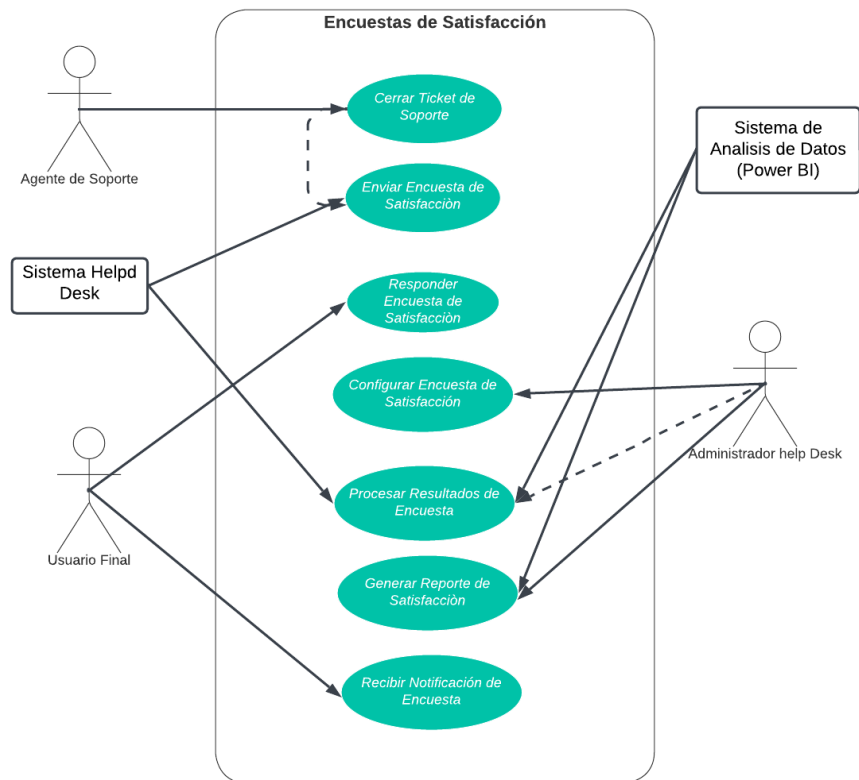
CODIGO: HELPDESK-NOTT-001



REQUERIMIENTO FUNCIONAL - Help Desk: Encuestas de satisfacción

CODIGO: HELPDESK-ENCT-001

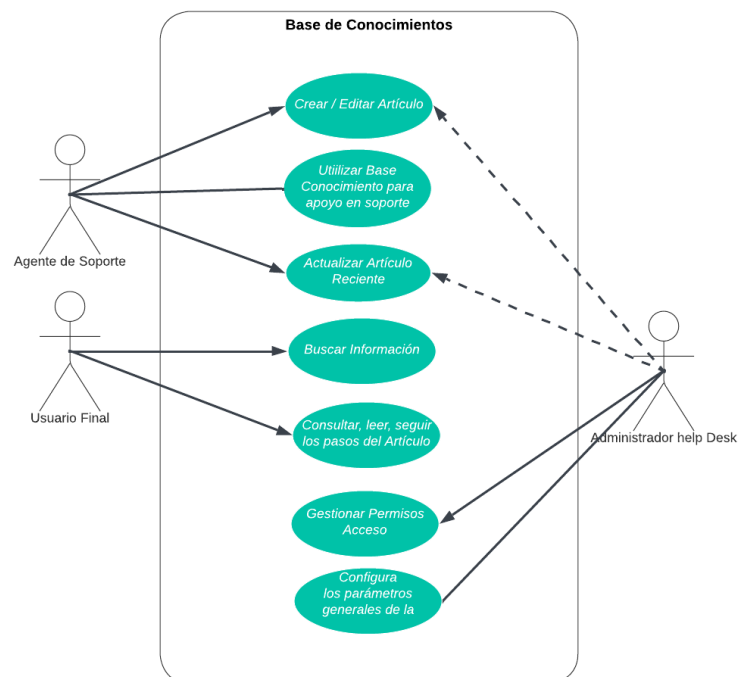
DIAGRAMA DE CASO DE USO:



REQUERIMIENTO FUNCIONAL - Help Desk: Base de Conocimientos

CODIGO: HELPDESK-BAST-001

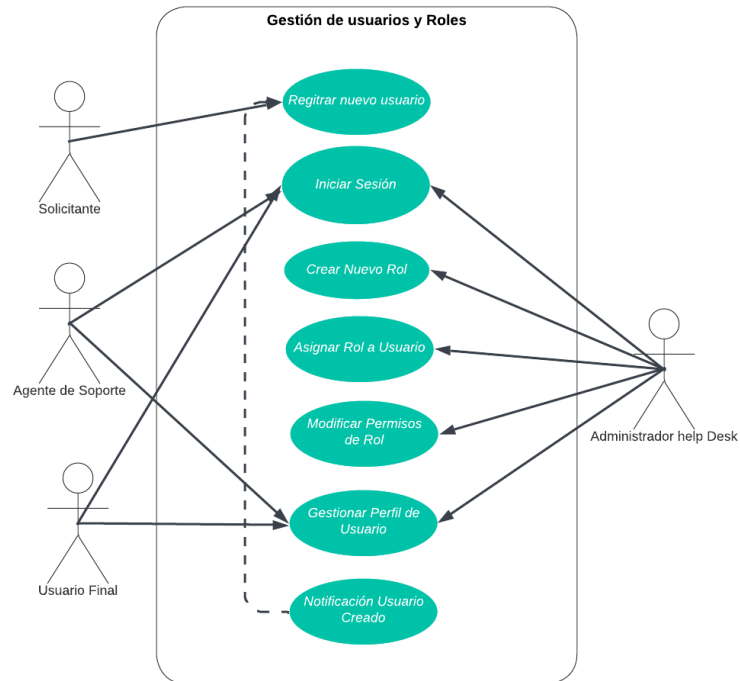
DIAGRAMA DE CASO DE USO:



REQUERIMIENTO FUNCIONAL - Help Desk: Gestión de usuarios y Roles

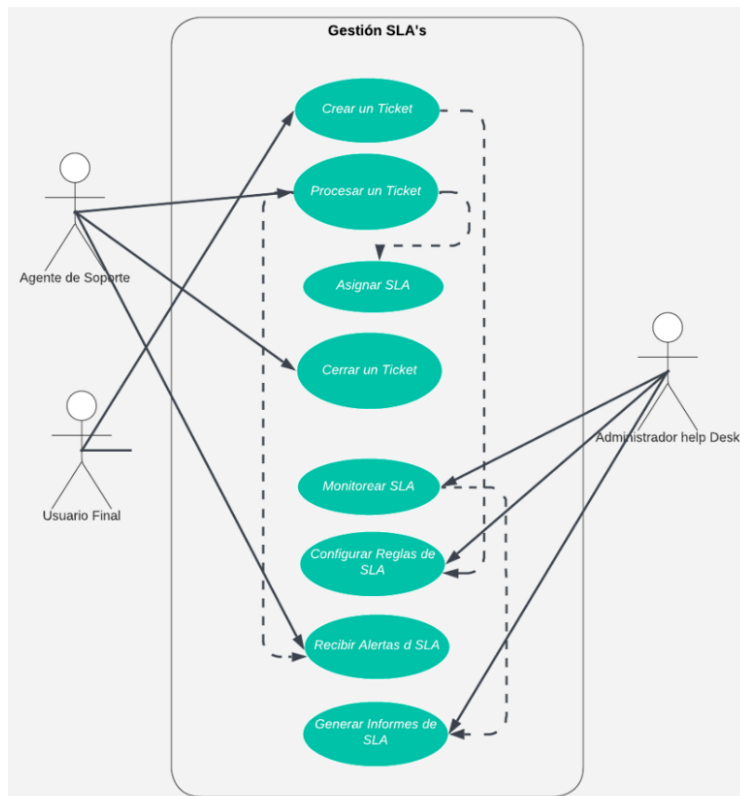
CODIGO: HELPDESK-GEUT-001

DIAGRAMA DE CASO DE USO:



REQUERIMIENTO FUNCIONAL - Help Desk: Gestión SLA's

CODIGO: HELPDESK-GEST-001



REQUERIMIENTO FUNCIONAL - Help Desk: Paneles Analíticos e Informes.

CODIGO: HELPDESK-PANT-001

DIAGRAMA DE CASO DE USO:

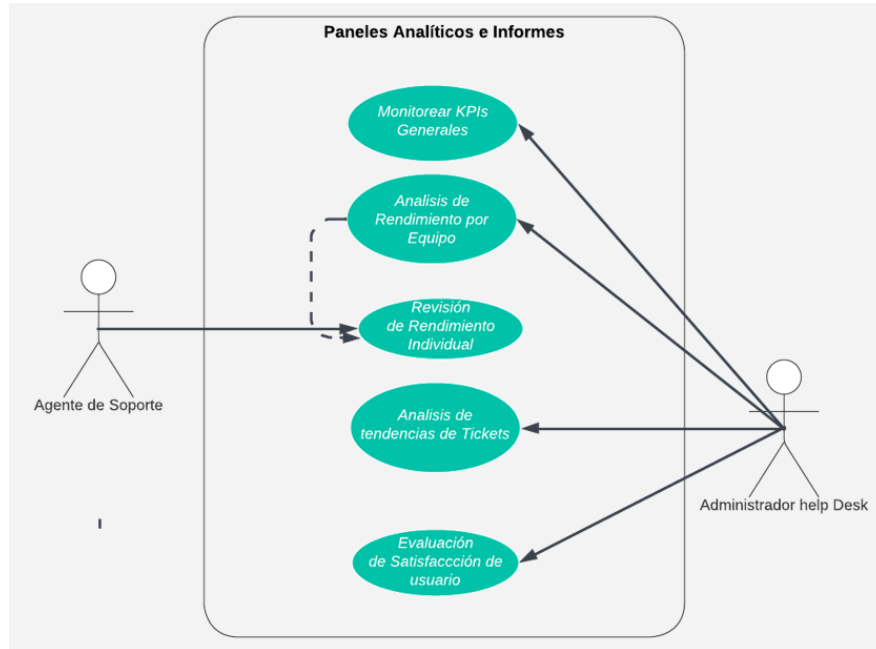
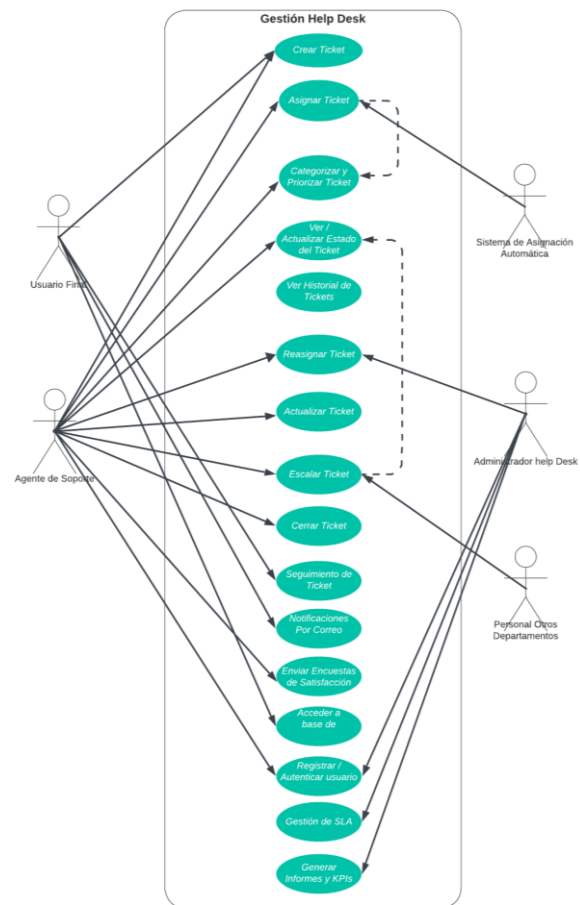
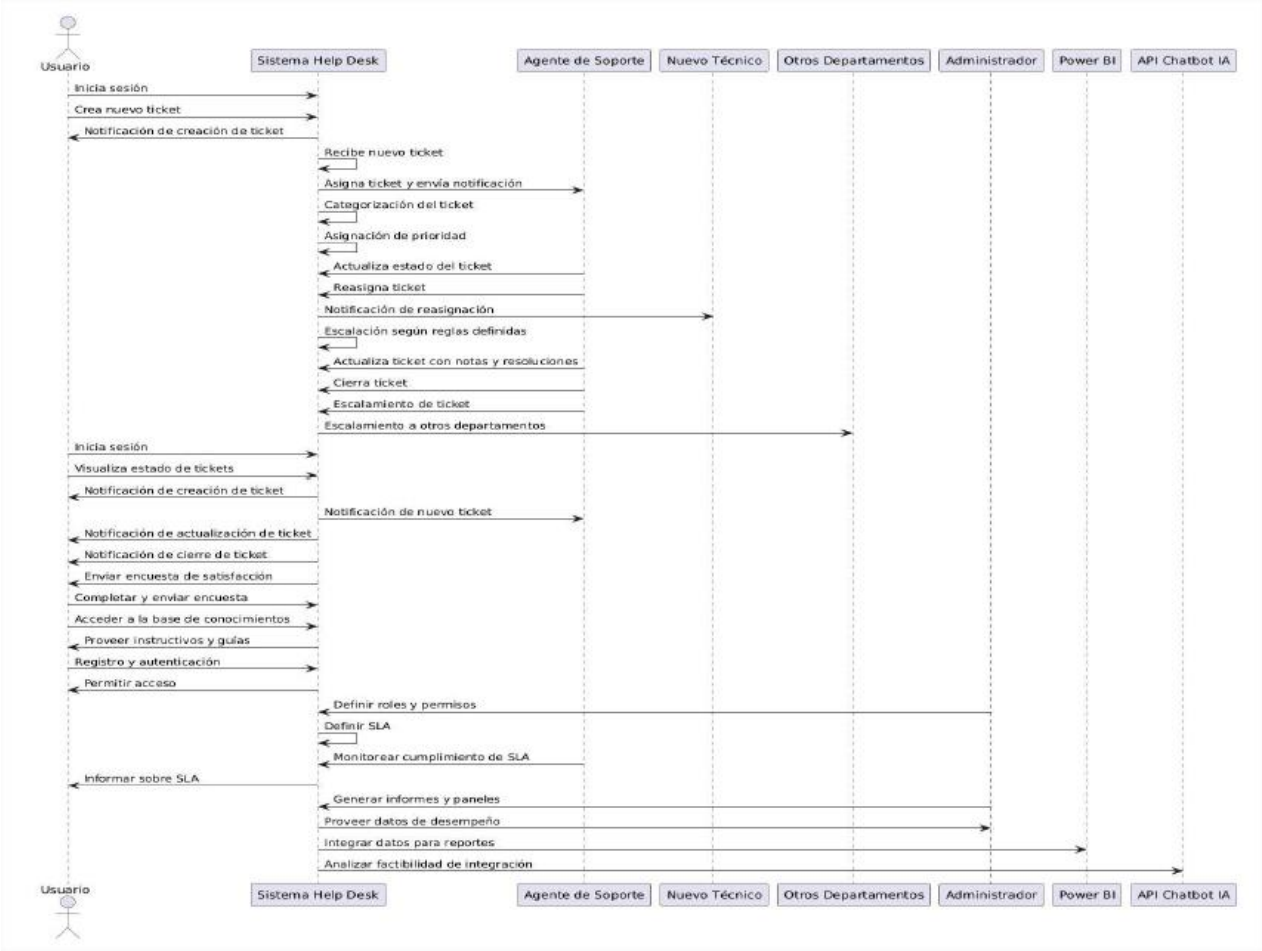


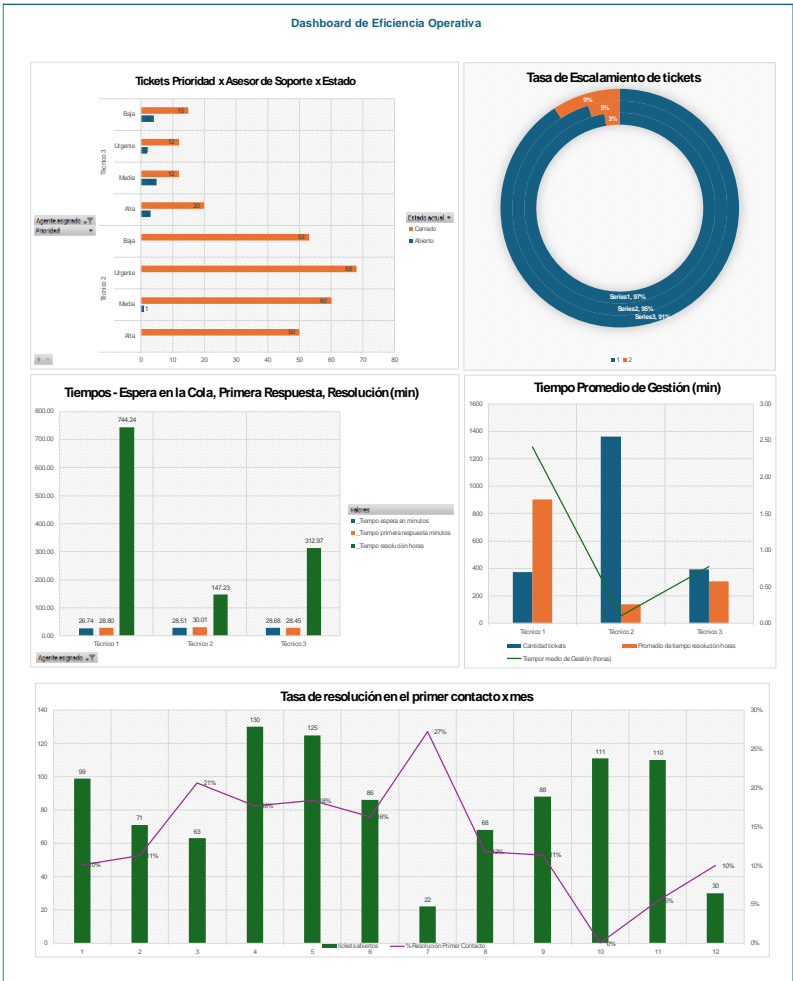
DIAGRAMA DE CASO DE USO GESTION HELP DESK



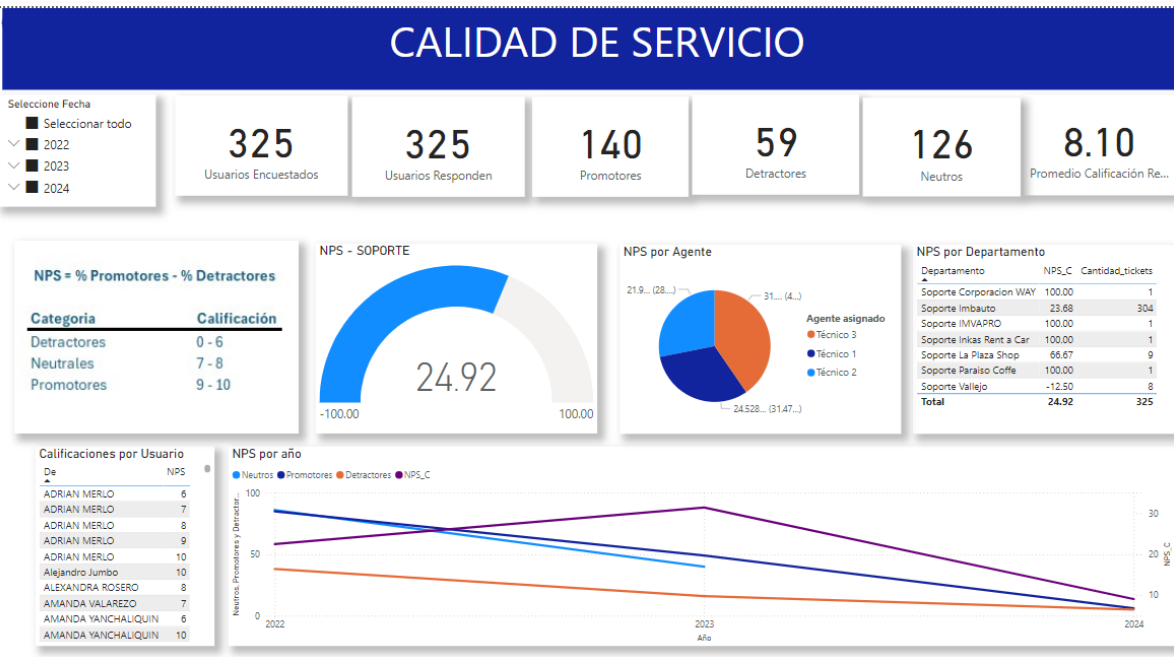
9.3. DIAGRAMA UML - SECUENCIA



9.4. DASHBOARD



9.6. CUADROS DE MANDO



9.7. DESARROLLO DEL CHATBOT IA

ASISTENTE INTELIGENTE

Los modelos de lenguaje como ChatGPT son entrenados en un conjunto de datos limitado y no tienen acceso a información en tiempo real o a información específica del usuario, por lo cual se ha decidido complementar en la “Aplicación web empresarial para la gestión de Help Desk del área de sistemas de la empresa Imbauto S.A. con inteligencia de negocios” con un Chatbot que brinde soporte en problema pequeños y fáciles de solucionar, para lo cual este bot dispondrá de información específica de soporte y reparación de equipos y paquetes informáticos de Imbauto.

EJEMPLO 1:

ChatBot

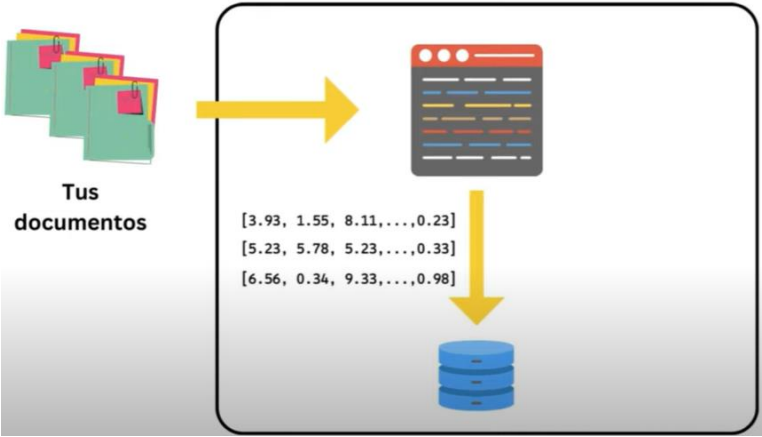
Por lo expuesto se ha desarrollado un Chatbot que permita usar nuestros propios datos, más útil e informativo usando la API de OpenAI, ChromaDb, Sveltekit y la biblioteca de Hugging Face Transformers; para el desarrollo se ha utilizado paquetes de Python como Flask, Flask_cors, Vite; para lo cual se ha procedido con la instalación de los paquetes mencionados con los siguientes pasos:

- `npm install -D vite`
- `npm install`
- `pip install --upgrade chroma`
- `pip install openai`
- `pip install flask_cors`
- `python -m pip install flask`
- `npm init vite@latest`
- `pip list`

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
1:00:27 p. m. [vite] page reload src/routes/bot/+page.svelte
1:56:02 p. m. [vite] hmr update /src/routes/+page.svelte, /node_modules/@skeletonlabs/skeleton/src/lib/styles/all.css
1:56:02 p. m. [vite] hmr update /src/routes/bot/+page.svelte, /node_modules/@skeletonlabs/skeleton/src/lib/styles/all.css
1:56:02 p. m. [vite] hmr update /src/routes/bot/+page.svelte, /node_modules/@skeletonlabs/skeleton/src/lib/styles/all.css
PS C:\Users\rober\docsbot\webapp> pip list
Package Version
-----
annotated-types 0.7.0
anyio 4.4.0
asgiref 3.8.1
backoff 2.2.1
bcrypt 4.1.3
blinker 1.8.2
build 1.2.1
cachetools 5.3.3
certifi 2024.7.4
charset-normalizer 3.3.2
Chroma 0.2.0
```

Para entrenarle al Chatbot será necesario cargar información con archivos markdown al bot los cuales será convertidos a embeddings y posteriormente almacenados en la base de datos ChromaDb que es una base de datos vectorial que permite buscar similitudes. Y de igual manera cuando se proceda con una consulta o búsqueda de respuestas (pregunta al bot) a través de la interfaz web, el bot convertirá esta consulta a embeddings y usará la búsqueda semántica para encontrar los embeddings que sean más relevantes para la consulta del usuario.

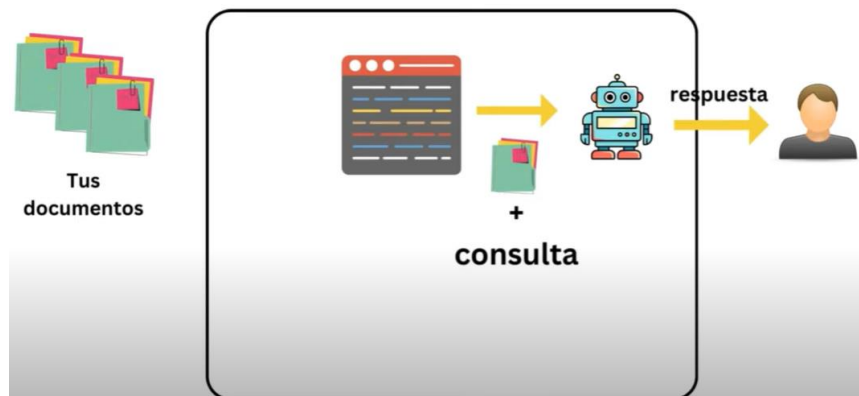
Proceso de carga de documentos:



Proceso de consulta al bot



Proceso de respuesta al usuario con la información entregada del bot.



PROCESO DE DESARROLLO

Convertir datos a embeddings: Para que un chatbot pueda usar tus datos, primero debes convertirlos en embeddings. Los embeddings son representaciones numéricas de texto que capturan el significado del texto. Es importante mencionar que el modelo del lenguaje utilizado debe tener conocimiento y entrenamiento del lenguaje natural por tal motivo se utiliza la API de OPENAI para lo cual se ha procedido a generar un API Key en la plataforma <https://platform.openai.com/api-keys>.

API keys

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

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

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI will automatically revoke your API key if it is leaked publicly.

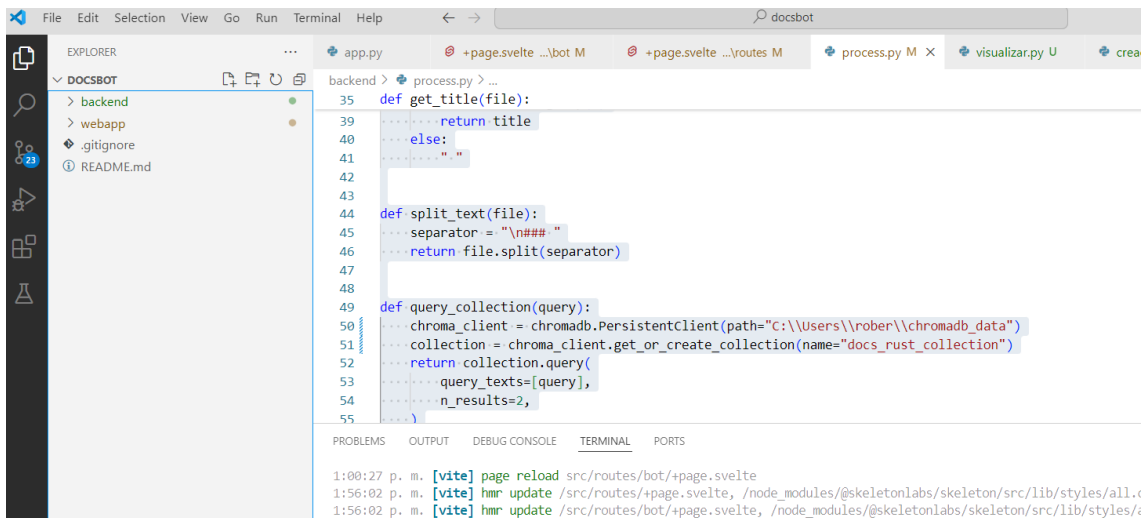
View usage per API key on the [Usage page](#).

NAME	SECRET KEY	CREATED	LAST USED ⓘ
IMBAUTO	sk-...C1s3	15 jul 2024	15 jul 2024

Continuando con el proyecto y con el desarrollo del Chatbot se ha creado un directorio para el desarrollo de todo el código del ChatBot

C:\Users\usuario\docsbot\

Este directorio contiene toda la programación del backend y de la webapp.



El backend es una aplicación Python con Flask que provee 2 endpoints process y query y la webapp es una app en sveltekit. Una vez desarrollado el código necesario se envía a ejecutar y lo hace a través de los puertos siguientes:

El backend Python en el puerto 5000

```
WARNING: This is a development server.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 812-836-460
```

La webapp se ejecuta en el puerto 5173

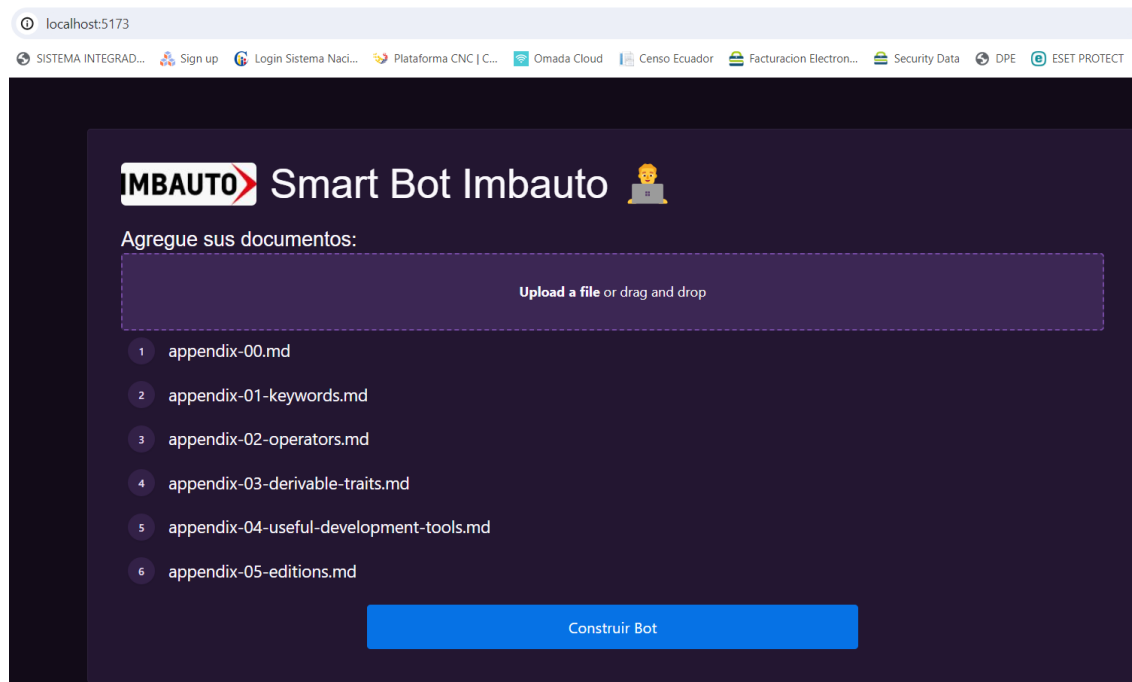
```
> docsbot@0.0.1 dev
> vite dev
```

```
VITE v4.3.3 ready in 2406 ms

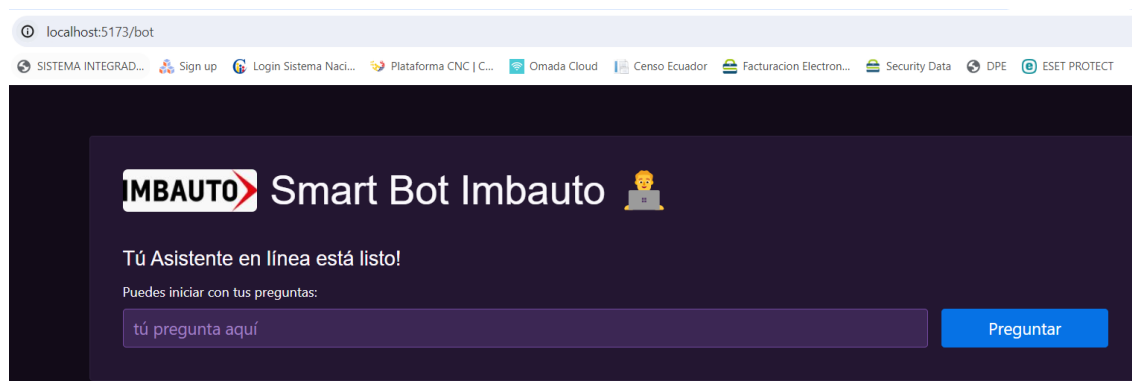
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h to show help
```

PROCESO DE EJECUCIÓN

Una vez en ejecución se procede a subir los documentos (formato markdown) de reparación de computadores, soporte técnico y diferentes soluciones a problemas que pueden ser resueltos por los usuarios de la empresa para lo cual se muestra la siguiente pantalla con el botón que permite subir y construir el bot..

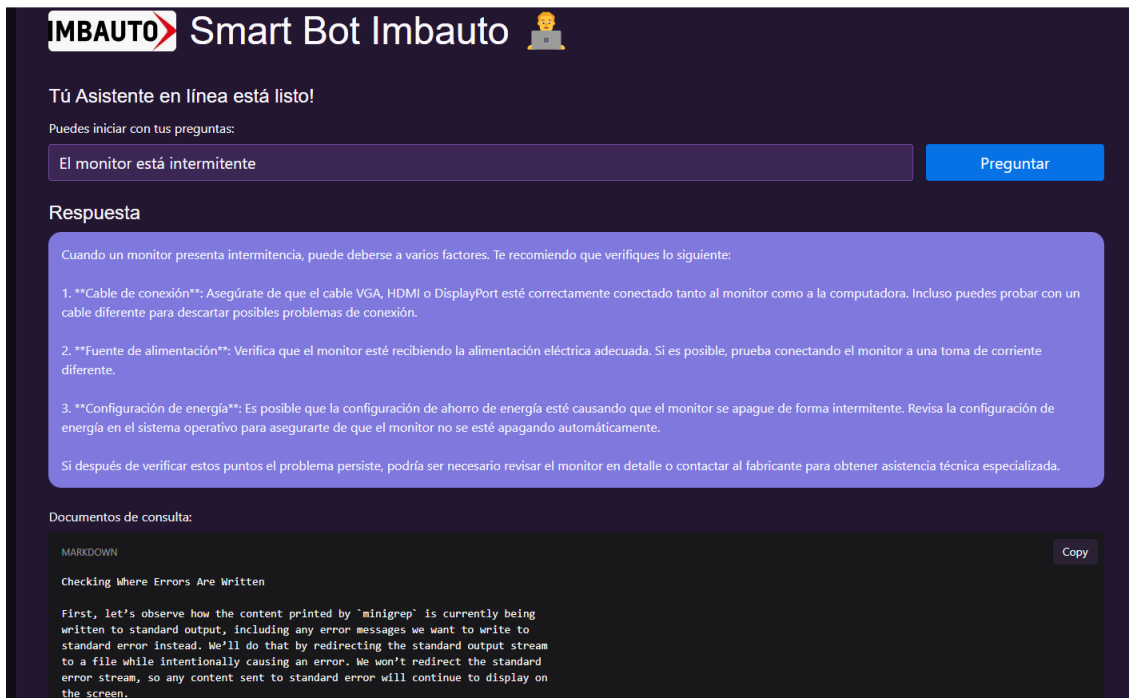


Una vez que se cargó los documentos markdown se muestra la siguiente pantalla donde podemos empezar a preguntar al bot.



Una vez ingresado una pregunta relacionada con los problemas de soporte técnico de sistemas en Imbauto se genera las respuestas y las fuentes de consulta de donde obtuvo información buscada en los archivos markdown, como se muestra en la siguiente pantalla. Este proceso lo hace

comparando los embeddings que tengan más similitud con la pregunta planteada, y toda esa información lo va a meter en un contexto y lo va a enviar a la API de OpenAi que en lenguaje humano sería como decir con esta información que se dispone puedes responder esta pregunta. Una vez realizado la pregunta entrega el resultado basado en la documentación que nosotros le pasamos seleccionando lo principal.



The screenshot shows a chat interface for 'IMBAUTO Smart Bot'. At the top, it says 'Tú Asistente en línea está listo!'. Below that, a prompt says 'Puedes iniciar con tus preguntas:'. A text input field contains 'El monitor está intermitente', and a blue button labeled 'Preguntar' is to its right. The response area, titled 'Respuesta', contains a message: 'Cuando un monitor presenta intermitencia, puede deberse a varios factores. Te recomiendo que verifiques lo siguiente:'. This is followed by a numbered list of three troubleshooting steps: 1. Check the connection cable (VGA, HDMI, or DisplayPort), 2. Check the power source, and 3. Check the power saving settings. A final note suggests contacting technical support if the problem persists. At the bottom, there is a 'Documentos de consulta:' section showing a snippet of a markdown document titled 'Checking Where Errors Are Written' with some introductory text about debugging.

Una vez visualizada la respuesta podemos continuar preguntando sin límite alguno y de acuerdo a la disponibilidad de soluciones que se encuentren los documentos markdowns subidos a la base de datos como embeddings.

PROGRAMACIÓN

La programación principal se encuentra en los siguientes archivos:

+page.svelte

Que contiene el script o java script y el diseño de formulario en html para la carga de documentos

```

webapp > src > routes > +page.svelte > script
1 <script lang="ts">
2   import { FileDropzone, ProgressBar } from '@skeletonlabs/skeleton';
3   import { PUBLIC_PROCESS_FILES_SERVER } from '$env/static/public';
4   import { goto } from '$app/navigation';
5
6   let files: FileList;
7   let processing = false;
8   let error = false;
9
10  function onChangeHandler(e: Event): void {
11    files = files;
12  }
13
14  async function startProcessing(event: Event) {
15    processing = true;
16    const formEl = event.target as HTMLFormElement;
17    const data = new FormData(formEl);
18    const response = await fetch(`${PUBLIC_PROCESS_FILES_SERVER}/process`, {
19      method: 'POST',
20      body: data
21    });
22    let result = await response.json();
23    if (result && result.success) {
24      goto('/bot');
25    } else {
26      error = true;
27    }
28  }
29 </script>

```

En este código se inicializa el cliente de chromaDB y luego agrega documentos, metadata y un ID a los documentos que internamente estaría creando el embedding con lo cual no es necesario llamar a la API de OPENAI ya que se lo hace de forma local.

Una vez cargado los documentos se ejecuta el método query al que se le pasa un texto e internamente convierte ese texto a un embedding y ejecuta la función de similitud y devuelve el resultado y se indica cuantos documentos se quiere que devuelva.

process.py

```

backend > process.py > process_files
1 import re
2 import chromadb
3 from chromadb.config import Settings
4
5 document_id = 1
6
7
8 def process_files(documents):
9     chroma_client = chromadb.PersistentClient(path="C:\\Users\\rober\\chromadb_data")
10    collection = chroma_client.get_or_create_collection(name="docs_rust_collection")
11
12    for file in documents:
13        print("processing file: " + file.filename)
14        markdown_text = file.read().decode()
15        chunks = split_text(markdown_text)
16        document_title = get_title(markdown_text)
17        generate_embeddings(chunks, document_title, file.filename, collection)
18
19
20
21 def generate_embeddings(chunks, document_title, file_name, collection):
22     global document_id
23     for chunk in chunks:
24         collection.add(
25             metadatas={
26                 "document_title": document_title if document_title is not None else "",
27                 "file_name": file_name
28             },
29             documents=chunk,

```

EJEMPLO 2:

BACKEND

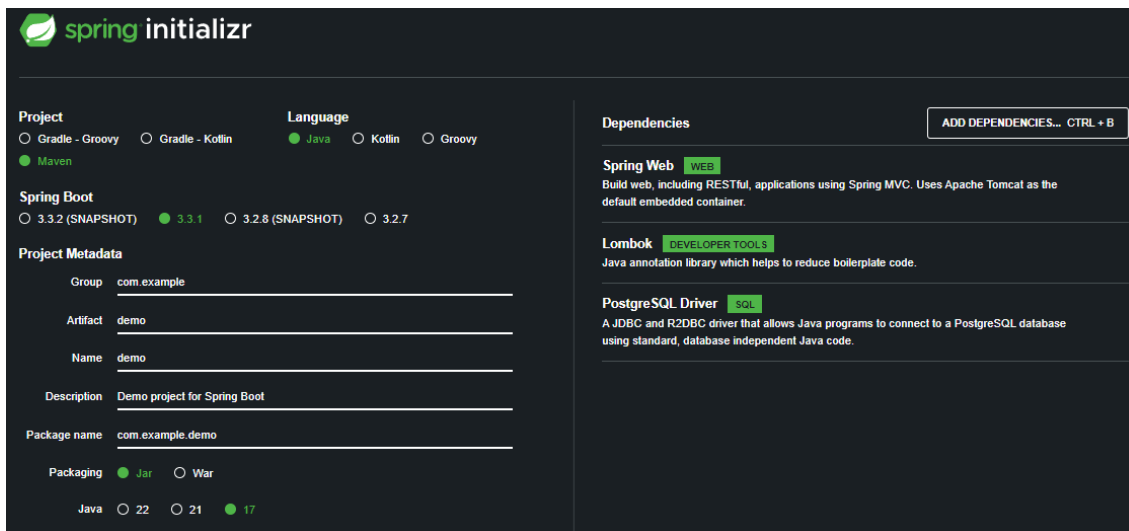
El sistema backend está desarrollado utilizando el framework Spring Boot, implementado en lenguaje Java. La arquitectura se basa en los siguientes componentes y tecnologías:

1. Spring Boot: Proporciona un entorno de desarrollo ágil y simplificado para aplicaciones Java, facilitando la configuración y el despliegue.
2. Java Web: Se emplean las APIs estándar de Java para el desarrollo web, incluyendo Servlets y JavaServer Pages (JSP), para el manejo de solicitudes HTTP y la generación de respuestas dinámicas.
3. JPA (Java Persistence API): Se utiliza como capa de abstracción para la persistencia de datos, simplificando las operaciones de base de datos y proporcionando un mapeo objeto-relacional eficiente.
4. Hibernate: Implementa la especificación JPA, actuando como el proveedor de persistencia principal. Hibernate gestiona el mapeo entre los objetos Java y las tablas de la base de datos, optimizando las consultas y el rendimiento de las operaciones de datos.
5. Lombok: Se integra esta biblioteca para reducir el código boilerplate, automatizando la generación de métodos getter, setter, constructores y otros elementos comunes. Esto mejora la legibilidad del código y aumenta la productividad del desarrollo.
6. Maven: Se utiliza como herramienta de gestión y construcción del proyecto. Maven facilita la administración de dependencias, la configuración del ciclo de vida del proyecto y la estandarización de la estructura del mismo. Además, permite la generación de informes y la integración con diversas herramientas de desarrollo.

INSTALACIÓN Y CONFIGURACIÓN:

Para inicializar el proyecto se puede realizar las configuraciones básicas desde el sitio web de SpringBoot, accediendo mediante el enlace: <https://start.spring.io/>

En este sitio podemos seleccionar las configuraciones base con las que se inicia el proyecto, una vez seleccionadas las opciones, Generamos el proyecto, con lo que obtenemos una carpeta con la estructura del proyecto que podemos importarlo a un IDE de desarrollo para continuar con el proceso.

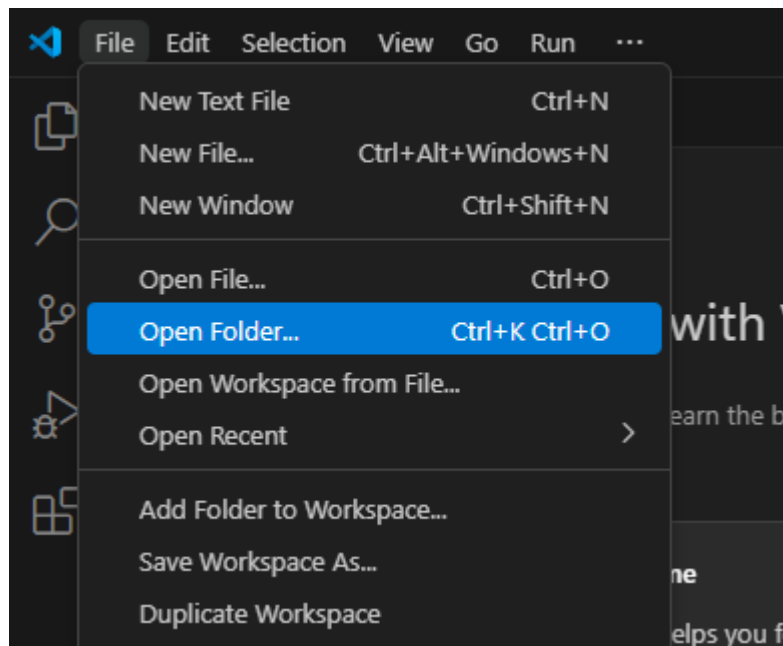


The image shows the Spring Initializr web interface. It has a dark theme. On the left, there are sections for 'Project' (with radio buttons for Gradle - Groovy, Gradle - Kotlin, and Maven), 'Language' (with radio buttons for Java, Kotlin, and Groovy), 'Spring Boot' (with radio buttons for 3.3.2 (SNAPSHOT), 3.3.1, 3.2.8 (SNAPSHOT), and 3.2.7), and 'Project Metadata' (with input fields for Group, Artifact, Name, Description, and Package name, and radio buttons for Packaging and Java version). On the right, there is a 'Dependencies' section with a button 'ADD DEPENDENCIES... CTRL + B'. Below this, there are three dependency cards: 'Spring Web' (WEB), 'Lombok' (DEVELOPER TOOLS), and 'PostgreSQL Driver' (SQL).

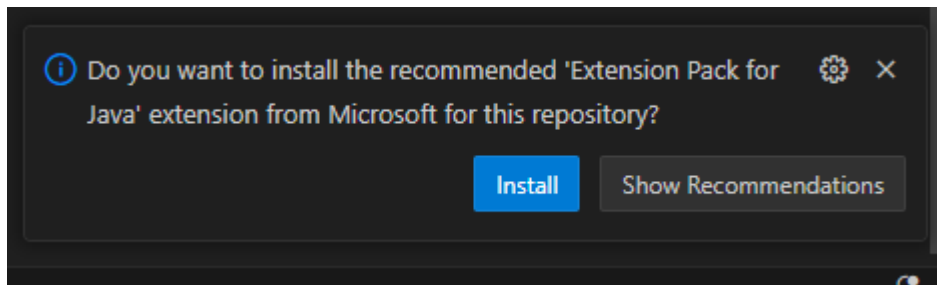
En este caso vamos a usar Visual Studio Code por las facilidades que brinda en el proceso de desarrollo y el mínimo consumo de recursos.

Es importante recalcar que para el desarrollo del BackEnd es necesaria la instalación de JAVA jdk en el computador, además de la instalación es necesario que se configuren las variables de entorno respectivas.

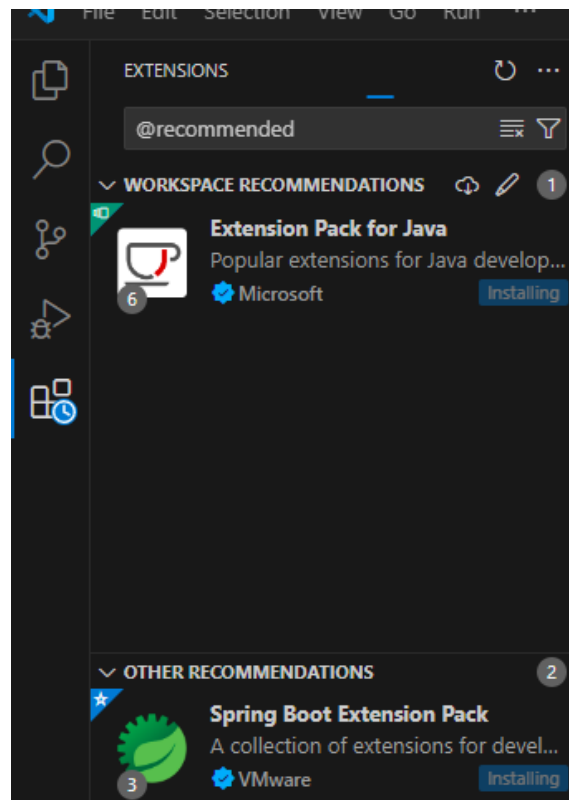
Abrimos Visual Studio Code y seleccionamos la opción Open Folder para acceder al proyecto que generamos en SpringBoot.



Dependiendo del proyecto, es necesario y recomendado el instalar complementos que ayuden al desarrollo de las funcionalidades, en el caso del BackEnd los complementos a configurar son los relacionados con SpringBoot y Java para el correcto funcionamiento del proyecto.

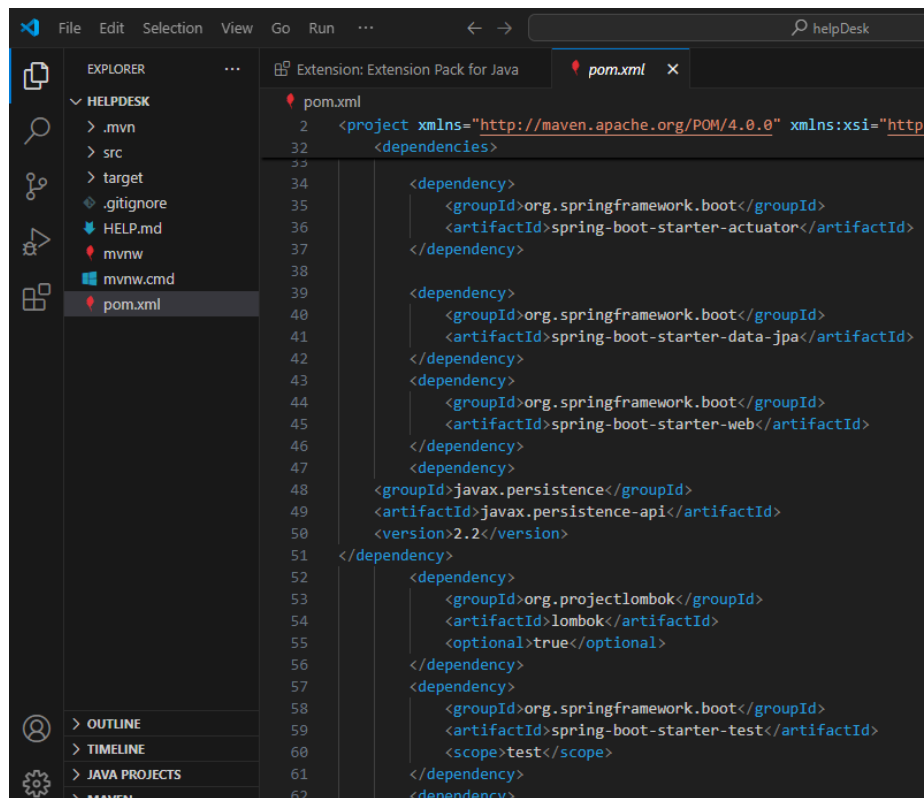


Una vez se acceda al proyecto, por defecto Visual Studio Code suele recomendar las extensiones necesarias o también podemos instalarlas desde las opciones de VSC.



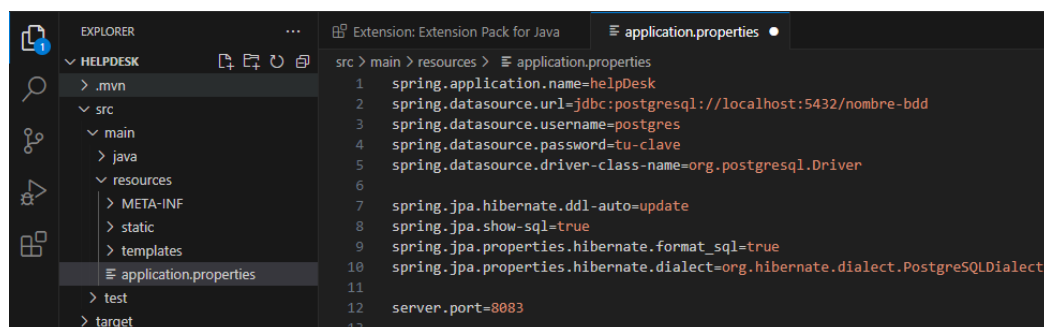
Una vez tenemos las configuraciones base del proyecto procedemos a generar las configuraciones internas y la estructura de carpetas, en nuestro caso tenemos un manejo MVC.

La herramienta de gestión y configuración del proyecto es Maven, basados en esta estructura todas las instalaciones y configuraciones de dependencias básicas o que se requiera agregar se encuentran en el archivo pom.xml alojado en la raíz del proyecto, este archivo es vital para poder utilizar las herramientas que se requiera.

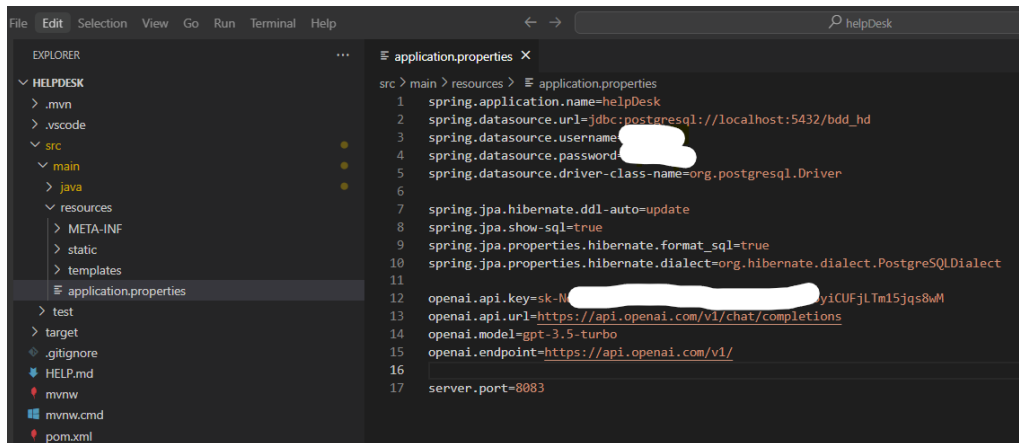


```
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://maven.apache.org/xsi:schemaLocation" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>org.springframework.boot</groupId>
5   <artifactId>spring-boot-starter-actuator</artifactId>
6   <version>2.2.0</version>
7   <packaging>war</packaging>
8   <name>Spring Boot Actuator</name>
9   <description>Spring Boot Actuator</description>
10  <dependencies>
11    <dependency>
12      <groupId>org.springframework.boot</groupId>
13      <artifactId>spring-boot-starter</artifactId>
14    </dependency>
15    <dependency>
16      <groupId>org.springframework.boot</groupId>
17      <artifactId>spring-boot-starter-data-jpa</artifactId>
18    </dependency>
19    <dependency>
20      <groupId>org.springframework.boot</groupId>
21      <artifactId>spring-boot-starter-web</artifactId>
22    </dependency>
23    <dependency>
24      <groupId>javax.persistence</groupId>
25      <artifactId>javax.persistence-api</artifactId>
26      <version>2.2</version>
27    </dependency>
28    <dependency>
29      <groupId>org.projectlombok</groupId>
30      <artifactId>lombok</artifactId>
31      <optional>true</optional>
32    </dependency>
33    <dependency>
34      <groupId>org.springframework.boot</groupId>
35      <artifactId>spring-boot-starter-test</artifactId>
36      <scope>test</scope>
37    </dependency>
38  </dependencies>
39  </project>
```

La estructura del proyecto tiene como base la carpeta **src**, esta es la que contiene las configuraciones y contendrá la estructura del modelo, el siguiente paso de la configuración es realizar cambios en el archivo **application.properties**, una de las configuraciones importantes a realizar es la conexión con la base de datos en este caso tenemos POSTGRESQL, además en este caso tenemos la configuración del puerto en el que se va a ejecutar el proyecto, por defecto este se ejecuta bajo el puerto 8080, sin embargo es importante este cambio para evitar que se tenga conflicto.

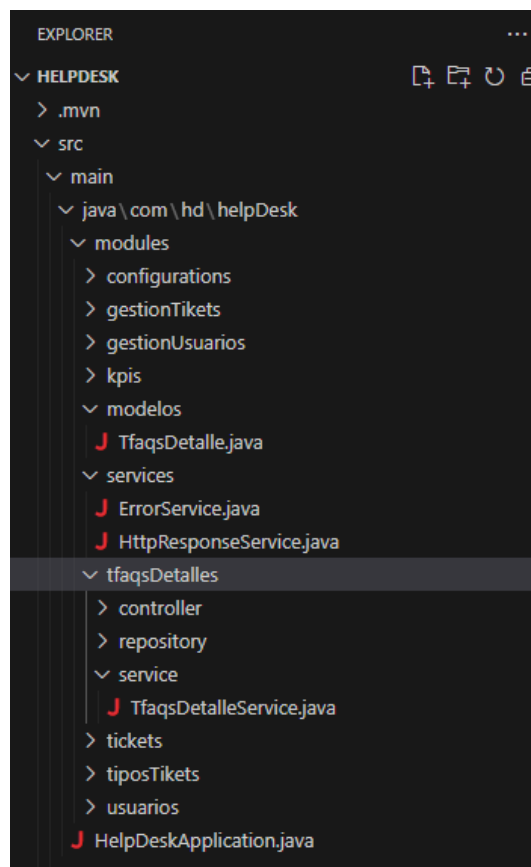


```
1 spring.application.name=helpDesk
2 spring.datasource.url=jdbc:postgresql://localhost:5432/nombre-bdd
3 spring.datasource.username=postgres
4 spring.datasource.password=tu-clave
5 spring.datasource.driver-class-name=org.postgresql.Driver
6
7 spring.jpa.hibernate.ddl-auto=update
8 spring.jpa.show-sql=true
9 spring.jpa.properties.hibernate.format_sql=true
10 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
11
12 server.port=8083
13
```



DESARROLLO DEL PROYECTO

Para una mejor administración de las funcionalidades del sistema se ha definido la siguiente estructura:



Creamos la carpeta **modules** que guardará toda la configuración y separará las funcionalidades del sistema considerando la estructura de la base de datos, como lo es tickets, tipos de tickets, gestión de tickets, usuarios, gestión de usuarios, kpis entre otros. Dado que vamos a realizar toda la gestión

del proyecto y la base de datos desde el BackEnd, es necesario que contemos con todos los módulos que permitan la generación de servicios para métodos CRUD.

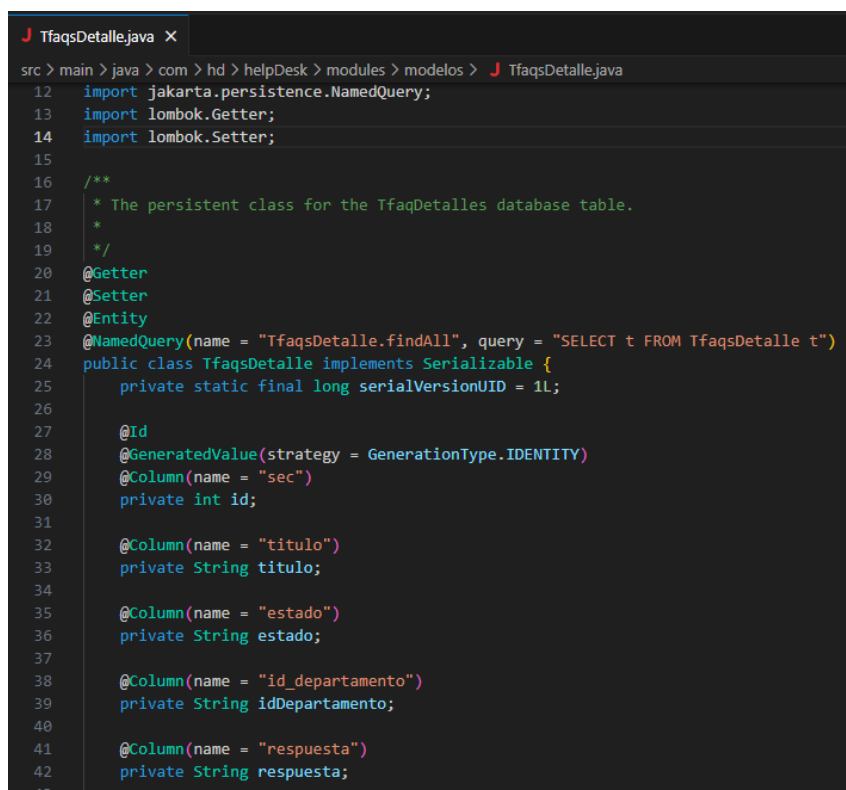
Cada módulo se creará dentro de la carpeta **modules**, el primer y más importante a configurar es modelos, este contendrá las clases java que referencian la estructura de la base de datos y permite su conexión mediante JPA e Hibernate.

Las clases java de esta sección deberán tener el nombre de la tabla de la base de datos, con las siguientes consideraciones:

- La primera letra debe ser mayúscula.
- Si existen dos palabras separadas por guion, estas estarán unidas por la primera letra de la segunda palabra en mayúscula.
- Si la tabla termina en la letra s, esta se omitirá del nombre de la clase.
- Dentro de cada clase se realizarán las anotaciones respectivas para ser identificada como la presentación de la tabla y declararan las variables que representan a las columnas.

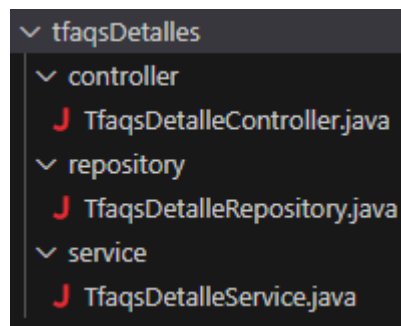
Considerando esto, tenemos el siguiente ejemplo:

- **Nombre de la tabla en la base de datos:** tfaqs_detalle
- **Nombre de la clase java:** TfaqsDetalle.java



```
J TfaqsDetalle.java X
src > main > java > com > hd > helpDesk > modules > modelos > J TfaqsDetalle.java
12 import jakarta.persistence.NamedQuery;
13 import lombok.Getter;
14 import lombok.Setter;
15
16 /**
17  * The persistent class for the TfaqsDetalles database table.
18  *
19  */
20 @Getter
21 @Setter
22 @Entity
23 @NamedQuery(name = "TfaqsDetalle.findAll", query = "SELECT t FROM TfaqsDetalle t")
24 public class TfaqsDetalle implements Serializable {
25     private static final long serialVersionUID = 1L;
26
27     @Id
28     @GeneratedValue(strategy = GenerationType.IDENTITY)
29     @Column(name = "sec")
30     private int id;
31
32     @Column(name = "titulo")
33     private String titulo;
34
35     @Column(name = "estado")
36     private String estado;
37
38     @Column(name = "id_departamento")
39     private String idDepartamento;
40
41     @Column(name = "respuesta")
42     private String respuesta;
43 }
```

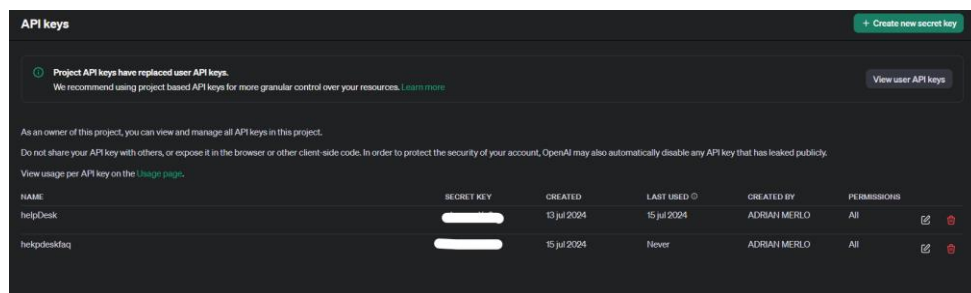
Para cada módulo tendremos las carpetas **repository**, **service** y **controller** que contienen las clases java necesarias que en conjunto permiten el funcionamiento del proyecto. Dentro de cada carpeta tendremos un archivo con el nombre del módulo y la palabra Repository, Service o Controller como corresponda.



USO DE OPENAI API

El primer paso es acceder a la página del API <https://platform.openai.com/>, es necesario el crear una cuenta e iniciar sesión para hacer uso del API.

En la página de OPENAI procedemos a crear una nueva APIKEY que se usará con los diferentes modelos existentes en la plataforma, esta APIKEY deberá ser almacenada en un sitio seguro y no se debe compartir.



Para referencia de la estructura de los modelos que se puede utilizar, hay que acceder a la página: <https://platform.openai.com/docs/models>

Vincular el API con el BackEnd

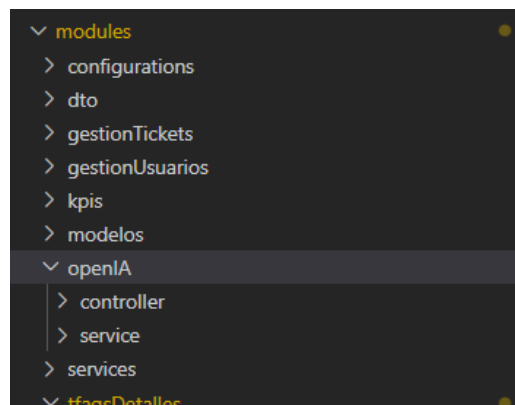
El primer paso es realizar cambios en el archivo de configuración para asignar los parámetros base para el APIKEY.

Editamos el archivo application.properties agregando las variables que utilizará el proyecto para la conexión con el APIKEY.

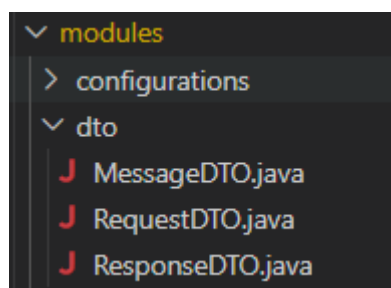
```
application.properties x  {} additional-spring-configuration-metadata.json
src > main > resources > application.properties
1  spring.application.name=helpDesk
2  spring.datasource.url=jdbc:postgresql://localhost:5432/[redacted]
3  spring.datasource.username=[redacted]
4  spring.datasource.password=[redacted]
5  spring.datasource.driver-class-name=org.postgresql.Driver
6
7  spring.jpa.hibernate.ddl-auto=update
8  spring.jpa.show-sql=true
9  spring.jpa.properties.hibernate.format_sql=true
10 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
11
12 openai.api.key=[redacted]
13 openai.api.url=https://api.openai.com/v1/chat/completions
14 openai.model=gpt-3.5-turbo
15 openai.endpoint=https://api.openai.com/v1/
16
17 server.port=8083
```

Los campos modelo, endpoint y url deben ser modificados en base a las diferentes opciones descritas en la documentación del APIKEY, es importante el definir el tipo de uso que se dará al APIKEY, ya que de ello dependen las configuraciones que se deberá aplicar.

Para el BackEnd vamos a crear una estructura para OPENAI que maneje los archivos para la creación de servicio y para el controlador que tendrá los servicios REST a ser consumidos.



Además de la estructura de archivos Data Transfer Object (DTO) que serán utilizados en la comunicación con el APIKEY y para tener una mejor organización del código.



Para la comunicación con el APIKEY es necesario agregar la cabecera de conexión, esta configuración la realizamos en la clase principal del proyecto, mediante una clase de configuración propia de springframework. De esta manera instanciamos al proyecto para que pueda usar el APIKEY desde cualquier servicio.

```
src > main > java > com > hd > helpDesk > HelpDeskApplication.java > Language Support for Java(TM) by Red Hat
18 public class HelpDeskApplication {
19
20     Run | Debug
21     public static void main(String[] args) {
22         SpringApplication.run(HelpDeskApplication.class, args);
23     }
24
25     @Value("${openai.api.key}")
26     private String openaiApiKey;
27
28     @Bean
29     public RestTemplate restTemplate() {
30         RestTemplate restTemplate = new RestTemplate();
31         restTemplate.getInterceptors().add((request, body, execution) -> {
32             request.getHeaders().add("Authorization", "Bearer " + openaiApiKey);
33             return execution.execute(request, body);
34         });
35         return restTemplate;
36     }
37 }
```

Generamos los servicios que se comunicarán con el APIKEY dentro de un archivo con la anotación `@Service`, en este caso se está utilizando un modelo que permite realizar una consulta directa al API y otro método para configurar la respuesta, orientando a obtener mejores resultados sobre el modelo establecido.

```
src > main > java > com > hd > helpDesk > modules > openAI > service > OpenAIService.java > Language Support for Java(TM) by Red Hat > Open
11
12 @Service
13 public class OpenAIService {
14
15     @Value("${openai.model}")
16     private String model;
17     @Value("${openai.api.url}")
18     private String apiUrl;
19
20     @Autowired
21     private RestTemplate restTemplate;
22
23     @SuppressWarnings("null")
24     public String callOpenAIByPrompt(String prompt) {
25         String respuesta = "";
26         RequestDTO request = new RequestDTO(model, this.getPrompt(prompt));
27         ResponseDTO response = restTemplate.postForObject(apiUrl, request, ResponseDTO.class);
28         System.out.println(response.getChoices().get(index:0).getMessage());
29         respuesta = response.getChoices().get(index:0).getMessage().getContent();
30         return respuesta;
31     }
32
33     @SuppressWarnings("null")
34     public MessageDTO getRespuesta(String prompt) {
35         RequestDTO requestDto = new RequestDTO(model, this.getPrompt(prompt));
36         ResponseDTO responseDto = restTemplate.postForObject(apiUrl, requestDto, ResponseDTO.class);
37         if (requestDto == null || responseDto.getChoices() == null || responseDto.getChoices().isEmpty())
38             return new MessageDTO(role:"user", content:"falló en la consulta");
39         return responseDto.getChoices().get(index:0).getMessage();
40     }
41
42     private String getPrompt(String consulta) {
43         String prompt = " Presenta la mejor opción de respuesta para la consulta : " + consulta
44             + " \n considerando que el tópico tratado es soporte de software.";
45         return prompt;
46     }
47
48 }
```

Finalmente generamos los servicios REST que permitirán dar una respuesta en el formato establecido.

```
OpenIAController.java x TfaqsDetalleService.java 1 TfaqsDetalleRepository.java
src > main > java > com > hd > helpDesk > modules > openIA > controller > OpenIAController.java > ...
12 import com.hd.helpDesk.modules.dto.MessageDTO;
13 import com.hd.helpDesk.modules.openIA.service.OpenIAService;
14
15 @RestController
16 @CrossOrigin
17 public class OpenIAController {
18
19     @Autowired
20     private OpenIAService openIAService;
21
22     @PostMapping("/chat")
23     public ResponseEntity<MessageDTO> chat(@RequestBody String prompt) {
24         MessageDTO messageDTO = new MessageDTO();
25         messageDTO = this.openIAService.getRespuesta(prompt);
26
27         return ResponseEntity.ok(messageDTO);
28     }
29
30     @GetMapping("/getchat")
31     public String pruebaChat(@RequestParam("prompt") String prompt) {
32         String respuesta = this.openIAService.callOpenAIByPrompt(prompt);
33         return respuesta;
34     }
35 }
36
```

Los servicios generados pueden consultarse desde el FrontEnd o mediante herramientas como Postman para verificar que estas funcionen correctamente.



FRONTEND

Para el desarrollo del frontend, hemos optado por utilizar Angular, un framework robusto y ampliamente adoptado en la industria. La elección de Angular se basa en los siguientes factores:

1. Arquitectura modular: Angular proporciona una estructura organizada y escalable para el desarrollo de aplicaciones web complejas.

2. **Typescript:** Angular utiliza TypeScript como lenguaje principal, ofreciendo tipado estático y características avanzadas de programación orientada a objetos, lo que mejora la calidad y mantenibilidad del código.
3. **Componentes reutilizables:** Facilita la creación de componentes independientes y reutilizables, promoviendo un desarrollo más eficiente y coherente.
4. **Integración eficiente con backend:** Angular ofrece herramientas poderosas para el consumo de servicios RESTful, permitiendo una comunicación fluida con el backend Spring Boot.
5. **Rendering del lado del cliente:** Proporciona una experiencia de usuario más rápida y receptiva al manejar la renderización en el navegador.
6. **Ecosistema rico:** Cuenta con un amplio conjunto de bibliotecas y herramientas que facilitan el desarrollo, pruebas y despliegue.

La estructura típica de una aplicación Angular incluye:

- **Componentes:** Unidades básicas de la interfaz de usuario.
- **Servicios:** Para la lógica de negocio y comunicación con el backend.
- **Módulos:** Para organizar y encapsular funcionalidades relacionadas.
- **Directivas:** Para extender el comportamiento del HTML.
- **Pipes:** Para la transformación de datos en la vista.

Angular utiliza un sistema de templates basado en HTML extendido con directivas propias, y emplea conceptos como el data binding bidireccional y la inyección de dependencias para facilitar el desarrollo de aplicaciones web dinámicas y de alto rendimiento.

INSTALACIÓN Y CONFIGURACIÓN:

Para utilizar Angular, es necesario cumplir con ciertos requisitos previos y seguir una serie de pasos de instalación. A continuación, se detallan los requisitos y el proceso de instalación:

1. Instalar Node.js y npm

Visitar la página oficial de Node.js nodejs.org y descargar el instalador adecuado para tu sistema operativo. Sigue las instrucciones de instalación. Una vez instalado, verifica la instalación ejecutando los siguientes comandos en tu terminal o línea de comandos:

```
node -v
npm -v
```

2. Instalar Angular CLI

Angular CLI (Command Line Interface) es una herramienta que facilita la creación, desarrollo y mantenimiento de aplicaciones Angular. Para instalar Angular CLI, ejecutar el siguiente comando en tu terminal:

```
npm install -g @angular/cli
```

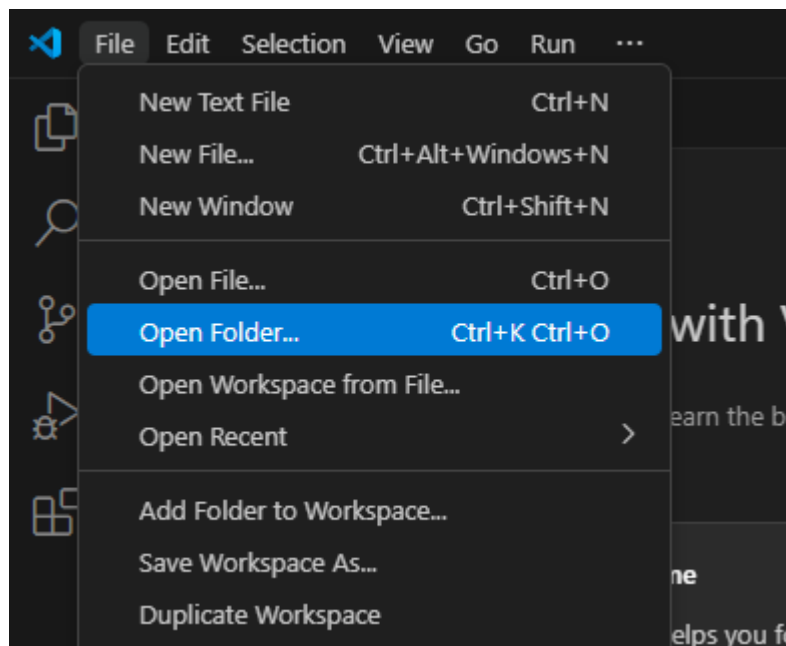
3. Crear un Proyecto Angular

Para crear un proyecto Angular se puede realizar mediante comandos o descargar templates personalizados de diferentes sitios web, en caso de generar el proyecto se debe utilizar el siguiente comando:

```
ng new nombre-del-proyecto
```

4. Configuraciones del proyecto

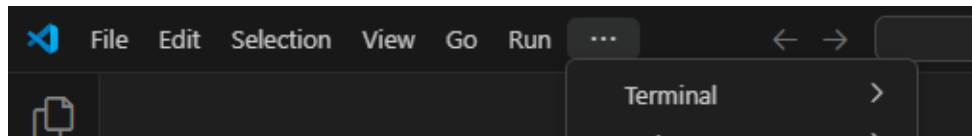
Una vez generado el proyecto tendremos una estructura de carpetas y archivos de configuración de Angular, estos pueden editarse utilizando Visual Studio Code, para lo cual abrimos el programa y seleccionamos la opción Open Folder, para cargar el proyecto.



Dependiendo del proyecto, es necesario y recomendado el instalar complementos que ayuden al desarrollo de las funcionalidades. El proyecto mantiene una estructura de Carpetas y archivos de configuración que se debe respetar para el correcto funcionamiento del mismo.

5. Ejecutar la Aplicación Angular

Para la ejecución del proyecto es necesario acceder a La Terminal de Visual Studio Code

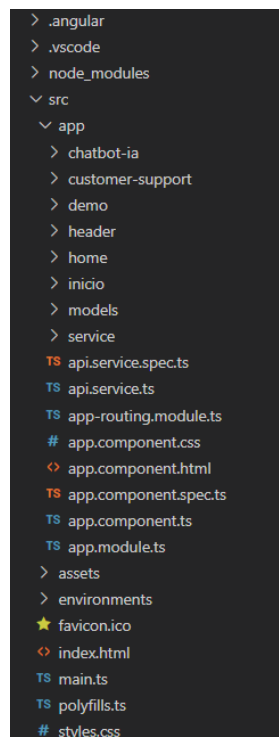


Verificar que se encuentre en la raíz de la estructura del proyecto e ingresar el siguiente comando:

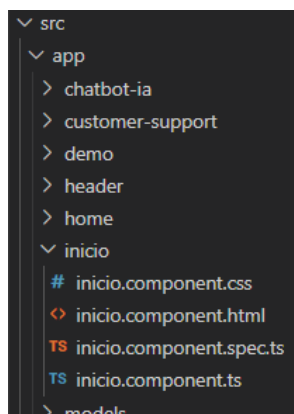
```
i-chat-bot-app-in-angular-using-chatGPT-main> ng serve
```

Luego es necesario abrir un navegador y ve a acceder a <http://localhost:4200> para ver la aplicación Angular en funcionamiento.

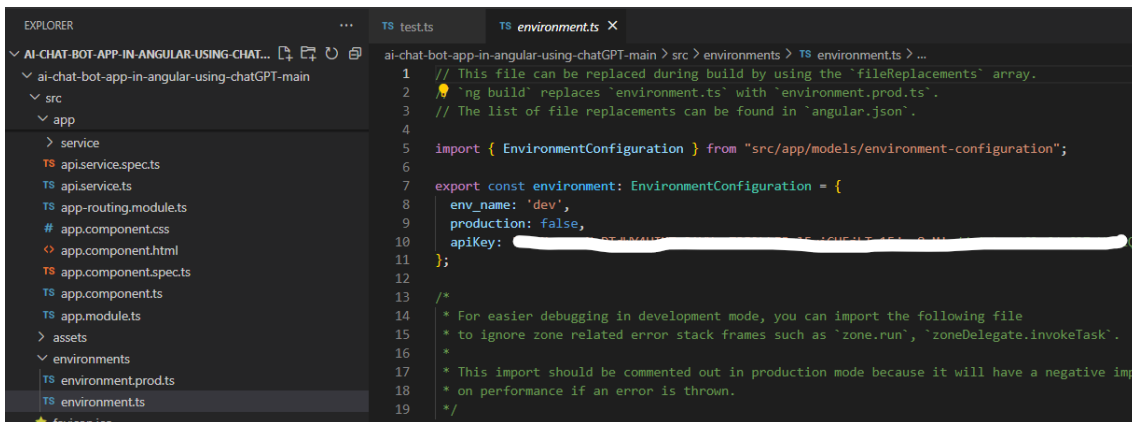
En cuanto a la estructura del proyecto, las acciones a realizarse se enfocarán en la ruta src/app.



Dada la naturaleza de angular se manejarán componentes, los cuales manejan los archivos para la vista (HTML) y el controlador (Typescript), además de la clase para manejo de css personalizado del componente.



Podemos generar la comunicación del FrontEnd con el APIKEY, para hacer consultas sin necesidad de acceder primero al BackEnd, en este caso es necesario configurar el archivo enviroment que manejará el APIKEY generado.



En este caso vamos a generar un chat para mantener conexión con el APIKEY y el usuario registrado en el sistema, usaremos el modelo: 'gpt-3.5-turbo-instruct', que permite el manejo de preguntas y respuestas, hay que considerar que dependiendo del modelo utilizado será necesaria la configuración del endpoint o url y otros parámetros como la temperatura y el máximo de tokens a utilizar.

```
let requestData = {
  model: 'gpt-3.5-turbo-instruct', // 'text-davinci-003', // "text-curie-001",
  prompt: this.promptText, // this.generatePrompt(animals),
  temperature: 0.5,
  max_tokens: 1000,
  top_p: 1.0,
  frequency_penalty: 0.0,
  presence_penalty: 0.0,
};

this.showSpinner = true;
let apiResponse = await openai.createCompletion(requestData);

this.response = apiResponse.data as ResponseModel;
```

De esta manera se realiza las configuraciones en el archivo del modulo .ts y se agrega archivo html a la estructura de presentación.

Para ejecutar la aplicación se abre la consola y se ejecuta el comando “ng serve”

```
PS D:\OneDrive - CORPORACION WAY\Users\Adrian Merlo\Downloads\ai-chat-bot-app-in-angular-using-chatGPT-main\ai-chat-bot-app-in-angular-using-chatGPT-main> ng serve
Node.js version v17.9.1 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please see https://nodejs.org/en/about/releases/.
Your global Angular CLI version (14.2.13) is greater than your local version (14.2.10). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
✔ Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size
vendor.js           | vendor | 2.80 MB
polyfills.js        | polyfills | 12.83 MB
```

Luego abrimos en el navegador el URL de la aplicación, nos logeamos y empezamos a interactuar con el Chat bot



SISTEMA HelpDesk

Sistema de Gestión de Tickets de IMBAUTO
S.A.

Usuario:

Contraseña:

Iniciar Sesión

IMBAUTO

Registrar Tickets Ver Tickets Chat Bot Imbauto

ADRIAN

Smart Bot Imbauto - HelpDesk

¿En qué te puedo ayudar hoy?

Ingresar tu pregunta

Enviar

9.8. INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Entrevista al jefe del departamento de sistemas

¿Podría describir brevemente cómo se gestiona actualmente el soporte técnico en Imbauto S.A.?

Actualmente, el soporte técnico en Imbauto S.A. se gestiona de manera descentralizada. Las solicitudes de soporte llegan a través de varios canales, como correo electrónico, llamadas telefónicas y mensajes de WhatsApp. No contamos con una herramienta unificada para registrar y gestionar todas estas solicitudes, lo que complica la organización y el seguimiento.

¿Qué canales utilizan los usuarios para solicitar soporte técnico?

Los usuarios utilizan principalmente tres canales para solicitar soporte: correo electrónico, llamadas telefónicas y WhatsApp. Cada uno de estos canales es gestionado de forma independiente, lo que crea una dispersión de información y dificulta tener un control efectivo sobre las solicitudes.

¿Cuáles son los principales problemas que ha identificado en la gestión actual del soporte técnico?

Los principales problemas son la desorganización y el riesgo de perder solicitudes. La ausencia de un sistema centralizado hace que sea difícil priorizar y dar seguimiento adecuado a las solicitudes. Además, sin una herramienta que nos permita ver todas las solicitudes en un solo lugar, es complicado monitorear la carga de trabajo del equipo y asegurar que se atiendan todas las solicitudes de manera oportuna.

¿Cómo afecta el no registro centralizado de solicitudes de soporte a la eficiencia operativa?

La ausencia de un registro centralizado afecta la eficiencia operativa de varias maneras. Primero, nos impide establecer prioridades claras, lo que significa que solicitudes urgentes pueden no ser atendidas a tiempo. Segundo, dificulta el seguimiento y control de las solicitudes, ya que no tenemos visibilidad sobre cuáles están en progreso, cuáles están resueltas y cuáles están pendientes. Esto también afecta nuestra capacidad para generar reportes y estadísticas sobre el desempeño del soporte técnico.

¿Cómo se establecen actualmente las prioridades para las solicitudes de soporte?

Las prioridades se establecen de manera ad hoc, dependiendo de la percepción del técnico que recibe la solicitud. Sin un sistema centralizado, es difícil aplicar criterios consistentes de priorización, lo que puede resultar en la atención tardía de solicitudes críticas.

¿Qué dificultades enfrenta el equipo para realizar el seguimiento y control de las solicitudes de soporte?

La principal dificultad es la no visibilidad y control. Sin una herramienta centralizada, los técnicos no pueden ver el estado de todas las solicitudes, lo que lleva a la dificultad de coordinación y posibles duplicaciones de esfuerzos. Además, no podemos fácilmente monitorear si todas las solicitudes han sido resueltas y en qué tiempos, lo que afecta la calidad del servicio que brindamos.

¿Cómo se notifica a los usuarios sobre el estado y la resolución de sus solicitudes de soporte?

Actualmente, las notificaciones se manejan de manera manual. Los técnicos se comunican con los usuarios a través de correo electrónico o llamadas telefónicas para informarles sobre el estado y la resolución de sus solicitudes. Este proceso manual es ineficiente y propenso a errores, lo que genera frustración entre los usuarios.

¿Qué tipo de información y métricas se recopilan actualmente sobre el soporte técnico?

Actualmente, no recopilamos de manera sistemática información o métricas sobre el soporte técnico. La ausencia de un sistema centralizado nos impide registrar datos importantes como el tiempo de respuesta, la satisfacción del usuario, el tiempo de resolución y la cantidad de tickets abiertos y cerrados. Sin estas métricas, es difícil evaluar nuestro desempeño y hacer mejoras basadas en datos.

¿Existe actualmente una base de conocimiento accesible para los usuarios?

No, actualmente no contamos con una base de conocimiento accesible para los usuarios. Esto significa que los usuarios dependen completamente del equipo de soporte para resolver incluso los problemas más simples, lo que incrementa la carga de trabajo de los técnicos y reduce la eficiencia operativa.

Cuestionario Tipo Likert

Objetivo del Cuestionario: Recolectar información sobre las experiencias, niveles de satisfacción y percepciones de los usuarios del sistema de soporte técnico en Imbauto S.A.

Instrucciones: Por favor, lea cada afirmación y seleccione el nivel de acuerdo que mejor refleje su experiencia con el soporte técnico

Preguntas:

- 1. ¿Cómo contactaste con el personal del departamento de Sistemas para solicitar soporte?**
 - En persona
 - Teléfono
 - Correo
 - Chat
 - Otro
- 2. Estoy satisfecho con el tiempo de respuesta del equipo de soporte técnico a mis solicitudes.**
 - Totalmente en desacuerdo
 - En desacuerdo
 - Neutral
 - De acuerdo
 - Totalmente de acuerdo
- 3. El equipo de soporte técnico maneja adecuadamente la priorización de solicitudes urgentes.**
 - Totalmente en desacuerdo
 - En desacuerdo
 - Neutral
 - De acuerdo
 - Totalmente de acuerdo
- 4. Los problemas que reporto al soporte técnico son resueltos de manera efectiva.**
 - Totalmente en desacuerdo
 - En desacuerdo

- Neutral
- De acuerdo
- Totalmente de acuerdo

5. Recibo actualizaciones periódicas sobre el estado de mis solicitudes de soporte.

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

6. La comunicación del equipo de soporte técnico respecto al estado y resolución de mis solicitudes es clara y efectiva.

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

7. Considero que el soporte técnico tiene suficiente información y métricas para mejorar continuamente sus servicios.

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

8. Me sería útil tener acceso a una base de conocimiento en línea para resolver problemas comunes por mi cuenta.

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

9. En general, estoy satisfecho con el servicio de soporte técnico que recibo.

- Totalmente en desacuerdo
- En desacuerdo

- Neutral
- De acuerdo
- Totalmente de acuerdo

10. Creo que la implementación de una aplicación web para la gestión de soporte técnico mejoraría significativamente la eficiencia del servicio.

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo