



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

TÍTULO DE ANTEPROYECTO DE TITULACIÓN

**“DESARROLLO DE UN ASISTENTE VIRTUAL DE GESTIÓN DE
INCIDENTES DE PRIMER NIVEL EN TICS UTILIZANDO
TECNOLOGÍAS EMERGENTES PARA EL MINISTERIO DE
INCLUSIÓN ECONÓMICA Y SOCIAL - MIES”**

AUTOR

BERNABÉ CHASIN JORDY JOSEPH

PROYECTO UNIDAD DE INTEGRACIÓN CURRICULAR

**Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

TUTOR

Ing. Coronel Suárez Marjorie Alexandra, Mgt

Santa Elena, Ecuador

Año 2024



UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino, Mgt.
DIRECTOR DE LA CARRERA

Ing. Marjorie Coronel Suárez, Mgt.
TUTOR

Ing. Carlos Sánchez León, Mgt.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez, Mgt.
DOCENTE GUÍA UIC



UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **Bernabé Chasin Jordy Joseph**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 12 días del mes de junio del año 2024

TUTOR

Ing. Marjorie Coronel S. Mgti



**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA
ELENA FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, **BERNABÉ CHASIN JORDY JOSEPH**

DECLARO QUE:

El trabajo de Titulación, Desarrollo de un asistente virtual de gestión de incidentes de primer nivel en TICS utilizando tecnologías emergentes para el Ministerio de Inclusión Económica y Social (MIES) previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 12 días del mes de junio del año 2024

EL AUTOR

A handwritten signature in black ink, appearing to read "Jordy Bernabé", is written over a horizontal line.

Jordy Joseph Bernabé Chasin



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado (Titulo del ensayo), presentado por el estudiante **Bernabé Chasin Jordy Joseph** fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 3%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTOR

Ing. Marjorie Coronel Suárez, Mgt.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, JORDY JOSEPH BERNABÉ CHASIN

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del trabajo de titulación con fines de difusión dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 12 días del mes de junio del año 2024.

EL AUTOR

A handwritten signature in black ink, appearing to read "Jordy Bernabé", is written over a light blue rectangular background.

Jordy Joseph Bernabé Chasin

AGRADECIMIENTO

Agradezco primeramente a Dios, quien me permitió seguir este largo proceso y nunca me dejó solo. A mis padres, por estar siempre presentes, dándome su apoyo incondicional. A mis hermanos, por siempre estar ahí motivándome a seguir adelante. A mis compañeros que conocí en la universidad, por siempre creer en mi potencial y en lo que puedo lograr. A la Ing. Marjorie Coronel, por la confianza que depositó en mí; le agradezco profundamente su guía, apoyo y paciencia a lo largo de este proceso. Su conocimiento y dedicación han sido fundamentales para la realización de este trabajo.

Jordy Joseph, Bernabé Chasin

DEDICATORIA

Dedico este trabajo principalmente a Dios, por guiarme en el camino correcto y no desfallecer en este largo proceso, permitiéndome así alcanzar este anhelado logro.

A mis queridos padres, Maurith Betsabeth Chasin González y Víctor Manuel Bernabé Ramírez, quienes me brindaron su apoyo y me enseñaron desde muy niño a luchar por todo lo que me proponga, siendo este mi principal motor para llegar a la meta.

A mi hermana, Janella Anahí Bernabé Chasin, y mi hermano, Gregory Ariel Bernabé Chasin, por siempre apoyarme y motivarme a culminar mi carrera universitaria.

A mis demás familiares, quienes confiaron en mis capacidades desde el día que empecé la carrera.

A mis amigos, quienes compartieron momentos muy especiales en mi vida, en especial a Nicolás Álvarez, quien es mi amigo desde el inicio de la carrera y siempre estuvo dispuesto a brindarme su ayuda.

Jordy Joseph, Bernabé Chasin

ÍNDICE GENERAL

TÍTULO DE ANTEPROYECTO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE DE TABLAS	XII
ÍNDICE DE ILUSTRACIONES	XIV
RESUMEN	XVI
ABSTRACT	XVII
INTRODUCCIÓN	1
1. Fundamentación	2
1.1 Antecedentes	2
1.2 Descripción del Proyecto.	4
1.3 Objetivos del proyecto	8
1.3.1 Objetivo general	8
1.3.2 Objetivos específicos	9
1.4 Justificación del proyecto	9
1.5 Alcance del proyecto.	11
1.6 Metodología del proyecto	12
1.6.1 Metodología de investigación	12
1.6.2 Beneficiarios del proyecto	13
1.6.3 Variables	13

1.6.4 Análisis de recolección de datos	13
1.7 Metodología de desarrollo	16
CAPÍTULO 2	17
2.Propuesta	17
2.1 Marco contextual	17
2.1.1 Ministerio de Inclusión Económica y Social.	17
2.1.2 Mapa organizacional de departamentos	18
2.2 Marco Conceptual.	19
2.2.1 ITIL V4	19
2.2.2. Tecnologías emergentes	21
2.2.3 Backend	23
2.2.4 Web Service	25
2.3 Marco Teórico	25
2.3.1 Innovaciones de ITIL 4: Revolucionando la Gestión de Incidentes	25
2.3.2 El poder del software libre.	26
2.2.3 Respuestas Inmediatas: Transformando la Atención al Cliente con Chatbots	26
2.4 Requerimientos	27
2.4.1 Requerimientos Funcionales	27
2.4.2 Requerimientos No Funcionales	31
2.5 Componentes de la propuesta	32
2.5.1 Arquitectura del Sistema	32
2.5.3 Comparativas de herramientas.	36
2.5.4 Diagramas de Casos de Uso	39
2.5.5 Modelado de datos.	48
2.6 Diseño de Interfaz	48

2.6 Pruebas	63
2.7 Resultados	73
CONCLUSIONES	76
REFERENCIAS	79
Anexo 1. Formato de entrevista.	83
Anexo 2. Árbol de Problemas	86
Anexo 3. Formato de encuesta de satisfacción	87

ÍNDICE DE TABLAS

Tabla 1 Población de Estudio	15
Tabla 2 Requerimientos Funcionales - Módulo de Seguridad.	27
Tabla 3 Requerimientos Funcionales - Módulo Registro de Incidentes.	27
Tabla 4 Requerimientos Funcionales - Módulo Seguimiento de Incidentes.	28
Tabla 5 Requerimientos Funcionales - Módulo de Seguimiento de Incidentes.	29
Tabla 6 Requerimientos Funcionales - Módulo de Gestión del Conocimiento.	29
Tabla 7 Requerimientos Funcionales – Módulo Cierre del Incidente.	30
Tabla 8 Requerimientos Funcionales - Módulo de Solicitar Reportes.	30
Tabla 9 Requerimientos Funcionales - Módulo Solicitar CSV.	31
Tabla 10 Requerimientos No Funcionales	32
Tabla 11 Comparativa entre Telegram y WhatsApp.	37
Tabla 12 Comparativa entre Zammad y Freshdesk.	38
Tabla 13 Comparativa entre Grafana y Datadog.	39
Tabla 14 Caso de uso - Acceso al sistema.	40
Tabla 15 Caso de uso – Proceso rol trabajador.	43
Tabla 16 Caso de uso – Proceso rol administrador.	46
Tabla 17 Caso de uso – General de uso del chatbot	48
Tabla 18 Prueba de funcionalidad – Identificar usuario con su rol.	64
Tabla 19 Prueba de funcionalidad – Informar Incidente.	65
Tabla 20 Generar un ticket en Zammad.	66
Tabla 21 Prueba de funcionalidad – Consultar Incidentes.	67
Tabla 22 Prueba de funcionalidad – Solicitar Reportes.	68
Tabla 23 Prueba de funcionalidad – Solicitar CSV.	69
Tabla 24 Prueba de funcionalidad – Cerrar Sesión del chatbot.	70

Tabla 25 Prueba de funcionalidad – Creación de usuarios para Ares.	71
Tabla 26 Prueba de funcionalidad – Tiempos de Resolución Ares.	72
Tabla 27 Recolección de datos – Tiempos de Resolución “Mesa de ayuda”	74
Tabla 28 Recolección de datos – Tiempos de Resolución “Ares y Zammad”	75

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Pasos a seguir para el desarrollo del asistente.	16
Ilustración 2 Departamentos del MIES	18
Ilustración 3 Mapa de ubicación del MIES.	18
Ilustración 4 Mapa de calor de la contribución del control de cambios a las actividades de la cadena de valor	21
Ilustración 5 Componentes del backend.	33
Ilustración 6 Diagrama del proceso de incidentes	34
Ilustración 7 Limpieza de datos - Script de cambio de fecha.	35
Ilustración 8 Limpieza de datos - Ajuste de campos.	36
Ilustración 9 Diagrama de caso de uso- acceso al sistema.	39
Ilustración 10 Diagrama de caso de uso - Proceso rol trabajador.	41
Ilustración 11 Diagrama de caso de uso - Proceso rol administrador.	43
Ilustración 12 Diagrama de caso de uso - General de uso del chatbot	46
Ilustración 13 Tablas principales para la integración.	48
Ilustración 14 Interfaz - Inicio de sesión	49
Ilustración 15 Interfaz trabajador.	50
Ilustración 16 Interfaz trabajador - Incidentes Reportados.	50
Ilustración 17 Interfaz trabajador - Informar Incidente.	51
Ilustración 18 Interfaz trabajador - Ver solución.	52
Ilustración 19 Interfaz trabajador - Incidente Resuelto.	52
Ilustración 20 Interfaz trabajador - Creación de Tickets	53
Ilustración 21 Interfaz Zammad.	54
Ilustración 22 Interfaz administrador.	54
Ilustración 23 Interfaz administrador - Gestionar.	55

Ilustración 24 Interfaz administrador – Reportes.	55
Ilustración 25 Enlace de reportes de Grafana.	56
Ilustración 26 Interfaz Grafana – Inicio de sesión.	56
Ilustración 27 Interfaz Grafana - Reportes.	57
Ilustración 28 Interfaz administrador - Archivo CSV.	58
Ilustración 29 Interfaz cierre de sesión.	58
Ilustración 30 Interfaz DialogFlow.	59
Ilustración 31 Bot de notificaciones.	60
Ilustración 32 Comunicación Webhook con DialogFlow.	60
Ilustración 33 Solicitud a la API de ChatGpt.	61
Ilustración 34 Comunicación con Zammad.	62
Ilustración 35 Creación de Usuarios para el chatbot.	63

RESUMEN

El presente proyecto trata sobre el desarrollo de un asistente virtual para la gestión de incidentes de TI haciendo uso del marco de trabajo ITIL v4 y tecnologías emergentes que se encuentran presente en el ámbito digital actual, para ello se usó como referencia la extensión de la entidad gubernamental de nombre Ministerio de Inclusión Económica y Social (MIES) ubicada en el cantón Salinas, lugar donde diariamente se reportan incidentes de primer nivel y que su resolución conlleva de un largo proceso.

El asistente virtual, denominado “Ares”, se desarrolló como un chatbot en la plataforma de mensajería Telegram, complementado con un sistema de gestión de bases de datos y servicios web necesarios para su funcionamiento. Además, se integró con Zammad, un software de gestión de tickets de código abierto, para optimizar el manejo de incidencias.

El proyecto consta de cinco fases: Identificación, Planificación, Diseño, Control. Gestión y su debida ejecución. Se consiguió determinar la funcionalidad de “Ares” mediante pruebas y reportes que dieron como resultado la reducción de tiempo desde el reporte del incidente hasta su debida asignación y resolución.

Palabras claves: chatbot, incidentes, ITIL v4, gestión de incidentes, Ares, tecnologías emergentes.

ABSTRACT

This project is about the development of a virtual assistant for IT incident management using the ITIL v4 framework and emerging technologies that are present in the current digital environment, for this we used as a reference the extension of the government entity named Ministry of Economic and Social Inclusion (MIES) located in the canton Salinas, a place where first level incidents are reported daily and their resolution involves a long process.

The virtual assistant, called "Ares", was developed as a chatbot on the Telegram messaging platform, complemented with a database management system and web services necessary for its operation. It was also integrated with Zammad, an open source ticket management software, to optimize incident management.

The project consists of five phases: Identification, Planning, Design, Control. Management and its proper execution. The functionality of "Ares" was determined through tests and reports that resulted in the reduction of time from the incident report to its proper assignment and resolution.

Keywords: chatbot, incidents, ITIL v4, incident management, emerging technologies.

INTRODUCCIÓN

En la actualidad, los chatbots se han vuelto esenciales para agilizar procesos en las empresas, especialmente en la resolución de incidentes. La implementación de prácticas de gestión como ITIL asegura una gestión eficiente y efectiva de estos incidentes. El objetivo de este proyecto es desarrollar un asistente virtual tipo chatbot que facilite la gestión de incidencias en el Ministerio de Inclusión Económica y Social (MIES). Esta entidad gubernamental se encarga de proporcionar servicios de ayuda a niños, niñas, adolescentes y adultos mayores. Sin embargo, enfrenta desafíos al momento de resolver tickets debido a la falta de personal en el área de tecnologías de la información y comunicación (TIC), así como la necesidad de monitorear constantemente el sistema.

Para alcanzar este objetivo, se utilizaron herramientas de software libre como Zammad, un sistema de gestión de tickets que permite organizar y priorizar las incidencias de manera eficiente. Además, se implementaron herramientas de monitoreo y creación de informes como Grafana y Elasticsearch, que proporcionan una visualización clara y en tiempo real de los datos relevantes para la toma de decisiones

Se emplearon también plataformas de mensajería como Telegram para asegurar que el chatbot estuviera accesible a los usuarios en todo momento. Por otro lado, se utilizaron tecnologías avanzadas como Dialogflow para la comprensión del lenguaje natural y la creación de respuestas contextualmente relevantes, así como Node.js y Express para el desarrollo del backend del chatbot, aprovechando las capacidades de JavaScript para una implementación rápida y escalable.

La integración del chatbot con Zammad fue un componente clave del proyecto, permitiendo la creación automática de tickets para incidentes que no pudieron ser resueltos por el chatbot. Esto garantiza que todos los problemas sean registrados y gestionados de acuerdo con los estándares de ITIL v4.

En conclusión, este proyecto tiene como objetivo desarrollar un chatbot de gestión de incidencias que se integre con tecnologías emergentes y logre resolver estos incidentes siguiendo el estándar ITIL v4.

CAPÍTULO 1

1. Fundamentación

1.1 Antecedentes

En la actualidad, el crecimiento de las demandas en los servicios de Tecnologías de la Información (TICS) por parte de las empresas ha generado la necesidad de realizar un análisis de la situación actual. Este análisis busca identificar problemas relacionados con la gestión de incidentes, los cuales han dado origen a situaciones de desorganización y a la falta de respuestas inmediatas frente a cuestiones técnicas, organizacionales y otros aspectos que afectan a toda la entidad. [1]

El Ministerio de Inclusión Económica y Social (MIES) es una entidad gubernamental ubicada en el cantón Salinas de la provincia de Santa Elena, fundada en 1979 con el propósito de elevar la calidad de atención de los servicios dirigidos a poblaciones diversas, que incluyen niñas, niños, adolescentes, jóvenes, adultos mayores y personas con discapacidad. A lo largo de los años, el MIES ha desempeñado un papel fundamental en la promoción de la inclusión social y económica en nuestra sociedad, brindando servicios esenciales a diversos grupos de población. [2]

El área de TICS se encarga de la gestión y mantenimiento de los sistemas y recursos tecnológicos. Su responsabilidad abarca desde la administración de la infraestructura informática hasta la implementación de soluciones tecnológicas. Sin embargo, a través de la entrevista el encargado del área (Ver anexo 1) dio como resultado como en ocasiones el área se encuentra saturada, ya que solo cuenta con una persona encargada para atender las solicitudes de numerosos usuarios, lo que dificulta la atención inmediata ante problemas y solicitudes de asistencia. Esta situación puede llevar a retrasos en la resolución de problemas y a una disminución en la eficiencia de los servicios tecnológicos, afectando la productividad de la organización.

La carencia de herramientas adecuadas de seguimiento de incidentes ha dado lugar a un proceso desafiante y poco eficiente en la gestión de problemas de TICS. Sin una plataforma centralizada para registrar, categorizar y dar seguimiento a los incidentes, el personal de soporte se ve obligado a depender de sistemas manuales y registros dispersos, lo que dificulta la identificación de patrones recurrentes y la asignación

adecuada de recursos. Esta falta de visibilidad y control afecta directamente la capacidad de la organización para brindar respuestas rápidas y efectivas a los incidentes.

Un obstáculo fundamental de la gestión de tecnologías es el retraso significativo en la obtención de respuestas técnicas para la resolución de incidentes. Estos retrasos no solo impactan negativamente la eficiencia operativa, sino que también afectan la satisfacción de los usuarios y la continuidad de los servicios. La demora en la obtención de soluciones técnicas puede resultar en una mayor interrupción de los procesos empresariales y en un aumento de la frustración de los usuarios, lo que pone de relieve la necesidad apremiante de abordar esta cuestión central.

Por otro lado, la falta de comunicación efectiva y de un sistema formal de seguimiento de incidentes conlleva a que los usuarios no reciban respuestas oportunas a sus necesidades y experimenten retrasos frustrantes en la resolución de problemas técnicos. Esta situación afecta negativamente la experiencia del usuario y puede tener un impacto en la productividad y eficiencia de la organización en su conjunto. Abordar esta problemática es esencial para mejorar la satisfacción del usuario y la calidad de los servicios de TICS ofrecidos.

En la Universidad de Guayaquil, Juan Martínez Carpio realizó su proyecto de titulación titulado "Desarrollo de un Asistente Virtual (Chatbot) para la Automatización de la Atención al Cliente". El propósito fundamental de este proyecto es mejorar significativamente los niveles de atención al cliente en un contexto en el que la eficiencia en la respuesta a las consultas se ha convertido en una necesidad crítica. La justificación de esta iniciativa se basa en la creciente demanda de los usuarios de recibir respuestas rápidas y efectivas a sus preguntas y requerimientos, lo que refleja la importancia de optimizar los procesos de atención al cliente mediante soluciones tecnológicas innovadoras como el asistente virtual. [3]

En su proyecto de titulación llamado "Implementación de un Asistente Virtual para la Atención al Cliente en Electrocentro S. A. de Huancayo", Edwing Cristian Artica Llacta propuso la implementación de un asistente virtual con el objetivo de ofrecer un servicio de atención óptimo y ágil a los clientes de la empresa Electrocentro S.A. Este asistente virtual se diseñó con la finalidad de facilitar la realización de consultas,

gestionar reclamos, proporcionar información relevante y brindar asistencia en diversas áreas de interacción con los clientes. La iniciativa se sustenta en la creciente importancia de la atención al cliente eficiente y personalizada en el entorno empresarial actual, con el propósito de mejorar la satisfacción del cliente y fortalecer la relación con la empresa. [4]

Por su parte, en el proyecto desarrollado por Luis Enrique Cubero Caba, se llevó a cabo la implementación de un asistente virtual en el sitio web de la Facultad de Informática. El propósito principal de este asistente virtual radica en simplificar la búsqueda de información para los usuarios dentro del dominio de la facultad mediante interacciones conversacionales con el chatbot. Esta iniciativa se orienta hacia la mejora de la experiencia de búsqueda en el sitio web y la consecuente elevación de la calidad del servicio ofrecido a la comunidad académica y visitantes. La implementación de un asistente virtual responde a la creciente demanda de soluciones tecnológicas que agilicen la obtención de información y proporcionen una experiencia de navegación más eficiente y efectiva en el entorno digital de la facultad. [5]

Según lo detallado con anterioridad, la implementación de un asistente virtual se presenta como una solución estratégica y prometedora para abordar los desafíos previamente mencionados en la gestión de TICS, como los retrasos en la resolución de incidentes y la insatisfacción de los usuarios. Este asistente virtual actúa como un intermediario eficaz entre el personal de TICS y los usuarios, proporcionando respuestas técnicas rápidas y precisas a las consultas comunes y los problemas técnicos recurrentes, garantizando así que los requerimientos se aborden de manera más eficiente y mejorando la calidad general de los servicios.

1.2 Descripción del Proyecto.

El MIES actualmente no dispone de un sistema que permita abordar las incidencias de TICS de manera ágil y efectiva. A pesar de contar con una mesa de ayuda, esta no garantiza una respuesta rápida a las incidencias, ya que depende de la supervisión constante del encargado del área de TICS, lo cual puede resultar complicado debido a que esa responsabilidad recae en una sola persona. Por esta razón, se propone el desarrollo de un chatbot que facilite la gestión de las incidencias.

Esta iniciativa tiene como objetivo mejorar la capacidad de respuesta ante las incidencias de TICS de primer nivel y garantizar una atención más eficiente, eliminando la dependencia de la supervisión constante por parte de un único responsable. El chatbot proporcionará una solución automatizada que contribuirá a gestionar y resolver las incidencias de manera más rápida y efectiva, lo que beneficiará a toda la organización al reducir tiempos de inactividad y mejorar la calidad del servicio de atención al usuario.

Information Technology Infrastructure Library (ITIL) es una biblioteca de mejores prácticas utilizada para gestionar y mejorar los servicios y el soporte de TICS. La aplicación de los principios de ITIL garantiza la rápida identificación de las causas subyacentes de los problemas en el entorno, al mismo tiempo que proporciona una visión completa de los sistemas y las personas involucradas, con el fin de prevenir futuros contratiempos de manera eficaz. [6]

En el marco de este proyecto, se llevará a cabo el desarrollo de un chatbot diseñado exclusivamente para la resolución de incidentes de TICS de primer nivel, siguiendo los principios y las mejores prácticas definidas en ITILV4. Además, para la gestión de tickets, se implementará Zammad, un software libre especializado en el seguimiento y control de tickets, asegurando una integración sin problemas con el entorno de trabajo del Ministerio de Inclusión Económica y Social (MIES)."

Este chatbot estará implementado en la plataforma de mensajería Telegram y se estructurará en ocho módulos específicos los cuales incluyen:

MÓDULO DE SEGURIDAD

- Inicio de sesión rol trabajador.
- Inicio de sesión rol administrador.

ROL TRABAJADOR MIES

MÓDULO DE REGISTRO DE INCIDENTES:

- Permitir a los usuarios registrar incidentes de TICS.
- Proporciona campos para informar el incidente que tiene el usuario.
- Crear ticket
- Asignar título al ticket.

MÓDULO DE SEGUIMIENTO DE INCIDENTES:

- Proporcionar actualizaciones sobre el estado de los incidentes a usuarios y personal de soporte.
- Facilitar una comunicación eficaz para mantener a todas las partes informadas acerca de los incidentes.

MÓDULO DE RESOLUCIÓN DE INCIDENTES:

- Implementar la automatización en la solución de incidentes de primer nivel, enfocándose en los estándares de ITIL V4.
- Consultar una solución de los incidentes de primer nivel haciendo uso de la Api de Openai.

MÓDULO DE GESTIÓN DE CONOCIMIENTO:

- Guardar la solución de un incidente en la base de conocimientos.
- Consultar, información de la base de conocimientos.

MÓDULO CIERRE DEL INCIDENTE:

- Confirmar que el incidente se ha resuelto.
- Actualizar el estado del incidente.
- Generar un mensaje de éxito para el cierre del incidente.

ROL ENCARGADO DE TICS

MODULO SOLICITAR REPORTE

- Enviar reportes diarios.
- Enviar reportes mensuales.
- Enviar un enlace de acceso a Grafana.

MODULO SOLICITAR CSV

- Enviar archivo csv de incidentes.

A continuación, se detallan las herramientas tecnológicas que se emplearán en el desarrollo del proyecto:

Visual Studio Code: es un editor de código desarrollado por Microsoft, está disponible en Windows, Linux, y macOS. Es una herramienta ampliamente personalizable y cuenta con un rico ecosistema de extensiones en distintos lenguajes. [7]

DialogFlow: es una destacada plataforma de desarrollo de chatbots y asistentes virtuales creada por Google, esta plataforma brinda la capacidad de diseñar interfaces de conversación natural para aplicaciones y servicios de manera eficiente, facilitando así la creación de experiencias de usuario conversacionales de alta calidad. [8]

PostgreSQL: es un sistema de gestión de bases de datos relacional de código abierto, con 35 años de desarrollo activo, es conocido por su robustez, capacidad de extensión y cumplimiento de los estándares SQL. [9]

Render: es una plataforma de alojamiento en la nube que ofrece servicios de alojamiento web y aplicaciones, permite a los desarrolladores implementar, administrar aplicaciones web y sitios de manera escalable. [10]

GitHub: es una plataforma de desarrollo colaborativo que se basa en el sistema de control de versiones Git, esta plataforma es ampliamente reconocida por su capacidad para alojar proyectos de software, facilitar la gestión de control de versiones y promover la colaboración en proyectos de código abierto, además de su función principal como un repositorio de código. [11]

Zammad: Es un software de código abierto que es utilizado para la gestión de tickets, ofrece una amplia gama de funciones para optimizar la administración de los tickets. Este sistema, centrado en la transparencia y personalización, permite realizar diversas acciones que resultan muy prácticas en el ámbito de la gestión de los tickets. [12]

Node js: es una plataforma de desarrollo de código abierto en JavaScript que sobresale por su capacidad para instalar módulos y dependencias en proyectos, además es ampliamente conocida por su versatilidad en la creación de aplicaciones de red altamente escalables. [13]

Express js: es una infraestructura o framework de backend destinada para aplicaciones web es la más popular de la librería Node.js destaca por su flexibilidad, express proporciona al desarrollador características de aplicación web y móviles básicas. Su

eficiencia y escalabilidad lo convierten en una opción ideal para construir aplicaciones de alta performance, mientras que su amplia comunidad asegura soporte continuo y abundantes recursos. [14]

JavaScript: es un lenguaje de programación ampliamente empleado en el desarrollo web. Se destaca por su versatilidad y es especialmente reconocido por su capacidad para crear páginas web interactivas y dinámicas que mejoran significativamente la experiencia del usuario. Además, su comunidad activa proporciona un soporte constante y recursos abundantes para desarrolladores de todos los niveles. [15].

Telegram: es una aplicación de mensajería simple y gratuita que tiene enfoque en la velocidad y seguridad, esta aplicación de mensajería está diseñada para ofrecer una experiencia rápida y segura de intercambio de mensajes, imágenes, archivos, etc. [16]

API ChatGPT: La API de OpenAI proporciona acceso a potentes modelos de lenguaje y aprendizaje automático, permitiendo a los desarrolladores crear aplicaciones más inteligentes y sofisticadas. Con una documentación clara y ejemplos de código, su integración se vuelve aún más sencilla y eficiente. [17].

Grafana: es una herramienta Open Source que permite visualizar datos de serie temporales, permite crear cuadros de mandos a través de múltiples fuentes, es muy popular y conocida por la capacidad de crear gráficos y dashboard mediante la transformación de datos en información en tiempo real [18].

Elastic Search: es un motor de búsqueda y analítica, facilita la recopilación, agregación y enriquecimiento de datos, esta herramienta proporciona búsqueda y análisis casi en tiempo real para todo tipo de datos permitiendo a las organizaciones tomar decisiones informadas rápidamente [19].

1.3 Objetivos del proyecto

1.3.1 Objetivo general

Desarrollar un asistente virtual para la gestión de incidentes de tecnologías de la información, basado en el marco de referencia ITIL V4 que contribuya con la calidad de resolución de incidentes de los servicios tecnológicos del MIES.

1.3.2 Objetivos específicos

- Seleccionar las herramientas de código abierto adecuadas para el desarrollo del asistente virtual, teniendo en cuenta criterios como su flexibilidad y su alineación con los principios de ITIL V4.
- Desarrollar el asistente virtual el cual abarcará su infraestructura y sus módulos claves para el cumplimiento de la resolución de incidentes de TICS.
- Realizar pruebas de funcionamiento para ver la capacidad de respuesta del asistente.
- Integrar el chatbot de incidencias con el software de gestión de tickets Zammad para agilizar la administración y resolución de incidentes que no fueron resueltos automáticamente por el asistente virtual.

1.4 Justificación del proyecto

En la actualidad, el uso de tecnologías desempeña un papel crucial en el funcionamiento de una empresa, ya que no solo facilita y agiliza las operaciones internas y externas, sino que también abre un abanico de oportunidades para la innovación y el crecimiento. El desarrollo de un chatbot ayudará a reducir los tiempos de respuesta a los incidentes que ocurren a diario en el MIES, con el objetivo de garantizar una respuesta rápida a las necesidades de los usuarios y ofrecer un servicio que mejore la atención que se les brinda.

ITIL v4 ofrece un conjunto de 34 prácticas de gestión que impactan positivamente en la operación de las empresas, la gestión de incidentes en particular destaca al reducir el impacto de problemas operativos a través de la identificación, notificación, análisis y resolución de incidentes [20]. Este enfoque no solo optimiza la respuesta ante situaciones adversas, sino que también proporciona una base sólida para diseñar procesos de gestión de incidentes más eficientes. Los beneficios incluyen la creación de plantillas personalizadas para capturar información relevante y la implementación de mejores prácticas que agilizan la resolución de problemas.

El desarrollo de un chatbot para la gestión de incidencias de TICS en el MIES beneficiará significativamente a la institución, ya que ayudará a abordar los incidentes de primer nivel de manera óptima y eficiente, contribuyendo a una notable reducción en los tiempos de respuesta ante dichas incidencias. Con el objetivo de fortalecer aún

más esta implementación, el as chatbot usará el marco de trabajo propuesto por ITIL v4, lo cual garantizará una gestión correcta y eficaz de los incidentes, además de mejorar sustancialmente la satisfacción de los usuarios.

Además, el chatbot, al estar disponible en toda la jornada laboral, garantiza una atención constante, lo que significa que los usuarios pueden obtener ayuda y resolver problemas en cualquier momento, lo que se alinea perfectamente con los principios de ITIL v4 de proporcionar un servicio de TICS ágil y orientado al cliente. Al automatizar tareas repetitivas y ofrecer respuestas inmediatas, el chatbot permite que el personal de TICS se concentre en la mejora continua y en la optimización de procesos, lo que contribuye aún más a la eficiencia en la gestión de incidencias y a la entrega de un servicio de mayor calidad.

El uso del marco de referencia de mejores prácticas ITIL junto con el chatbot, ayudará a contribuir en el aumento de la eficiencia de los usuarios, al poder hacer uso del chatbot permitirá a los usuarios alertar y solucionar los incidentes de manera rápida, con esto se evitará la práctica ineficiente de tener que solicitar ayuda al encargado del área de TICS por incidentes básicos. Esto tendrá un impacto significativo, no solo para los usuarios sino también para le encargado del área, ya que le ayudará a tener controlado los incidentes que se generan diariamente, y no desperdiciar su tiempo en asuntos que se pueden resolver fácilmente con la ayuda del chatbot.

El tema propuesto está alineado a los objetivos del Plan de Creación de Oportunidades 2021-2025, específicamente en:

Objetivos del eje social

Objetivo 5. Proteger a las familias, garantizar sus derechos y servicios, erradicar la pobreza y promover la inclusión social [21].

Políticas.

5.5 Mejorar la conectividad digital y el acceso a nuevas tecnologías de la población [21].

Pol. 5.4.

A4. Fortalecer la conectividad y el acceso a las TIC como una vía para mejorar el acceso a otros servicios [21].

1.5 Alcance del proyecto.

Se desarrollará un chatbot que ayudará a resolver las incidencias de primer nivel que se presentan diariamente en el MIES. Este chatbot, implementado como un chatbot, será de gran ayuda tanto para los trabajadores como para el encargado del área de TICS, y funcionará en la plataforma de mensajería Telegram. Estará diseñado para abordar y solucionar los incidentes frecuentes que ocurren a diario.

El marco de trabajo seleccionado para esta implementación será ITIL V4, reconocido por sus mejores prácticas en la gestión de servicios de tecnología de la información. Este enfoque proporcionará una sólida base para la definición de procesos, roles y responsabilidades en la gestión de incidentes. Además, el chatbot también estará integrado con Zammad, lo que permitirá una gestión más completa de las incidencias. En los casos en que el chatbot no pueda resolver una incidencia de primer nivel, se iniciará automáticamente un proceso de creación de ticket. Esta integración con Zammad asegurará que todas las incidencias se registren de manera centralizada y que aquellas que requieran una atención más especializada sean gestionadas por el equipo de soporte de TICS.

El desarrollo del chatbot contará con ocho módulos distribuidos por su respectivo rol al iniciar sesión. El Módulo "Seguridad", aquí el usuario deberá loguearse por medio de su cedula de identidad, para que el chatbot identifique el rol que cumple en la organización, los módulos presentados a continuación corresponden al rol trabajador, módulo "Registro de incidentes" es aquí donde el usuario podrá reportar el problema junto con su descripción. Módulo "Seguimiento de Incidentes", donde se proporcionan actualizaciones automáticas sobre el estado de los incidentes y se facilita la comunicación para mantener a las partes informadas.

Además, también contará con el módulo "Resolución de Incidentes", en este se implementa la automatización en la resolución de incidentes de primer nivel, basándose en estándares de ITIL V4 y donde se obtiene la respuesta a los incidentes mediante el uso de la API de ChatGpt. Módulo "Gestión de Conocimiento" aquí se guardará la información que cumplió con la resolución de un incidente, se crea, consulta, información en la base de conocimientos, y se alimentará de la información exitosa de incidentes para usarla en un futuro.

Por último, el módulo del rol trabajador es "Cierre de Incidentes", donde se confirma que el incidente se ha resuelto, se actualiza su estado, se guarda la información que fue útil para alimentar a la base de conocimientos y se genera un mensaje de éxito sobre cierre del incidente. Este módulo tiene una etapa adicional, si el incidente se resuelve con éxito mediante el chatbot, se registrará como "resuelto" y se cerrará el proceso. Caso contrario, se iniciará la etapa de envío hacia Zammad, para lo cual el chatbot creará un ticket, que estará a la espera de la recepción y posterior asignación de un encargado del área.

Por lo consiguiente el rol administrador cuenta con dos módulos, el primer módulo se denomina "Solicitar Reportes" aquí el administrador podrá solicitar informes por medio de un botón que ofrece el chatbot, estos reportes pueden ser mensuales o diarios, también el chatbot nos brinda un link para poder ver el reporte completo haciendo uso de Grafana. El segundo y último modulo para este rol se llama "Solicitar csv" en esa sección el chatbot le proporcionará al administrador todos los registros de los incidentes para que puedan ser usados a futuro.

Por ende, el desarrollo del chatbot basado en la gestión de incidentes de TICS según ITIL V4 ofrece una sólida estructura para aplicar directrices y mejoras tecnológicas a la situación actual. Esto permite abordar soluciones eficientes y precisas conforme a lo estipulado en los incidentes previstos en el chatbot. En consecuencia, este proyecto de chatbot propuesto se enmarca en la correcta administración de resoluciones de eventos que surgen diariamente en el MIES.

1.6 Metodología del proyecto

1.6.1 Metodología de investigación

En el presente proyecto, se utilizó la metodología de investigación denominada "estudio exploratorio" [22] con el objetivo de complementar y profundizar el conocimiento sobre el proyecto a realizar. Para ello, se llevó a cabo la revisión de diversos trabajos relacionados con el tema del proyecto a nivel nacional, local e internacional. A partir de esta revisión, se determina la viabilidad y factibilidad del proyecto, guiándose por la obtención de resultados favorables que garanticen su correcto funcionamiento en la empresa. Esta fase de investigación exploratoria ha sido fundamental para el proceso de toma de decisiones, asegurando que el proyecto esté

respaldado por evidencia sólida y se enfoque en las mejores prácticas a nivel global, lo que contribuirá al éxito del emprendimiento.

Por consiguiente, mediante el uso de la metodología diagnóstica [22] se logró conocer el proceso que siguen habitualmente los incidentes en el MIES diariamente, y se determinó que los incidentes no son gestionados, procesados y seguidos de la mejor manera. Este diagnóstico revela una oportunidad invaluable para implementar el chatbot, ya que permitirá mejorar la eficiencia y la calidad en la atención de las incidencias, lo que en última instancia contribuirá a un funcionamiento óptimo de la institución.

1.6.2 Beneficiarios del proyecto

El principal beneficiario de este proyecto es el encargado de TICS, ya que el chatbot ayudará a reducir los tiempos de respuesta y permitirá que el encargado pueda utilizar el tiempo que habitualmente se invierte en resolver incidencias, tanto las simples como aquellas que requieren una mayor inversión de tiempo y concentración, en otras actividades críticas.

1.6.3 Variables

Se medirá el tiempo en el que se atienden los incidentes de TI, tomando en consideración variables clave como el registro inicial del incidente y el proceso de resolución. Esta medición permitirá evaluar la eficacia del chatbot en la agilización de la gestión de incidentes, garantizando una respuesta más rápida, lo que contribuirá a optimizar la productividad de los equipos de TICS y a mejorar la experiencia de los usuarios.

1.6.4 Análisis de recolección de datos

Para la recolección de la información del MIES se optó por la técnica de entrevista realizada al encargado del área de TICS, como se detalla en el ([anexo1](#)). La elección de esta técnica se fundamenta en la amplia experiencia del encargado en la gestión de los procesos tecnológicos del MIES, lo que le otorga un conocimiento importante sobre las operaciones y sistemas que se llevan a cabo en la institución.

En la entrevista realizada al encargado del área de, quien es el único responsable de los procesos que se llevan a cabo en dicha área, se realizó un análisis detallado. Se

pudo observar que el MIES tiene un enfoque centrado en la resolución rápida de incidentes para minimizar cualquier impacto en la institución. Además, se implementan normas y políticas de seguridad con el fin de reducir al mínimo los incidentes de seguridad. Sin embargo, el MIES enfrenta un gran desafío, ya que la falta de personal en el área de TICS dificulta la atención oportuna de estos incidentes. El tiempo de atención depende en gran medida de las escalas de prioridad y, en particular, de la necesidad constante de consultar a la mesa de ayuda

Debido a que solo una persona se encarga del área de TICS, esta tarea involucra una amplia gama de responsabilidades, que incluyen el soporte de usuarios, la resolución de problemas de infraestructura, la escalada manual de incidentes según niveles jerárquicos, la elaboración de informes de incidentes, el análisis de equipos, entre otras. Esto conlleva a que el encargado se encuentre ocupado en todo momento, lo que, en algunos casos, provoca que los incidentes deban esperar para su resolución. Esta carga de trabajo intensiva puede generar demoras en la atención de incidentes, lo que puede impactar la eficiencia y la satisfacción del usuario final.

El proceso típico utilizado para gestionar las incidencias en el MIES comienza con un diagnóstico inicial del incidente. Luego, se le asigna un nivel o escala jerárquica para clasificarlos en orden de prioridad, que pueden ser bajo, normal o alto. Además, se implementa una herramienta llamada "Mesa de Ayuda" en la que cada usuario registra su incidente y espera la atención correspondiente. Sin embargo, el problema con esta herramienta es que el encargado debe estar constantemente monitoreándola para poder notar cuando llegan los incidentes. Como se mencionó anteriormente, la falta de personal hace que no siempre se pueda mantener un monitoreo constante de esta herramienta, lo que resulta en tiempos de resolución prolongados para incidentes sencillos

También detalló que, en muchas ocasiones, los incidentes básicos son resueltos al siguiente día debido a la carga de trabajo y la necesidad de monitoreo continuo. El promedio de resolución de incidentes puede variar significativamente dependiendo de la frecuencia y la efectividad del monitoreo de la Mesa de Ayuda. En términos generales, los incidentes de baja prioridad pueden tardar entre 20 a 24 horas en resolverse, mientras que los de prioridad normal suelen ser resueltos en un plazo de 8

a 15 horas. Por otro lado, los incidentes de alta prioridad reciben atención inmediata y suelen ser resueltos en un promedio de 2 a 4 horas.

Por último, se detalló que el MIES no tiene la costumbre de realizar simulacros. Cuando se implementa una nueva herramienta tecnológica, se lleva a cabo un plan piloto y su respectiva capacitación. Este punto hace referencia a que, en caso de que se llegara a implementar un chatbot, se deben realizar pruebas y verificaciones para identificar posibles problemas y vulnerabilidades

Para garantizar la representatividad y exhaustividad de los resultados, se optó por incluir a los cinco directivos principales de cada área, conformando así una muestra que abarca la totalidad de la población objetivo. Este enfoque censal permite obtener una comprensión integral y detallada de las opiniones de los directivos (ver anexo 3), asegurando que ningún individuo perteneciente a la población objetivo quede excluido de la investigación, tal y como se detalla en la tabla 1.

<i>Departamento</i>	<i>Total, Encuestado</i>
<i>Unidad de Asesoría Jurídica</i>	1
<i>Unidad de Inclusión Social</i>	1
<i>Unidad Administrativa Financiera</i>	1
<i>Dirección Distrital</i>	1
<i>Unidad de Inclusión Económica</i>	1
TOTAL	5

Tabla 1 Población de Estudio

En la encuesta realizada a los jefes de áreas ([ver anexo 3](#)), indicaron que es complicado reportar un incidente. También mencionaron que los rangos de tiempo en los que se tarda en responder a un incidente son entre 2-4, 8-15, y 20-24 horas. Los jefes manifestaron estar insatisfechos con los tiempos de resolución, ya que son demasiado prolongados. Asimismo, expresaron su descontento con la comunicación del encargado de la mesa de ayuda y con el proceso actual que lleva el MIES para la gestión de las incidencias. Estos problemas resaltan la necesidad urgente de mejorar los procedimientos y herramientas utilizadas actualmente para la gestión de incidentes que ocurren a diario.

1.7 Metodología de desarrollo

Para el desarrollo del chatbot en el MIES, se adaptará el proyecto a la metodología ITILv4 con la gestión de incidentes con el objetivo de mejorar y optimizar los procesos que brinda el área de TI, por lo que se detalla de la siguiente manera [22].

- **Identificación:** En esta fase, se identifican los procesos de gestión de incidencias y se establecen los requisitos necesarios para la creación del chatbot. Esto implica una comprensión profunda de las necesidades y objetivos de la institución en cuanto a la resolución de incidencias.
- **Planificación:** Se realiza un análisis del estado actual de la institución para recopilar información relevante. Luego, se crea un plan detallado que define las actividades a realizar y determina el alcance del chatbot de gestión de incidencias. Esta etapa es crucial para garantizar que el proyecto cumpla con los objetivos establecidos.
- **Diseño:** Se llevan a cabo las actividades planificadas para diseñar el chatbot. Se abordan los objetivos establecidos y se determina la arquitectura y la interfaz, incluyendo la creación de diagramas de flujo de conversación para guiar la interacción con los usuarios.
- **Control y Gestión:** A lo largo del desarrollo del chatbot, se implementa un proceso constante de monitoreo y seguimiento. Se evalúa la funcionalidad a través de pruebas y se gestionan los procesos para realizar ajustes y mejoras en caso necesario. Esta etapa es esencial para garantizar la eficacia y calidad del producto final.
- **Ejecución:** Se realiza la ejecución del chatbot de gestión de incidencias, asegurando que se cumplan los objetivos previamente establecidos. Además, se documentan y analizan los resultados obtenidos como parte del proceso de mejora continua.



Ilustración 1 Pasos a seguir para el desarrollo del chatbot.

CAPÍTULO 2

2.Propuesta

2.1 Marco contextual

2.1.1 Ministerio de Inclusión Económica y Social.

Es una entidad gubernamental encargada de promover y fomentar la inclusión en la población, asegurando el logro de una adecuada calidad de vida para todos los ciudadanos, también ayuda a promover la atención integral de la población a lo largo de su ciclo de vida haciendo énfasis en los niños, niñas, adolescentes y adultos mayores, que viven en situación de exclusión, discriminación, pobreza o vulnerabilidad. [2]

Visión.

Ser la entidad pública líder en la inclusión social para los grupos de atención prioritaria y aquellos que se encuentran en situación de pobreza para aportar a su movilidad social [23].

Misión.

Establecer y ejecutar políticas, regulaciones, programas y servicios para la inclusión social y atención al ciclo de vida de niños, niñas, adolescentes, jóvenes, adultos mayores, personas con discapacidad y aquellos que se encuentran en situación de pobreza, a fin de aportar a su movilidad social y salida de la pobreza [23].

Principios.

La gestión del MIES se sustentará en los siguientes principios [23]:

- Universalidad
- Progresividad
- Inclusión
- Cohesión
- Equidad
- Igualdad en la diversidad
- Solidaridad

- Participación
- Organización social

2.1.2 Mapa organizacional de departamentos

En la ilustración 2 se aprecia la distribución departamental, donde la Dirección Distrital funge como ente rector encargado de supervisar las operaciones de las demás áreas y garantizar el cumplimiento cabal de los objetivos organizacionales. Cabe destacar que el área de Tecnologías de la Información se encuentra adscrita al área de Unidad Económica y Social, desempeñando un rol estratégico en el soporte tecnológico de los procesos inherentes a dicha unidad.

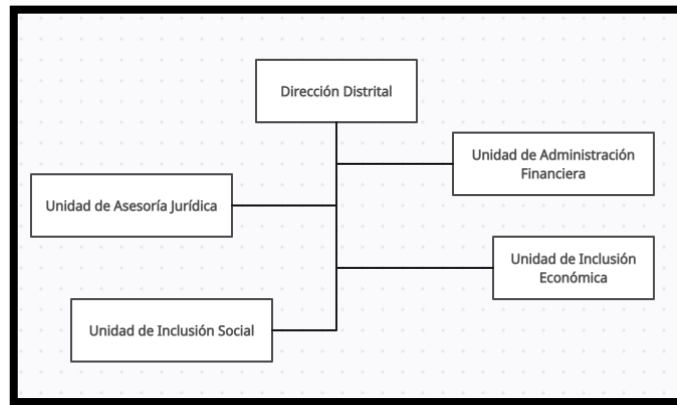


Ilustración 2 Departamentos del MIES

Ubicación Sectorial.

Según se refleja gráficamente en la ilustración 3, la ubicación sectorial de las instalaciones donde se alberga el proyecto en cuestión se ubica en las inmediaciones de la Avenida Carlos Espinoza, colindando con el recinto educativo INNOVA.

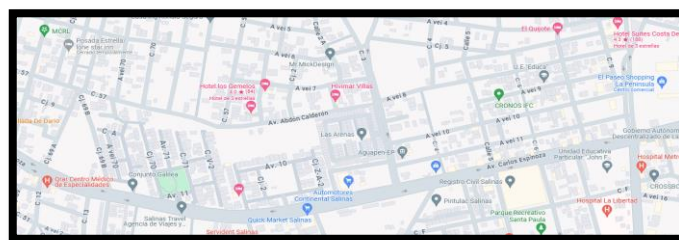


Ilustración 3 Mapa de ubicación del MIES.

2.2 Marco Conceptual.

2.2.1 ITIL V4

ITIL v4 proporciona un conjunto de 34 prácticas de gestión diseñadas para mejorar la operación de las empresas. Entre estas prácticas, la gestión de incidentes se destaca por su capacidad para mitigar el impacto de problemas operativos al facilitar la identificación, notificación, análisis y resolución oportuna de incidentes. Esta metodología ofrece un marco sólido para garantizar la continuidad del servicio y la satisfacción del cliente al abordar de manera eficiente los contratiempos en el entorno operativo de una organización [20].

2.2.1.1 Incidente.

El libro titulado ITIL 4 define a un incidente como una interrupción no planificada de un servicio o una reducción de la calidad del servicio. Esto significa que cualquier problema que afecte el funcionamiento normal de un servicio se considera un incidente. Ya sea una caída del sistema o un desempeño más lento de lo habitual, ambos escenarios son tratados como incidentes bajo ITIL 4 [24].

2.2.1.2 Gestión de Incidentes

En el proyecto de Huertas Camacho, Gina Lida mencionan a la gestión de incidentes como un proceso que se basa en impedir que sucedan problemas en el establecimiento, pero en tal caso que se den estos problemas, es necesario analizarlos para encontrar una solución oportuna lo más rápido posible, evitando la interrupción de algunos procesos de la empresa [25].

La gestión de incidentes contribuye mucho en la satisfacción del cliente y del usuario, cada incidente debe registrarse y gestionarse para de esta forma garantizar que se resuelva en un tiempo que cumpla con las expectativas del cliente y del usuario. Las organizaciones deben diseñar las prácticas de gestión de incidentes para proporcionar una gestión y asignación de recursos a los diferentes tipos de incidentes de manera adecuada, los incidentes de bajo nivel o impacto deben gestionarse de manera eficiente para que no consuman demasiados recursos, en cambio los incidentes graves o de mayor impacto pueden requerir de más recursos, por lo que su gestión es más compleja [26].

Los incidentes pueden ser diagnosticados y resueltos por personas de diferentes grupos, según la complejidad del problema o el tipo de incidente. Todos estos grupos deben comprender el proceso de gestión de incidentes y como su contribución a esto ayuda a gestionar el valor, los resultados, los costos y los riesgos de los servicios prestados [26]:

- Las incidencias individuales serán resueltas por los propios usuarios mediante autoservicio. Se debe registrar el uso de registros de autoservicio específicos para las actividades de medición y mejora.
- Algunas incidencias serán resueltas por el servicio de asistencia.
- Los incidentes más complejos generalmente se derivan a los equipos de soporte para resolverlos, el enrutamiento se basa en la categoría del incidente para ayudar a identificar el equipo correcto.
- Los incidentes pueden derivarse a proveedores o socios que brindan soporte para sus productos y servicios.
- Para los incidentes más complejos y todos los incidentes importantes, los equipos ad hoc deben trabajar juntos para determinar soluciones.

En la ilustración 4. Se puede observar cómo contribuye la gestión de incidencias a la cadena de valor del servicio, y la práctica se aplica principalmente a las actividades de involucrar, entregar y apoyar en la cadena de valor con excepción del plan, algunas actividades pueden utilizar información sobre incidentes para ayudar a establecer prioridades como [26]:

Mejorar: Los registros de incidentes son un recurso clave para las actividades de mejora y se priorizan en términos entre términos de mejora y de gravedad.

Comprometerse: Los incidentes importantes son visibles tanto para el usuario como para los clientes. Una buena gestión de incidentes requiere una comunicación regular para comprender los problemas, establecer expectativas, proporcionar constantes actualizaciones de estado y aceptar el problema está resuelto para que el incidente pueda cerrarse.

Diseño y transición: Los incidentes pueden ocurrir en entornos de prueba, también en el lanzamiento y la implementación del servicio. La práctica asegura que estos incidentes se resuelvan de manera oportuna y controlada.

Obtener / construir: Los incidentes también pueden ocurrir en entornos de desarrollo. La práctica de la gestión de incidentes va a garantizar que estos incidentes se resuelvan de manera oportuna y controlada.

Entregar y apoyar: La gestión de incidentes hace una construcción significativa al soporte.

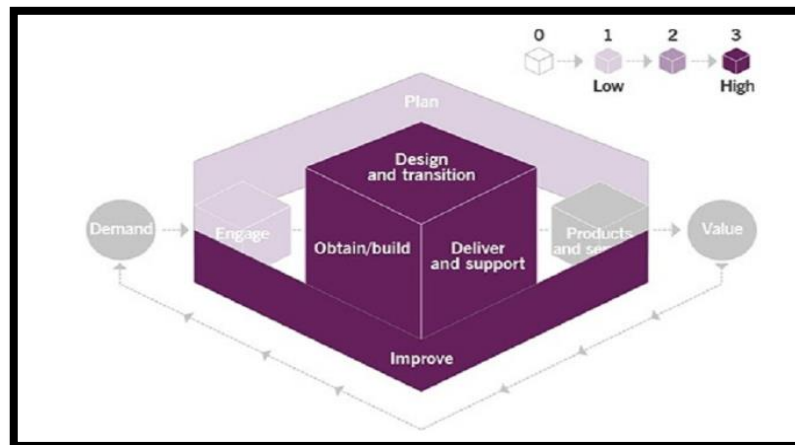


Ilustración 4 Mapa de calor de la contribución del control de cambios a las actividades de la cadena de valor [26]

2.2.1.2.1 Primer Nivel de Soporte TICS

El primer nivel de soporte de TICS hace referencia a la asistencia en primera línea donde el técnico tendrá que reunir toda la información del problema para determinar la posible causa, por lo general este tipo de soporte suele tratarse como problemas de fácil solución y que están relacionados con [27]:

- ✓ Resolución de Incidencias a usuarios y contraseñas
- ✓ Instalación de software
- ✓ Solución o consulta de incidencias.
- ✓ Problemas de comunicación y redes.

2.2.2. Tecnologías emergentes

Este término, se entiende como tecnologías nuevas que se están incorporando al mercado de los servicios. Estas innovaciones incluyen desde la inteligencia artificial y el aprendizaje automático hasta la computación en la nube y el Internet de las Cosas (IoT) [28]. Estas tecnologías están transformando la manera en que las empresas

operan, permitiéndoles ofrecer servicios más eficientes, personalizados y accesibles para los usuarios.

2.2.2.1 Chat bot

En su trabajo de titulación Jorge Fernández se refiere a un asistente como un programa que se comunica con los usuarios a través de mensajes de texto. El sistema está programado para que interactúe con el cliente y le resuelva dudas, pero sin que haya una persona física contestando. Tienen la ventaja de que están disponibles siempre para resolver las dudas de los usuarios que quieran contactar con las empresas o servicios a cualquier hora del día. [29]

2.2.2.2 DialogFlow

Es una destacada plataforma de desarrollo de chatbots y asistentes virtuales creada por Google, esta plataforma brinda la capacidad de diseñar interfaces de conversación natural para aplicaciones y servicios de manera eficiente, facilitando así la creación de experiencias de usuario conversacionales de alta calidad [8].

2.2.2.3 GitHub

Es una plataforma de desarrollo colaborativo que se basa en el sistema de control de versiones Git, esta plataforma es ampliamente reconocida por su capacidad para alojar proyectos de software, facilitar la gestión de control de versiones y promover la colaboración en proyectos de código abierto, además de su función principal como un repositorio de código [11].

2.2.2.4 Zammad

Zammad es un sistema de servicio de asistencia de código libre basado en la web que incluye muchas funciones interesantes para administrar la comunicación con el cliente. Ofrece integración con múltiples canales como teléfono, Facebook, Twitter y correos electrónicos, facilitando que las empresas mantengan todo en un solo lugar. Además, permite el seguimiento de tickets, la gestión de bases de conocimiento y la automatización de tareas repetitivas, lo que mejora la eficiencia del equipo de soporte y asegura que los clientes reciban respuestas rápidas y efectivas [12].

2.2.2.5 Telegram

Telegram es una plataforma de mensajería popular y gratuita que se destaca por su velocidad y seguridad. Es muy apreciada por su capacidad para crear bots de manera instantánea y por ofrecer una API accesible para programarlos [16]. Además, permite realizar llamadas de voz y videollamadas, lo que la convierte en una opción completa para la comunicación tanto personal como profesional. Ya sea para chatear, hacer llamadas o integrar bots en tus servicios, Telegram ofrece una solución versátil y confiable.

2.2.2.6 Grafana

Es una herramienta Open Source que permite visualizar datos de serie temporales, permite crear cuadros de mandos a través de múltiples fuentes, es muy popular y conocida por la capacidad de crear gráficos y dashboards mediante la transformación de datos en información en tiempo real. Además, ofrece integraciones con una amplia gama de bases de datos y servicios en la nube. Su comunidad activa y en crecimiento asegura una constante mejora y actualización de sus funcionalidades además su facilidad de uso la convierten en una elección preferida para analistas y desarrolladores [18].

2.2.2.7 Elastic Search.

Es un motor de búsqueda y analítica que facilita la recopilación, agregación y enriquecimiento de datos. Esta herramienta proporciona búsqueda y análisis casi en tiempo real para todo tipo de datos, permitiendo a las organizaciones obtener información valiosa de manera rápida y eficiente. Además, ofrece capacidades avanzadas de visualización y generación de informes para una mejor comprensión de los datos [19].

2.2.3 Backend

El backend se encarga de la manipulación de los datos, es decir es la capa de acceso de los datos al software que no es accesible por el usuario final, en esta capa del desarrollo se encuentra toda la lógica de la aplicación que maneja los datos que se encuentran alojados en un servicio de base de datos. Esta parte se centra en la gestión eficiente y segura de la información [30].

2.2.3.1 Node js

Es una plataforma de desarrollo de código abierto en JavaScript que sobresale por su capacidad para instalar módulos y dependencias en proyectos, además es ampliamente conocida por su versatilidad en la creación de aplicaciones de red altamente escalables [13].

2.2.3.2 Express js

Es una infraestructura o framework de backend destinada para aplicaciones web es la más popular de la librería Node.js destaca por su flexibilidad, express proporciona al desarrollador características de aplicación web y móviles básicas. Su eficiencia y escalabilidad lo convierten en una opción ideal para construir aplicaciones de alta performance, mientras que su amplia comunidad asegura soporte continuo y abundantes recursos. [14]

2.2.3.3 JavaScript

Es un lenguaje de programación ampliamente empleado en el desarrollo web. Se destaca por su versatilidad y es especialmente reconocido por su capacidad para crear páginas web interactivas y dinámicas que mejoran significativamente la experiencia del usuario. Además, su comunidad activa proporciona un soporte constante y recursos abundantes para desarrolladores de todos los niveles. [15].

2.2.3.4 API.

Es una interfaz informática que define la interacción entre varios intermediarios, el software determina los tipos de llamadas o solicitudes que se pueden realizar, cómo se realizan, formato de datos a utilizar, convenciones a seguir, etc. También son los encargados de proporcionar un mecanismo de extensión para que los usuarios puedan ampliar la funcionalidad existente [31].

2.2.3.5 Webhooks

Los webhooks son los encargados de establecer de enviar solicitudes http post a una url especificada por un usuario, cuando un evento sucede. La conexión es de larga duración y son ampliamente utilizados para enviar eventos en tiempo real, estos suelen activarse para ejecutar alguna parte de código, enviar mensajes, actualizaciones, etc. [31].

2.2.3.6 Base de datos

Se define el termino base de datos una recopilación organizada de información o datos estructurados, que por lo general se almacenan de forma electrónica en un sistema informático, normalmente una base de datos es controlada por un sistema de gestión de base de datos [32].

2.2.3.6.1 PostgreSQL

Es un sistema de gestión de bases de datos relacional de código abierto, con 35 años de desarrollo activo, es conocido por su robustez, capacidad de extensión y cumplimiento de los estándares SQL [9].

2.2.4 Web Service

Un Web Service son aplicaciones de sistema que exponen métodos para que el usuario pueda consultar, insertar, actualizar o eliminar información, la información de consulta y respuesta se las realiza por medio de protocolos de comunicación, sirven para proporcionar un componente de comunicación estándar estas interactúan entre sí para presentar información a quien la solicita [33].

2.2.4.1 Render

Es una plataforma de alojamiento en la nube que ofrece servicios de alojamiento web y aplicaciones. Permite a los desarrolladores implementar y administrar aplicaciones web y sitios de manera escalable, proporcionando recursos computacionales flexibles y una infraestructura robusta para satisfacer las necesidades de crecimiento de cualquier proyecto en línea [10].

2.3 Marco Teórico

2.3.1 Innovaciones de ITIL 4: Revolucionando la Gestión de Incidentes

ITIL v4 está revolucionando la forma en la que las organizaciones se enfrentan a desafíos de la administración de servicios. En su trabajo de investigación Luis Bravo y Miguel Andrade titulado “ITIL v4 en la gestión de solicitudes e incidentes de la mesa de ayuda de la Universidad Nacional de Loja”, propuso como objetivo principal la forma en la que se está dando uso al marco de trabajo ITIL v3 para que se pueda reestructurar y ajustar mediante buenas prácticas y tácticas la adaptación a la nueva

versión de ITIL denominado v4, para ello realizaron pruebas desde el 17 de marzo hasta el 30 de noviembre del año 2020, aplicando las recomendaciones que dicta la versión 4, dando como resultado, el alza del grado de resolución de incidentes, llegando al pico de 98,80% en el mes de Mayo, donde se presentaron 1993 solicitudes de los cuales 1969 fueron resueltas con éxito [34].

2.3.2 El poder del software libre.

En su trabajo de titulación José Castro, hace uso de herramientas libres como Zammad y Grafana, para la implementación de un software de monitoreo de eventos y fallas dando como justificación que el uso de estas herramientas para la gestión adecuada y presentación de datos en tiempo real se debe a que existen un sin número de aplicaciones que ofrecen el mismo servicio pero que ponen barreras a las empresas al momento de implementar, una de estas barreras son los altos costos que demandan su uso, la finalidad del proyecto fue demostrar la capacidad de utilizar sistemas de código libre e integrarlas para brindar una solución global a la gestión de incidencias, subrayando que el uso de estas herramientas no solo reducen costos, sino que también permiten mayor flexibilidad transparencia y control sobre procesos [35].

2.2.3 Respuestas Inmediatas: Transformando la Atención al Cliente con Chatbots

Héctor Jover Ibáñez, en su proyecto donde realizó un chatbot para ayudar a las personas a encontrar puntos de interés en la ciudad de Valencia/España, atribuye el uso de los chatbots como una tecnología que en la actualidad se encuentra en alza, por la mejora constante en el área de inteligencia artificial y el uso masivo de aplicaciones móviles y web que facilitan su integración. Implementar un chatbot ayuda significativamente a que el usuario final se encuentre siempre atendido y a la espera de respuestas, es por ello por lo que esta tecnología es ampliamente utilizada y que cada vez sea más común observar a clientes interactuando con chatbots. Además, permiten una atención al cliente más eficiente y personalizada, reduciendo los tiempos de espera y mejorando la experiencia del usuario. Su flexibilidad y capacidad de aprendizaje continuo los convierten en una herramienta clave para diversas industrias. [36].

2.4 Requerimientos

2.4.1 Requerimientos Funcionales

MÓDULO DE SEGURIDAD	
Código	Descripción
RF-1	Ares contará con un login antes de iniciar cualquier proceso. Para ello el usuario deberá proporcionar su número de cédula.
RF-2	Los perfiles de usuario que cuenta el chatbot están restringidos y la información que se muestra depende del rol que ocupa el usuario en la empresa.
RF-3	Ares no permitirá ejercer algún cambio de información de datos personales.

Tabla 2 Requerimientos Funcionales - Módulo de Seguridad.

MÓDULO REGISTRO DE INCIDENTES	
Código	Descripción
RF-4	Ares deberá permitir a los usuarios con rol trabajador, registrar nuevos incidentes de TICS.
RF-5	Ares deberá permitir a los usuarios con rol trabajador, describir el incidente.
RF-6	Ares deberá proporcionar un campo para que el usuario con rol trabajador escriba el título del incidente.

Tabla 3 Requerimientos Funcionales - Módulo Registro de Incidentes.

MÓDULO DE SEGUIMIENTO DE INCIDENTES	
Código	Descripción
RF-7	Ares deberá permitir a los usuarios con rol trabajador, poder consultar los incidentes que ha reportado.
RF-8	Ares deberá permitir a los usuarios con rol administrador redireccionar hacia Zammad para dar seguimiento seguir los incidentes que han sido asignados a los agentes.
RF-9	Ares deberá notificar por medio de Notify bot, los cambios de estado de los tickets, hasta su debido cierre.

Tabla 4 Requerimientos Funcionales - Módulo Seguimiento de Incidentes.

MÓDULO RESOLUCIÓN DE INCIDENTES	
Código	Descripción
RF-10	Ares deberá consultar una solución mediante el uso de la API de chat Gpt
RF-11	Ares deberá implementar la automatización en la resolución del incidente de primer nivel (soporte técnico) siguiendo los estándares de ITILv4.
RF-12	Ares deberá proporcionar cinco títulos de resolución del incidente dependiendo la descripción del incidente.

<i>Código</i>	<i>Descripción</i>
<i>RF-13</i>	Ares deberá preguntar que título de resolución desea leer, en caso de que ese título no le haya ayudado, se podrá elegir otro.

Tabla 5 Requerimientos Funcionales - Módulo de Seguimiento de Incidentes.

MÓDULO DE GESTIÓN DEL CONOCIMIENTO	
<i>Código</i>	<i>Descripción</i>
<i>RF-14</i>	Ares deberá obtener la información de resolución exitosa.
<i>RF-15</i>	Ares deberá guardar la información de resolución exitosa en la base de conocimientos.

Tabla 6 Requerimientos Funcionales - Módulo de Gestión del Conocimiento.

MÓDULO DE CIERRE DEL INCIDENTE	
<i>Código</i>	<i>Descripción</i>
<i>RF-16</i>	Ares deberá confirmar que el incidente ha sido resuelto.
<i>RF-17</i>	Ares deberá enviar una notificación de que el incidente se ha guardado como resuelto.
<i>RF-18</i>	Ares deberá conectarse con Zammad para generar un ticket en caso de que no pueda resolver el incidente.

<i>Código</i>	<i>Descripción</i>
<i>RF-19</i>	Ares deberá cerrar la sesión una vez que se haya generado el ticket o el incidente haya sido resuelto.
<i>RF-20</i>	Ares guardará los siguientes datos para tener mejor control de los tickets: <ul style="list-style-type: none"> ✓ Nombre del reportador ✓ Fecha y hora del registro ✓ Título del incidente ✓ Descripción del incidente
<i>RF-21</i>	Ares no permitirá ejercer alguna acción una vez que haya cerrado sesión.

Tabla 7 Requerimientos Funcionales – Módulo Cierre del Incidente.

MÓDULO SOLICITAR REPORTES	
<i>Código</i>	<i>Descripción</i>
<i>RF-22</i>	Ares deberá permitir al usuario con rol administrador solicitar reportes diarios y mensuales.
<i>RF-23</i>	Ares deberá enviar los reportes por pdf al usuario con rol administrador de
<i>RF-24</i>	Ares deberá proporcionar un enlace al usuario con rol administrador donde se muestre el reporte detallado de incidentes ofrecido por Grafana.

Tabla 8 Requerimientos Funcionales - Módulo de Solicitar Reportes.

MÓDULO SOLICITAR CSV	
Código	Descripción
RF-25	Ares deberá permitir al usuario al usuario con rol administrador de TICS solicitar archivo csv de todos los incidentes reportados.

Tabla 9 Requerimientos Funcionales - Módulo Solicitar CSV.

2.4.2 Requerimientos No Funcionales

Código	Descripción
RNF-1	Ares estará disponible para responder a las solicitudes solo en el horario de 8am hasta 17:00 pm.
RNF-2	Ares responderá en aproximadamente 1 segundo al momento de su invocación. garantizando así el rendimiento del sistema.
RNF-3	Los datos de incidentes se guardarán en la tabla de tickets del software Zammad.
RNF-4	Por medio de Zammad se crean los usuarios y automáticamente tendrán acceso a Ares.
RNF-5	Por medio de Zammad se podrá bloquear y eliminar usuarios y automáticamente perderán el acceso a Ares.

<i>Código</i>	<i>Descripción</i>
<i>RNF-6</i>	Los datos de los tickets se actualizan cada que el agente de Zammad realice algún cambio.
<i>RNF-7</i>	Ares enviará las respuestas del incidente reportado en menos de 3 segundos.
<i>RNF-8</i>	Los datos están protegidos por mediante las capas de seguridad que ofrece Zammad.
<i>RNF-9</i>	Las contraseñas de los usuarios estarán encriptadas en la base de datos.
<i>RNF-10</i>	Ares cuando no es usado por un lapso aproximado de 5 minutos se pondrá en reposo debido al Web Service gratuito que se está usando, por esa razón cuando un usuario lo invoque deberá esperar entre 30 segundos a 1 minuto para que vuelva a estar en línea.

Tabla 10 Requerimientos No Funcionales

2.5 Componentes de la propuesta

2.5.1 Arquitectura del Sistema

En la ilustración 5 se puede observar la arquitectura del proyecto, donde el usuario al momento de interactuar con el chatbot genera una petición la cual va a ser procesada por medio de DialogFlow una plataforma de Google para desarrollar interfaces conversacionales, esto con el objetivo de verificar si lo que escribió el usuario corresponde a una intención previamente registrada, para que el chatbot sea más

intuitivo, se usará un webhook el cual está desarrollado con Node.js, Express, con la ayuda de esto se va a lograr que la respuestas y acciones sean tomadas directamente del webhook, permitiendo de esta manera tener la posibilidad de colocar botones para el usuario, gracias a la ayuda de la librería node-telegram-bot.

El webhook se conecta con la API de Chat gpt cuando se requiera una solución a un incidente que el usuario ha tenido. Cuando el chatbot requiere crear un ticket inmediatamente, el webhook lanza una solicitud HTTP POST con los datos necesarios hacia el servicio de gestión de tickets Zammad, logrando de esta manera que el ticket sea creado con éxito. A su vez, utiliza servicios como Elasticsearch, esto indica que todos los datos que se guarden en la base de datos PostgreSQL son compartidos por dicho servicio para indexarlos. Estos datos van a ser utilizados por el servicio de Grafana, el cual ayudará a crear reportes detallados y estéticos de los tickets.

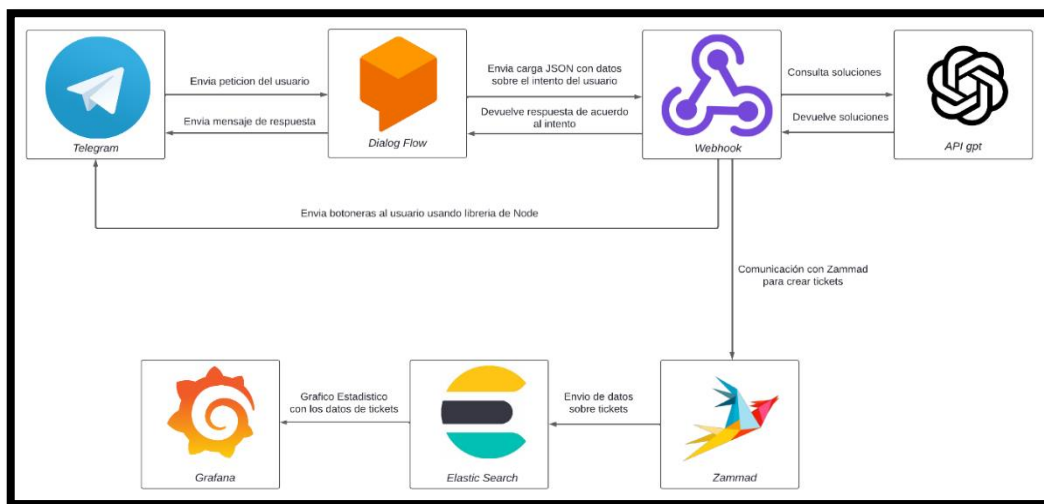


Ilustración 5 Componentes del backend.

2.5.2 Proceso de incidentes

En la ilustración 6 se observa el proceso que realiza Ares para registrar un incidente. El usuario, para poder usar el chatbot, necesita ingresar los datos solicitados. Una vez que el usuario se ha logueado e indicado cuál es su incidente, el chatbot lo registrará y consultará soluciones a ese problema utilizando la API de ChatGPT. Cuando se obtenga una respuesta, se enviarán las sugerencias de posibles soluciones. El usuario validará si esas soluciones le sirven para resolver su incidente o no.

Cuando el usuario indica que el incidente pudo ser resuelto por Ares, el chatbot se desconectará automáticamente y el incidente se registrará como solucionado en la base de datos. En caso de que Ares no logre solucionarlo, se inicia el proceso de creación de ticket, donde Ares le pedirá al usuario un título para su incidente y procederá a crearlo con su respectivo número. Este ticket llegará a Zammad y se guardará en la base de datos con estado pendiente.

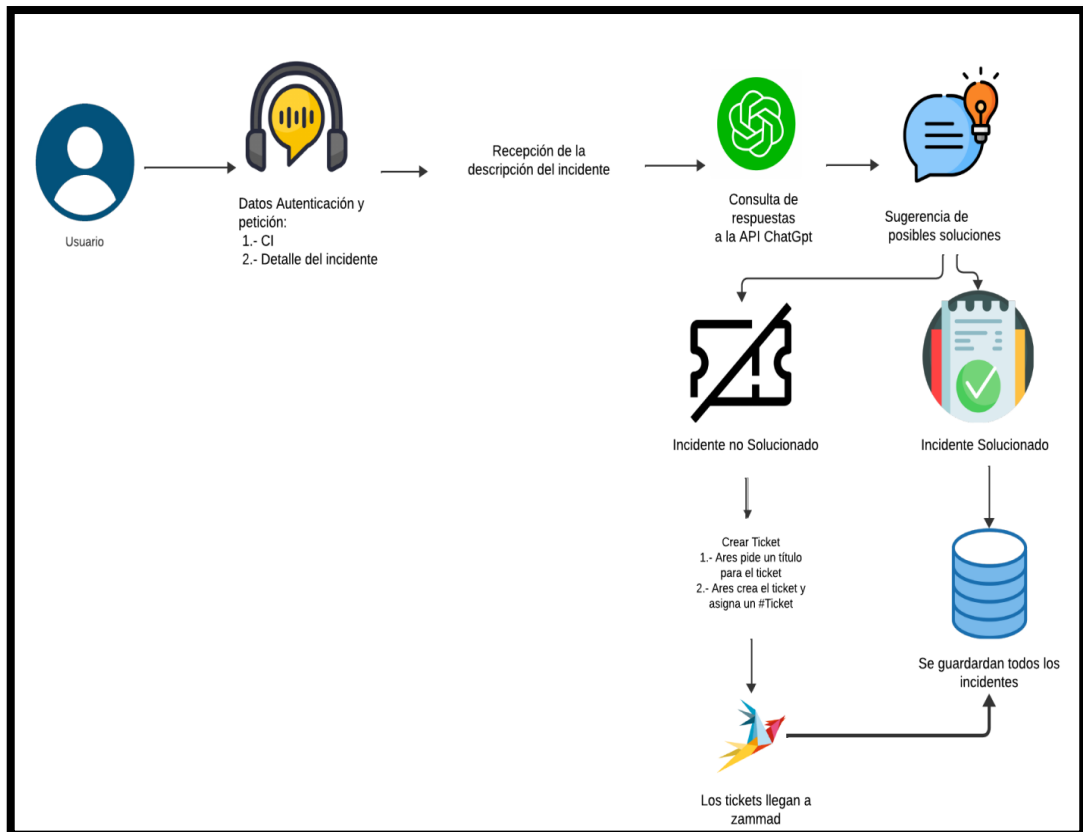


Ilustración 6 Diagrama del proceso de incidentes

2.5.2.1 Limpieza de datos.

Para realizar pruebas de funcionamiento y generar gráficos estadísticos de tickets basados en un entorno real, se utilizó un dataset de incidentes que contiene 12,000 registros de logs, correspondientes a incidentes reales ocurridos en una empresa. Este dataset fue elegido por su similitud con la estructura de datos de Zammad. Sin embargo, fue necesario un proceso de limpieza de datos para ajustar completamente la estructura. Para la limpieza de datos se utilizó el lenguaje de programación Python y la biblioteca Pandas. El proceso consta de dos fases:

Fase 1 Cambio de fechas.

El cambio de fechas en los logs fue crucial para asegurar que los datos fueran relevantes y útiles para la generación de gráficos estadísticos actualizados. Este proceso de actualización mediante un script garantizó que las visualizaciones reflejen con precisión la situación actual, facilitando así la toma de decisiones informadas basadas en datos recientes. La ilustración 7 ofrece una visión clara del código utilizado para este propósito, destacando la eficiencia y la precisión del proceso de transformación de datos.

```
import pandas as pd
import numpy as np

df = pd.read_csv('sinRepetidos.csv').sample(n=300, random_state=42)
date_columns = ['opened_at', 'sys_created_at', 'sys_updated_at', 'closed_at']
for col in date_columns:
    df[col] = pd.to_datetime(df[col], errors='coerce')
def random_date(start, end):
    return start + (end - start) * np.random.rand()

# Rango de fechas: del 1 de junio de 2024 al 31 de julio de 2024
start_date = pd.Timestamp('2024-06-01')
end_date = pd.Timestamp('2024-07-31')
for col in date_columns:
    df[col] = [random_date(start_date, end_date) for _ in range(len(df))]
    df[col] = df[col].apply(lambda x: x if x.weekday() < 5 else random_date(start_date, end_date))
df.to_csv('zammadTickets.csv', index=False)
```

Ilustración 7 Limpieza de datos - Script de cambio de fecha.

Fase 2 Estructurar los datos.

Los datos del dataset presentaban cierta similitud con la estructura de Zammad, pero no encajaban completamente; aún faltaban campos necesarios. Para alinear los datos con la estructura deseada, se desarrolló un script que unificó las columnas del dataset, eliminó aquellas innecesarias del original y agregó campos adicionales para asegurar el éxito del proceso, dejando únicamente aquellas compatibles con Zammad. Este proceso de ajuste y normalización garantizó que los datos estuvieran listos para ser integrados de manera efectiva en el sistema Zammad, asegurando la coherencia y la precisión en la gestión de información. La ilustración 8 muestra el código utilizado para esta tarea.

```

import pandas as pd
import random
df_original = pd.read_csv('zammadTickets.csv')
filas = []
mapeo_columnas = {
    'number': 'number',
    'incident_state': 'state_id',
    'opened_by': 'created_by_id',
    'sys_created_at': 'created_at',
    'sys_created_by': 'created_by_id',
    'sys_updated_by': 'updated_by_id',
    'sys_updated_at': 'updated_at',
    'closed_at': 'close_at',
    'assigned_to': 'owner_id'
}

for indice, fila in df_original.iterrows():
    nueva_fila = {}
    for columna_orig, columna_nueva in mapeo_columnas.items():
        nueva_fila[columna_nueva] = fila[columna_orig]

    nueva_fila['title'] = 'Título de incidente de TIC de primer nivel'
    nueva_fila['state_id'] = random.randint(1, 4)
    nueva_fila['created_by_id'] = random.choice([2, 5, 6, 7, 8, 9, 10])
    nueva_fila['updated_by_id'] = nueva_fila['created_by_id']
    nueva_fila['owner_id'] = random.choice([3, 11, 12, 13])
    nueva_fila['customer_id'] = nueva_fila['created_by_id']

    filas.append(nueva_fila)

nuevo_df = pd.DataFrame(filas)

# Guardar el nuevo DataFrame en un archivo CSV
nuevo_df.to_csv('tickets.csv', index=False)

```

Ilustración 8 Limpieza de datos - Ajuste de campos.

Tras finalizar el proceso mencionado anteriormente, se obtuvo un total de 300 tickets que fueron seleccionados aleatoriamente para su uso en un entorno de prueba en Zammad. Dado el tamaño considerable del conjunto de datos, cargar todos los tickets en el sistema sería poco práctico. Esta muestra representativa permitirá realizar pruebas sin sobrecargar el sistema de manera innecesaria.

2.5.3 Comparativas de herramientas.

Durante gran parte del desarrollo del proyecto, se optó por utilizar herramientas de código abierto. Aunque en el mercado existen varias aplicaciones que ofrecen servicios similares, el alto costo asociado a algunas de ellas puede ser una barrera para su implementación. Por esta razón, se prefirieron herramientas de código abierto que, en conjunto, cumplen con los requisitos necesarios para el desarrollo del chatbot. Este enfoque no solo permite un desarrollo más accesible económicamente, sino que también fomenta la transparencia, la colaboración y la personalización según las necesidades del proyecto.

2.5.3.1 App de mensajería

<i>Característica</i>	<i>Telegram</i>	<i>WhatsApp</i>
<i>Facilidad de uso de la API</i>	Proporciona una API documentada, fácil de usar y con muchos ejemplos disponibles, lo que facilita el desarrollo de bots y aplicaciones.	API de negocio limitada, menos intuitiva.
<i>Documentación</i>	Extensa y clara, con muchos recursos para desarrolladores	No tan completa, puede requerir investigación adicional.
<i>Configuración Inicial</i>	Simple y rápida	Compleja y requiere configuración con terceros (e.g., Twilio).
<i>Librerías y SDKs</i>	Disponibilidad de múltiples librerías y SDKs en varios lenguajes	Poca disponibilidad de librerías y SDKs oficiales.
<i>Costos</i>	Gratuito	Puede incurrir en costos con servicios como Twilio
<i>Escalabilidad</i>	Fácil de escalar sin costos adicionales	Escalar puede aumentar los costos debido a proveedores externos.
<i>Integración de Funciones Avanzadas</i>	Soporta comandos, teclados personalizados, menús, etc.	Ofrece opciones básicas de integración y tiene menos funciones avanzadas en comparación con Telegram
<i>Despliegue de Bots</i>	Sencillo, sin necesidad de aprobación adicional	Requiere aprobación y configuración de terceros.

Tabla 11 Comparativa entre Telegram y WhatsApp.

2.5.3.2 Software de Gestión de Tickets.

<i>Característica</i>	<i>Zammad</i>	<i>Freshdesk</i>
<i>Facilidad de uso de la API</i>	API documentado, RESTful, fácil de integrar	API documentado, RESTful, con múltiples endpoints.
<i>Librerías y SDKs</i>	Soporte en múltiples lenguajes (Python, Ruby, etc.)	Amplia disponibilidad de SDKs y bibliotecas en varios lenguajes.
<i>Costos</i>	Gratuito (Open Source)	Modelos de precios basados en suscripción (varía según el plan).
<i>Integración de Funciones Avanzadas</i>	Soporta automatizaciones, integraciones con múltiples servicios	Soporta automatizaciones avanzadas y muchas integraciones.
<i>Despliegue</i>	Puede ser auto alojado o en la nube, más control sobre los datos	Basado en la nube, fácil de desplegar.

Tabla 12 Comparativa entre Zammad y Freshdesk.

2.5.3.3 Software de gráficos estadísticos.

<i>Característica</i>	<i>Grafana</i>	<i>Datadog</i>
<i>Facilidad de uso de la API</i>	API documentado, fácil de usar y personalizar	API documentado, rica en características, fácil integración
<i>Configuración Inicial</i>	Puede requerir configuración manual, pero es flexible	Sencilla con asistentes guiados para una rápida configuración

<i>Características</i>	<i>Grafana</i>	<i>Datadog</i>
<i>Costos</i>	Gratuito (Open Source), posible costo de infraestructura	Basado en suscripción, costo varía según el plan y uso
<i>Integración de Funciones Avanzadas</i>	Amplia gama de integraciones, plugins y paneles personalizados	Funciones avanzadas, integraciones nativas y alertas automatizadas
<i>Despliegue</i>	Autoalojado o en la nube, más control sobre la infraestructura	Basado en la nube, fácil de desplegar y gestionar

Tabla 13 Comparativa entre Grafana y Datadog.

2.5.4 Diagramas de Casos de Uso

2.5.4.1 Diagrama de acceso al chatbot

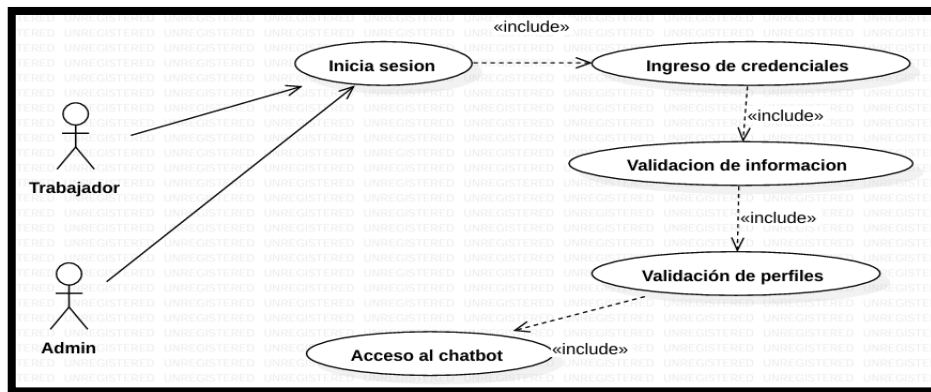


Ilustración 9 Diagrama de caso de uso- acceso al sistema.

<i>Caso de uso: Acceso al sistema</i>	
<i>Actores</i>	Trabajador, Administrador
<i>Tipo</i>	Flujo básico

<i>Caso de uso: Acceso al sistema</i>	
<i>Resumen</i>	Permite acceder, a través de la validación de roles, el trabajador o administrador ingresa sus credenciales, se realiza la validación y permite el acceso al chatbot según el rol que cumple en la empresa.
<i>Flujo Principal</i>	<ul style="list-style-type: none"> • El usuario invocará al chatbot. • El chatbot pedirá las credenciales. • El chatbot le dará acceso. • El chatbot mostrará opciones de acuerdo con el rol del usuario.
<i>Subflujos</i>	El chatbot deberá realizar la validación de las credenciales del usuario, consultando a la base de datos la existencia de estos datos, luego verifica el rol y retorna las opciones a las que puede acceder.
<i>Excepciones</i>	El chatbot no permitirá el ingreso de usuarios que no estén registrados en la base de datos, garantizando de esta manera que solo usuarios autorizados puedan interactuar con el asistente virtual.

Tabla 14 Caso de uso - Acceso al sistema.

2.5.4.2 Diagrama rol trabajador

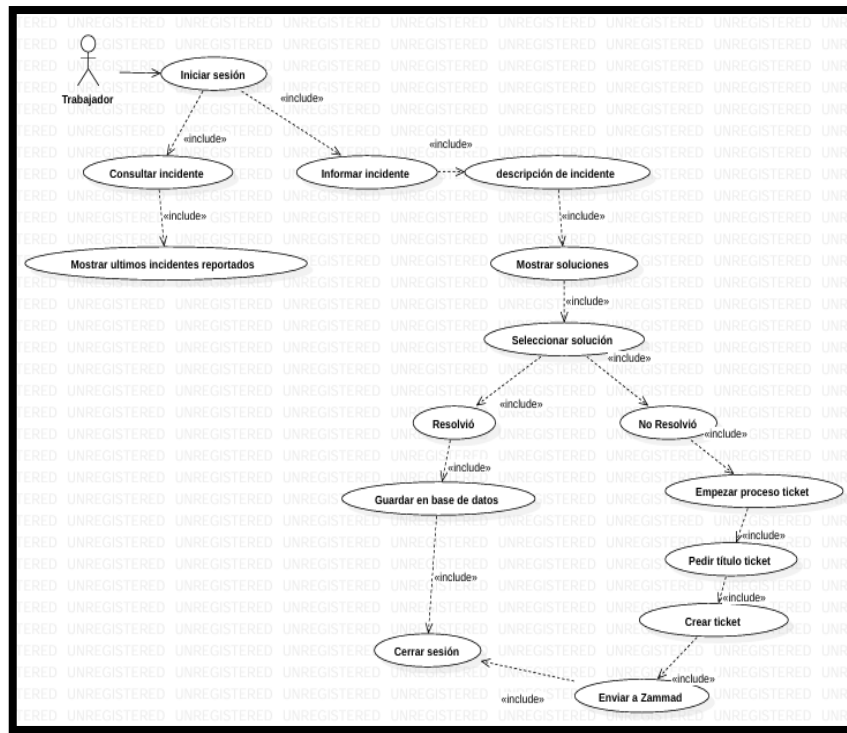


Ilustración 10 Diagrama de caso de uso - Proceso rol trabajador.

Caso de uso: Proceso rol trabajador	
Actores	Trabajador
Tipo	Flujo básico
Resumen	Proceso de todo lo que puede realizar el actor trabajador, el chatbot pedirá iniciar sesión y mostrará dos opciones, informar incidente y consultar incidente, en la primera opción el chatbot pedirá que le detalle el incidente y mostrará opciones de posibles soluciones para que el usuario elija, cual desea ver y posterior a eso preguntará al

<i>Caso de uso: Proceso rol trabajador</i>	
	<p>usuario si resolvió o no el incidente, si la respuesta es afirmativa el chatbot registrará el incidente y cerrará sesión, caso contrario creará un ticket hacia Zammad y cerrará sesión.</p> <p>En la segunda opción el chatbot mostrará detalles de los últimos incidentes que ha reportado el trabajador.</p>
<i>Flujo Principal</i>	<ul style="list-style-type: none"> • El usuario invocará al chatbot. • El chatbot pedirá las credenciales. • El chatbot mostrará opciones. • El chatbot pedirá detalles del incidente. • El chatbot pedirá que seleccione entre las opciones de resolución. • El chatbot preguntará si se resolvió el problema. • El chat bot cerrará la sesión y guardará los datos si la respuesta es afirmativa. • El chatbot creará un ticket y pedirá un título si la respuesta es negativa • El chatbot enviará los datos a zammad y cerrará sesión. • El chatbot mostrará últimos incidentes reportados, si la

Caso de uso: Proceso rol trabajador	
	opción elegida es la de consultar incidentes.
Subflujos	El chatbot deberá en constante comunicación con la base de datos, realizando consultas de incidentes y registros de estos.
Excepciones	<ul style="list-style-type: none"> • Error al crear el ticket, el chatbot notificará si ocurre algún error. • El chatbot notificará si hay problemas de conexión al realizar una consulta.

Tabla 15 Caso de uso – Proceso rol trabajador.

2.5.4.3 Diagrama rol Administrador

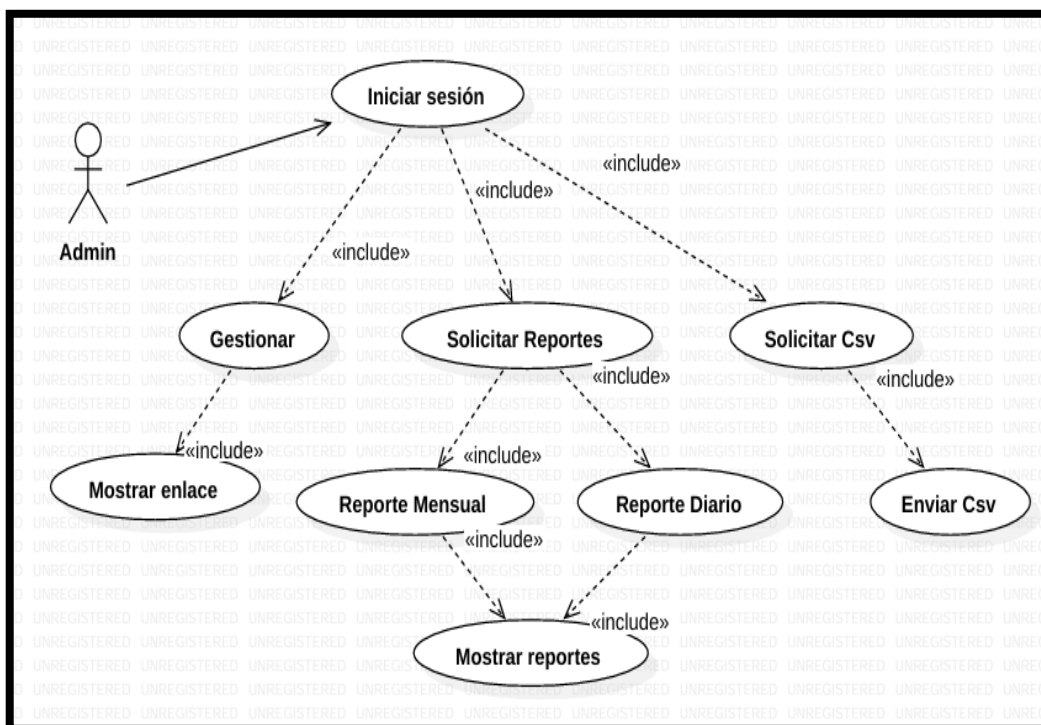


Ilustración 11 Diagrama de caso de uso - Proceso rol administrador.

<i>Caso de uso: Rol Administrador</i>	
<i>Actores</i>	Administrador
<i>Tipo</i>	Flujo básico
<i>Resumen</i>	Permite ejecutar las diversas funciones disponibles para el rol administrador tales como gestionar, solicitar reportes, solicitar archivo CSV y salir del chatbot.
<i>Flujo Principal</i>	<ul style="list-style-type: none"> • El administrador invoca al chatbot • El chatbot mostrará las tres opciones disponibles. • En la opción de gestionar el chatbot enviará un enlace donde el administrador podrá ingresar a Zammad. • En la opción de solicitar reportes el chatbot mostrará un submenú donde el administrador deberá elegir entre reporte diario y mensual y esperar su reporte. • En la opción de solicitar csv, el chatbot enviará un archivo csv de todos los tickets registrados en la base de datos
<i>Subflujos</i>	Gestión del Chatbot a través de Zammad:

Caso de uso: Rol Administrador

1. El administrador selecciona la opción de gestionar.
2. El chatbot envía un enlace a Zammad.
3. El administrador ingresa a Zammad a través del enlace.
4. El administrador realiza las acciones de gestión necesarias en Zammad.

Solicitud de Reportes:

1. El administrador selecciona la opción de solicitar reportes.
2. El chatbot muestra un submenú con opciones para reporte diario y reporte mensual.
3. El administrador elige entre reporte diario o mensual.
4. El chatbot genera el reporte solicitado.
5. El chatbot envía el reporte al administrador.

Solicitud de Archivo CSV:

1. El administrador selecciona la opción de solicitar CSV.
2. El chatbot genera un archivo CSV con todos los tickets registrados en la base de datos.
3. El chatbot envía el archivo CSV al administrador.

Caso de uso: Rol Administrador	
Excepciones	<ul style="list-style-type: none"> • Error en la conexión con Zammad, el chatbot deberá notificar • Error en la generación de reportes, el chatbot deberá notificar. • Error en la generación de csv, el chatbot deberá notificar. • Error en la conexión con la base de datos, el chatbot deberá notificar.

Tabla 16 Caso de uso – Proceso rol administrador.

2.5.4.4 Diagrama general de uso del chatbot

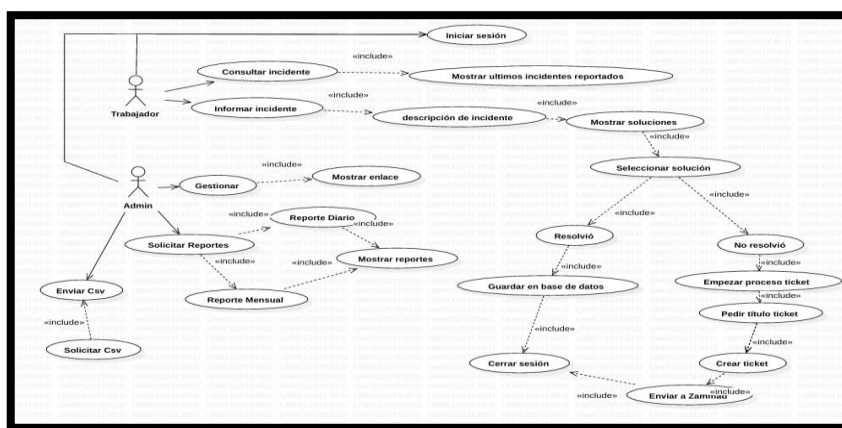


Ilustración 12 Diagrama de caso de uso - General de uso del chatbot

Caso de uso: General de uso del chatbot	
Actores	Administrador, Trabajador

Tipo	Flujo básico
Resumen	Permite ejecutar todas las funciones del chatbot, tanto para el rol administrador, como para el rol trabajador.
Flujo Principal	<ul style="list-style-type: none"> • El administrador o trabajador invoca al chatbot • El chatbot mostrará opciones de trabajador • El chatbot mostrará el proceso desde consulta, soluciones y cierre el incidente • El chatbot mostrará opciones de administrador. • El chatbot mostrará acciones y procesos a seguir para el administrador.
Subflujos	El chatbot ejecutará consultas SQL, a la base de datos, y comunicación con Zammad por medio de http post y get, con el objetivo de realizar las acciones de consulta, registro de incidentes y creación de tickets.
Excepciones	<ul style="list-style-type: none"> • Error en la conexión con Zammad, el chatbot deberá notificar • Error en la conexión con la base de datos, el chatbot deberá notificar.

que ha accedido al chatbot. A continuación, se nombran los componentes que conforman esta interfaz:

- Título que indica el nombre del bot.
- Área de texto para escribir mensajes, junto con el botón de enviar.
- Botones proporcionados por Telegram para opciones adicionales.
- Contenedor de mensajes que muestra la conversación entre el chatbot y el usuario.

Interfaz Inicio de Sesión

En la ilustración 14 se observa cómo el usuario invoca al chatbot Ares, el cual solicita al usuario que ingrese su número de identificación. Posteriormente, el chatbot verifica la identidad del usuario y le proporciona acceso, mostrando la información correspondiente según su rol.

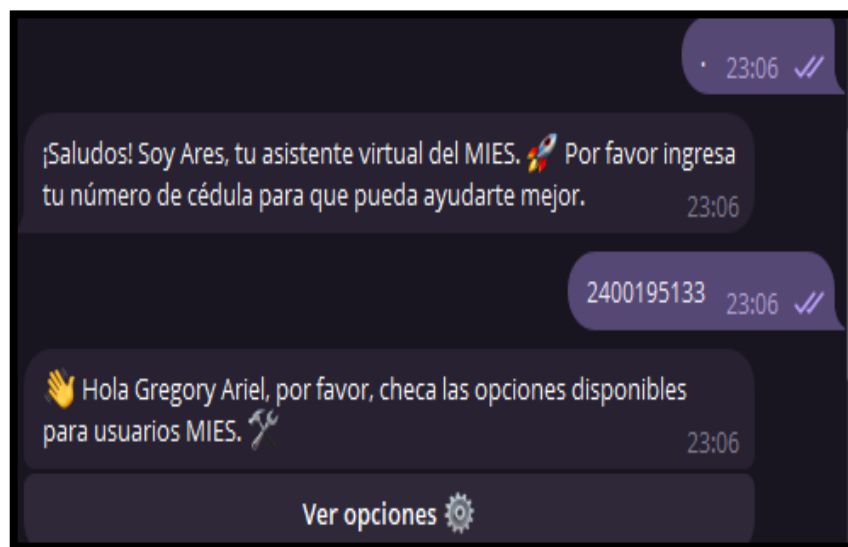


Ilustración 14 Interfaz - Inicio de sesión

Interfaz de trabajador

En la interfaz del trabajador, representada en la ilustración 15, se presentan tres opciones claves. La primera, "Mis incidentes reportados", permite al trabajador revisar los últimos incidentes que ha reportado. La segunda opción, denominada "Informar incidente", brinda al trabajador la posibilidad de registrar un nuevo incidente y seguir los pasos para su resolución. Por último, el botón de "Salir" proporciona al usuario la opción de cerrar sesión y desconectarse de la interfaz.

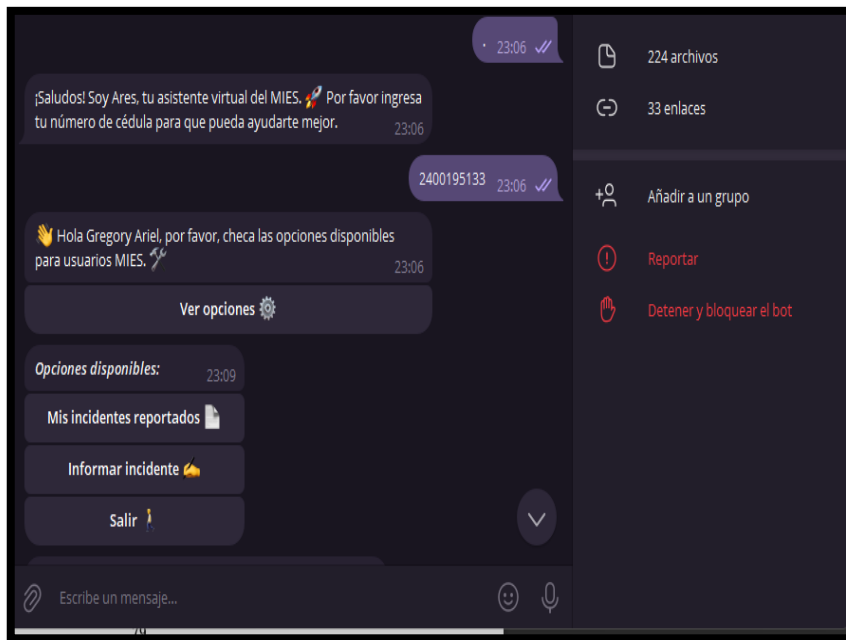


Ilustración 15 Interfaz trabajador.

Interfaz de trabajador – Mis Incidentes Reportados

En la ilustración 16, se observa cómo Ares muestra información sobre los últimos incidentes reportados por el trabajador. Dentro de esta visualización, se incluye el detalle del incidente, su estado, título y el responsable del ticket. Además, se proporciona un enlace donde el usuario podrá visualizar los incidentes a través de la integración con Zammad. Así mismo, la opción ofrece la posibilidad de volver al menú o salir.

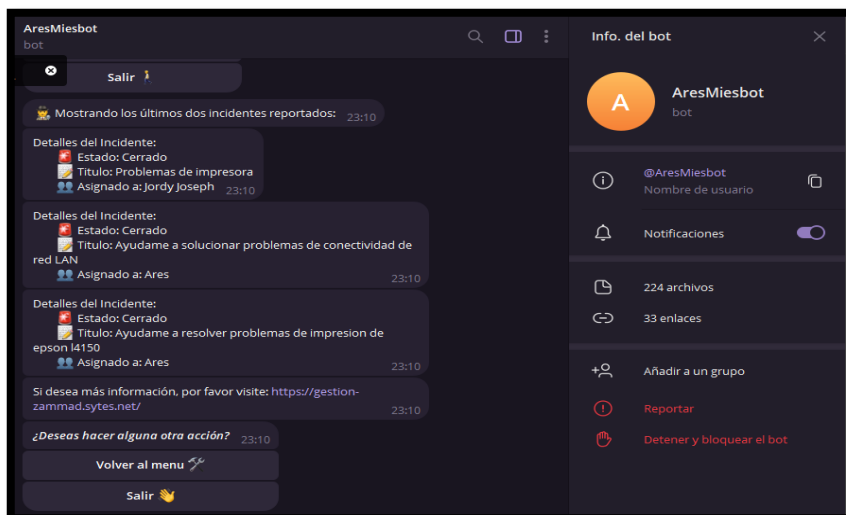


Ilustración 16 Interfaz trabajador - Incidentes Reportados.

Interfaz de trabajador – Informar Incidente

Como se observa en la ilustración 17, el usuario con rol de trabajador selecciona la opción de reportar un incidente. Inmediatamente, Ares responde solicitando al usuario que proporcione detalles sobre el incidente. Una vez que el usuario introduce estos datos, Ares utiliza la API de Chat GPT para brindar cinco posibles soluciones al usuario, quien luego podrá elegir cuál desea leer. Además, Ares ofrece dos botones: uno para enviar directamente el incidente a Zammad y otro para ver la solución propuesta.

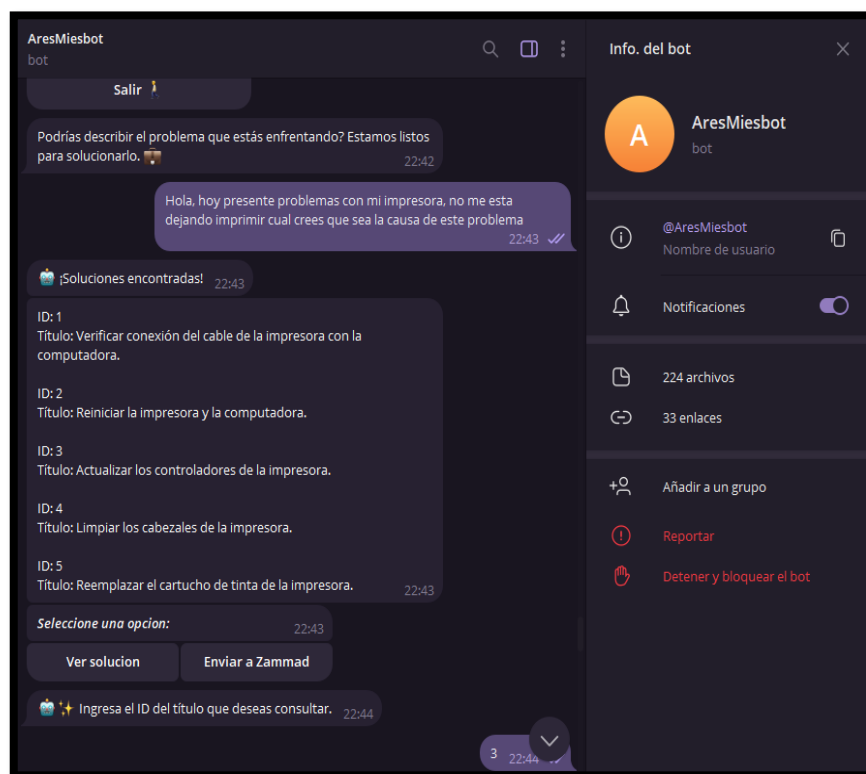


Ilustración 17 Interfaz trabajador - Informar Incidente.

En la ilustración 18, se observa cómo Ares presenta la solución de acuerdo con el título que el usuario eligió, proporcionándole detalladamente los pasos a seguir para solucionar su incidente. Posterior a la solución, Ares ofrece al usuario con rol de trabajador tres opciones: dos de ellas son para confirmación, mientras que la otra es para ver otra solución. Además, Ares demuestra su capacidad para adaptarse a las necesidades del usuario al ofrecer opciones claras y relevantes después de proporcionar la solución

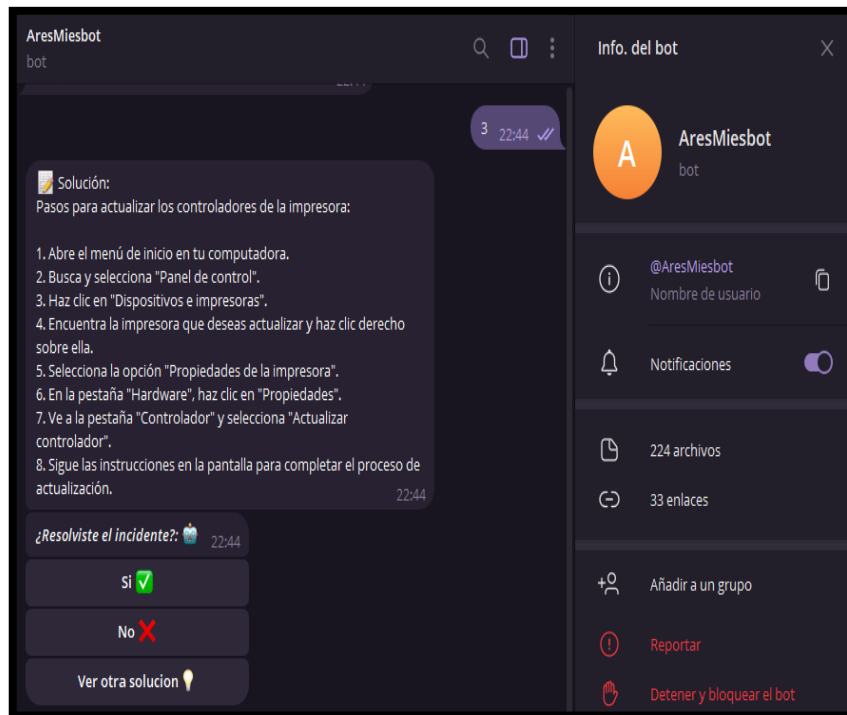


Ilustración 18 Interfaz trabajador - Ver solución.

Interfaz Trabajador – Incidente Resuelto

En la ilustración 19, se observa cómo Ares muestra los botones de confirmación, donde el usuario, al presionar sobre el botón “sí”, confirma que el incidente ha sido resuelto. A continuación, Ares envía un mensaje de resolución exitosa.

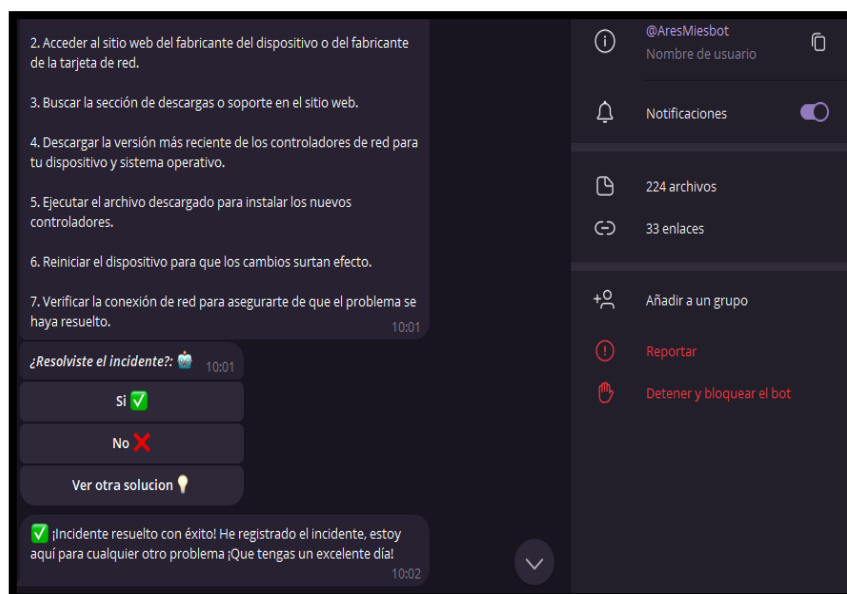


Ilustración 19 Interfaz trabajador - Incidente Resuelto.

Interfaz Trabajador – Incidente No Resuelto

Como se observa en la ilustración 20, cuando el incidente no pudo ser resuelto mediante las sugerencias dadas por Ares, se procede a la creación de un ticket. En este proceso, el usuario debe proporcionar el título del ticket, tras lo cual Ares lo crea y lo envía hacia Zammad. Como mensaje de salida, Ares informa al usuario que el ticket ha sido creado, proporcionando además su respectivo número. Después de esto, Ares cierra la sesión.

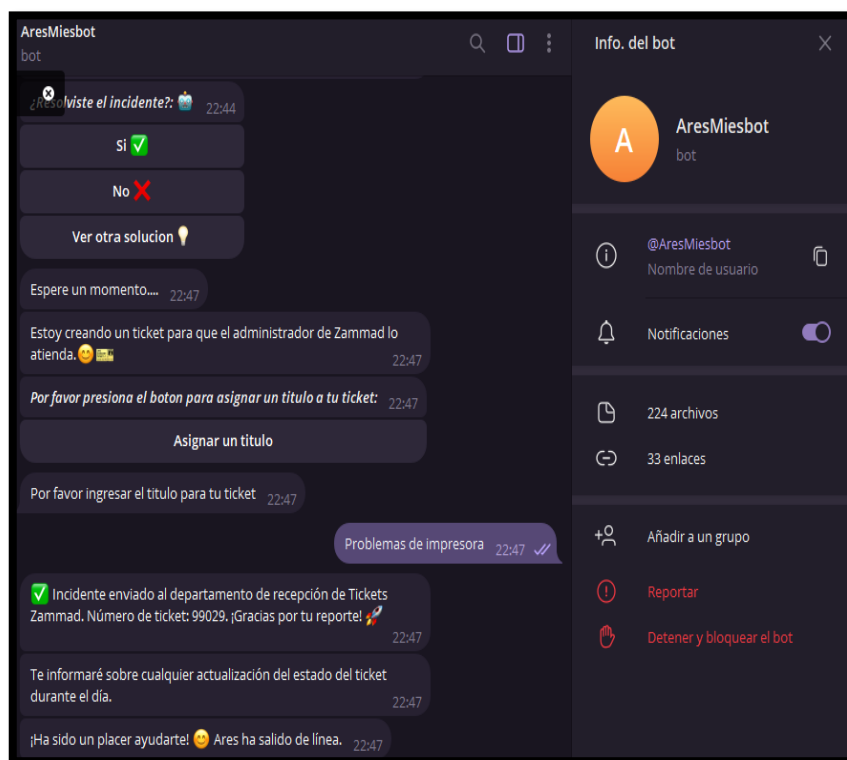


Ilustración 20 Interfaz trabajador - Creación de Tickets

Interfaz de Tickets Zammad

En la ilustración 21, se observa cómo los tickets creados por Ares cuando un incidente no se resuelve llegan al encargado de Zammad. En esta interfaz, se detalla el título del ticket proporcionado por el usuario, así como su descripción. Además, en la parte derecha se muestra a quién se le asignó el ticket, junto con su estado y prioridad. Finalmente, en la parte inferior derecha, se encuentra un botón para actualizar los estados del ticket.

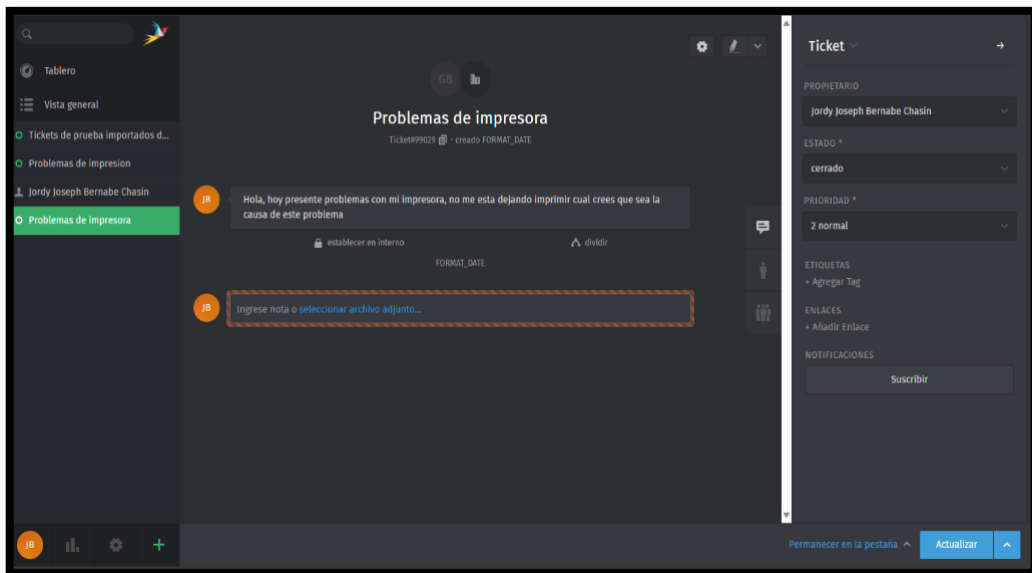


Ilustración 21 Interfaz Zammad.

Interfaz de Administrador

Como se observa en la ilustración 22, el administrador tiene acceso a cuatro opciones. En la primera opción, denominada "Gestionar", el chatbot proporciona un enlace que lo dirigirá hacia la administración de Zammad, donde podrá gestionar los incidentes. En la opción "Solicitar reportes", el administrador puede solicitar reportes diarios y mensuales. En la tercera opción, llamada "Solicitar CSV", el chatbot enviará un archivo CSV con los datos de los incidentes. Por último, la opción de "Salir" permite al administrador cerrar sesión.

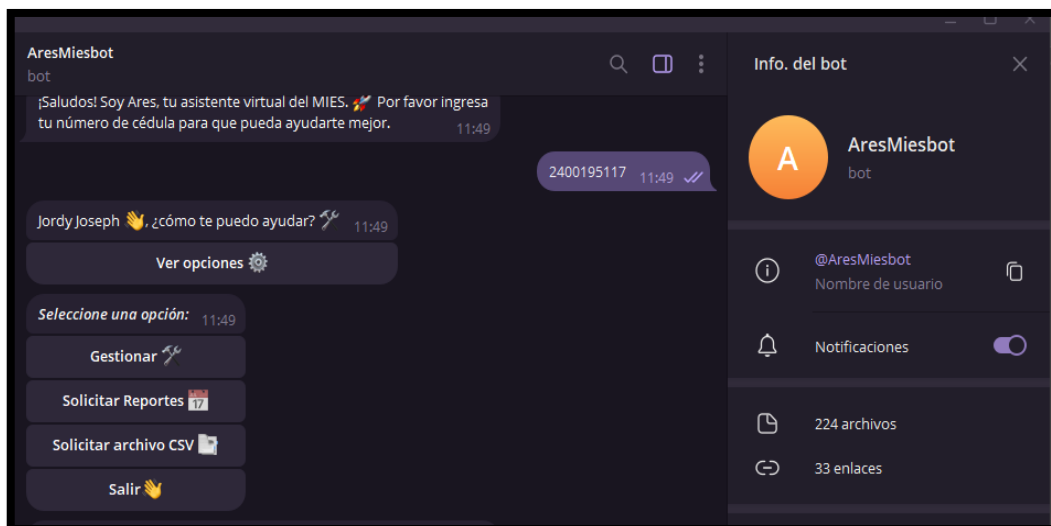


Ilustración 22 Interfaz administrador.

En la ilustración 23, se muestra el mensaje enviado por Ares cuando el usuario administrador inicia sesión y selecciona la opción de gestión. Ares envía un enlace que permite al administrador conectarse con Zammad, una plataforma de gestión de tickets, para supervisar y administrar tickets.

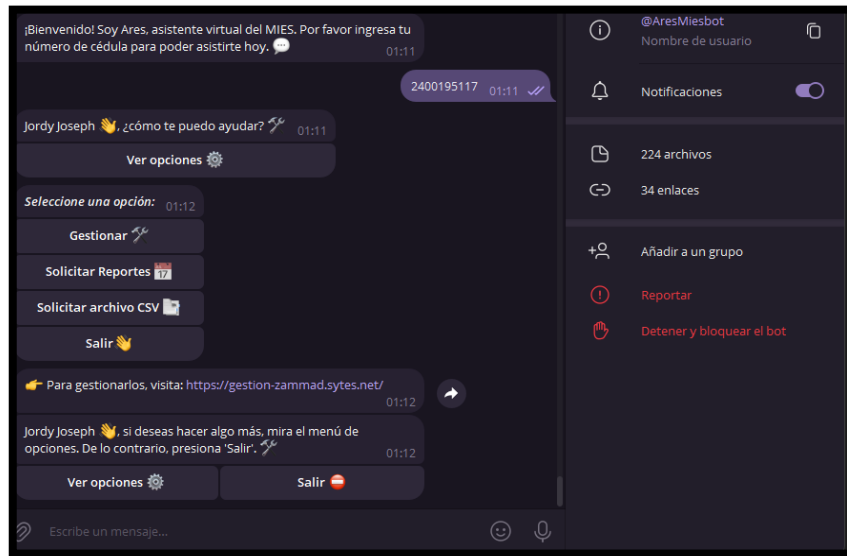


Ilustración 23 Interfaz administrador - Gestionar.

En la ilustración 24, se muestra cómo Ares presenta los botones de reportes diarios y mensuales después de que el usuario selecciona la opción de reportes. En estas opciones, el usuario puede elegir qué tipo de reporte desea ver, y Ares lo enviará en formato PDF.

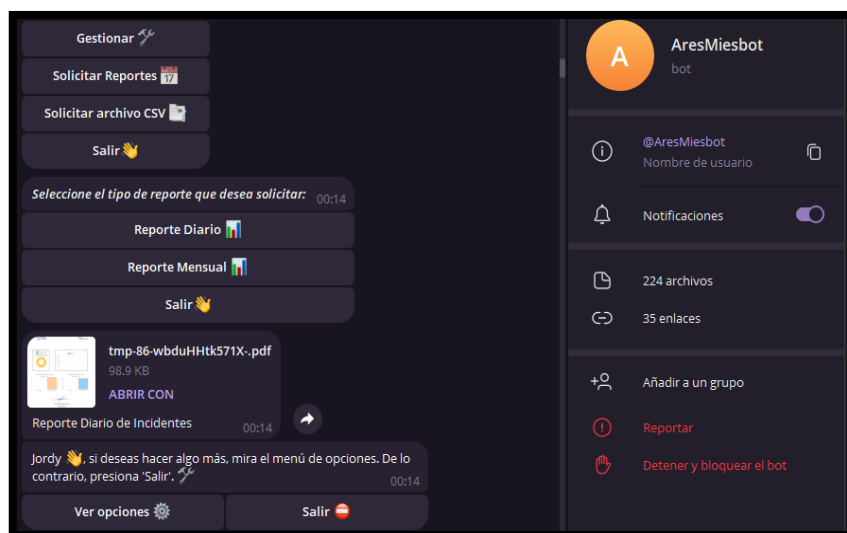


Ilustración 24 Interfaz administrador – Reportes.

En la ilustración 25, además de mostrar los botones para generar reportes en formato PDF, Ares ofrece un reporte completo a través de un enlace que se conecta con el servidor de Grafana, tal como se muestra en la ilustración 24.

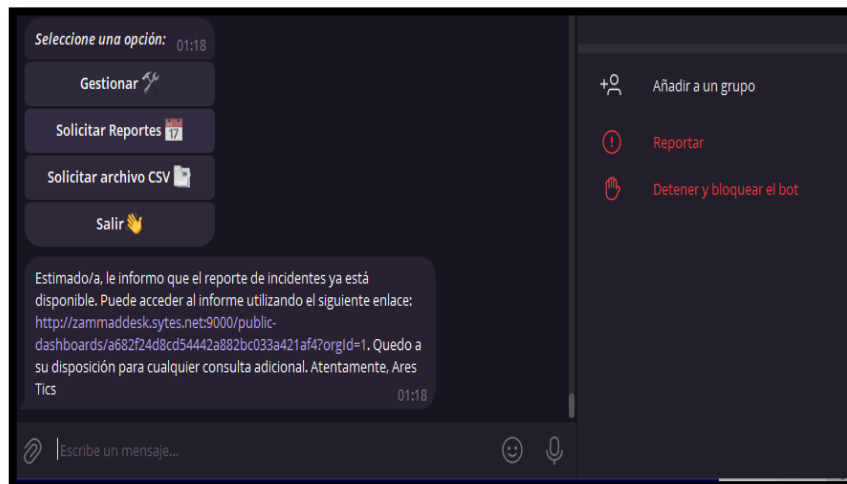


Ilustración 25 Enlace de reportes de Grafana.

Interfaz de Inicio de Sesión de Grafana

En la ilustración 26, se muestra la interfaz de Grafana al administrador, quien accede mediante el enlace proporcionado por el chatbot. Aquí, se requiere validar las credenciales de administrador para poder acceder a los dashboards de reportes.

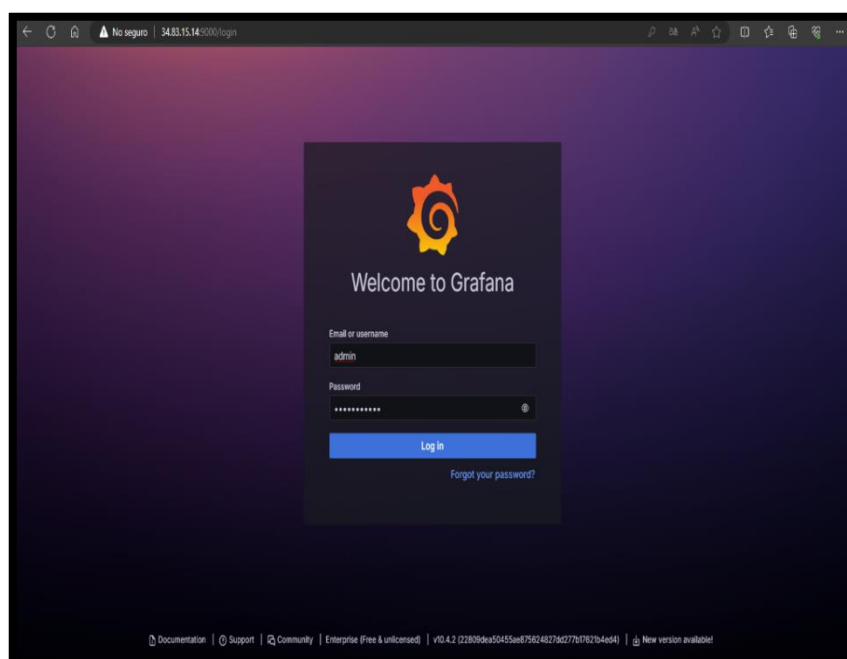


Ilustración 26 Interfaz Grafana – Inicio de sesión.

Interfaz de Reportes de Grafana

Una vez que el administrador inicia sesión, tiene la capacidad de examinar con mayor detalle el reporte de incidentes. Esta funcionalidad se habilita gracias a la integración entre Grafana y Zammad. Esta integración permite que los datos se presenten en tiempo real en paneles de fácil comprensión. Observar la ilustración 27 para visualizar este proceso en acción. La interfaz de dashboard de Grafana es altamente intuitiva, ofreciendo opciones de personalización que permiten a los usuarios adaptar los paneles a sus necesidades específicas. Los gráficos interactivos facilitan la identificación de tendencias y patrones, mientras que las alertas configurables aseguran que los administradores se mantengan informados sobre cualquier cambio crítico en los datos. Además, la capacidad de combinar múltiples fuentes de datos en un solo panel proporciona una visión holística y detallada del rendimiento y los incidentes del sistema

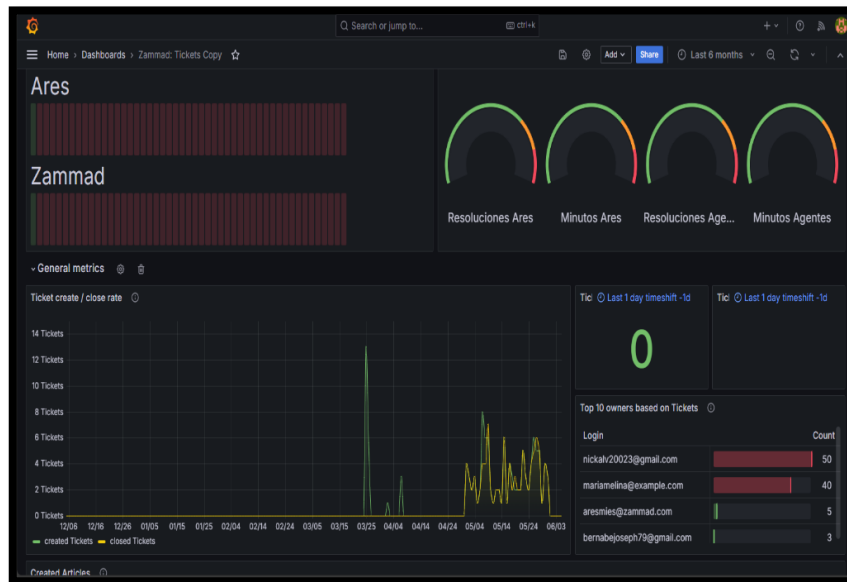


Ilustración 27 Interfaz Grafana - Reportes.

Interfaz de Archivo CSV

En la ilustración 28, se muestra el apartado de solicitud de archivo CSV de tickets. Cuando el usuario presiona esta opción, solicita al chatbot un archivo CSV que contiene información sobre los tickets. Ares responde de inmediato con el archivo CSV deseado.

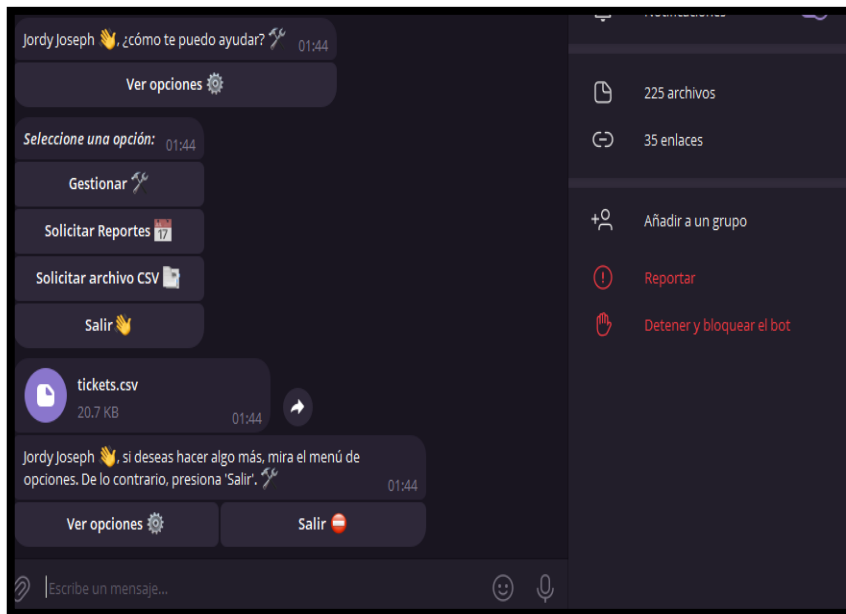


Ilustración 28 Interfaz administrador - Archivo CSV.

Interfaz Cierre de Sesión

Cuando el usuario con rol de administrador o trabajador cierra la sesión, el chatbot ya no permitirá la selección de ninguna opción de procesos anteriores. En su lugar, mostrará una advertencia indicando que es necesario iniciar sesión nuevamente, tal como se muestra en la ilustración 29.

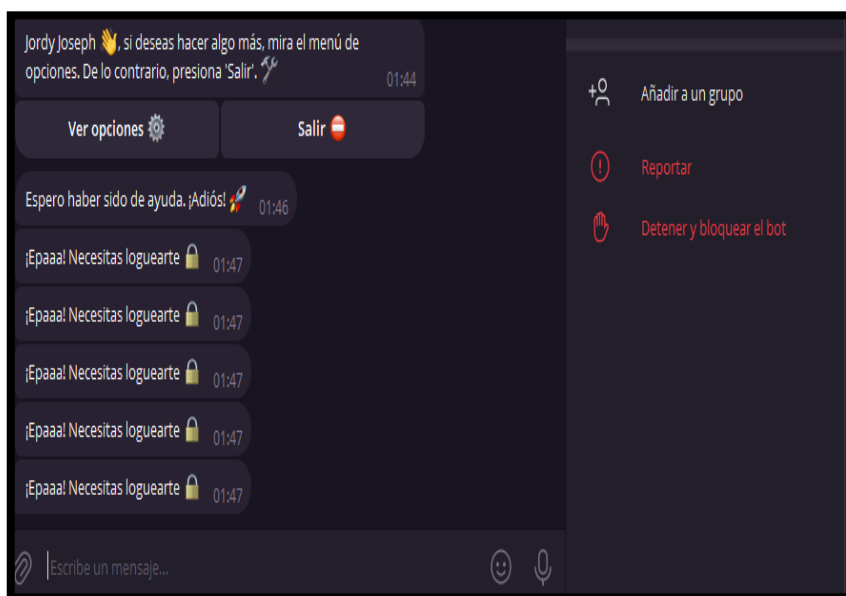


Ilustración 29 Interfaz cierre de sesión.

Interfaz de DialogFlow

En esta interfaz se contabilizan los intentos realizados por Ares para ejecutar las acciones requeridas. El sistema procesa las entradas del usuario en Telegram y ejecuta las acciones pertinentes, interactuando con el webhook de acciones. Se lleva un registro detallado de cada intento, permitiendo un seguimiento preciso y análisis de fallos. Las entradas de los usuarios en Telegram son capturadas y procesadas en tiempo real, garantizando una respuesta ágil. La interfaz se comunica continuamente con el webhook de acciones para actualizar el estado de las tareas y recibir nuevas instrucciones, manteniendo una sincronización óptima. Este flujo de trabajo observa en la ilustración 30.

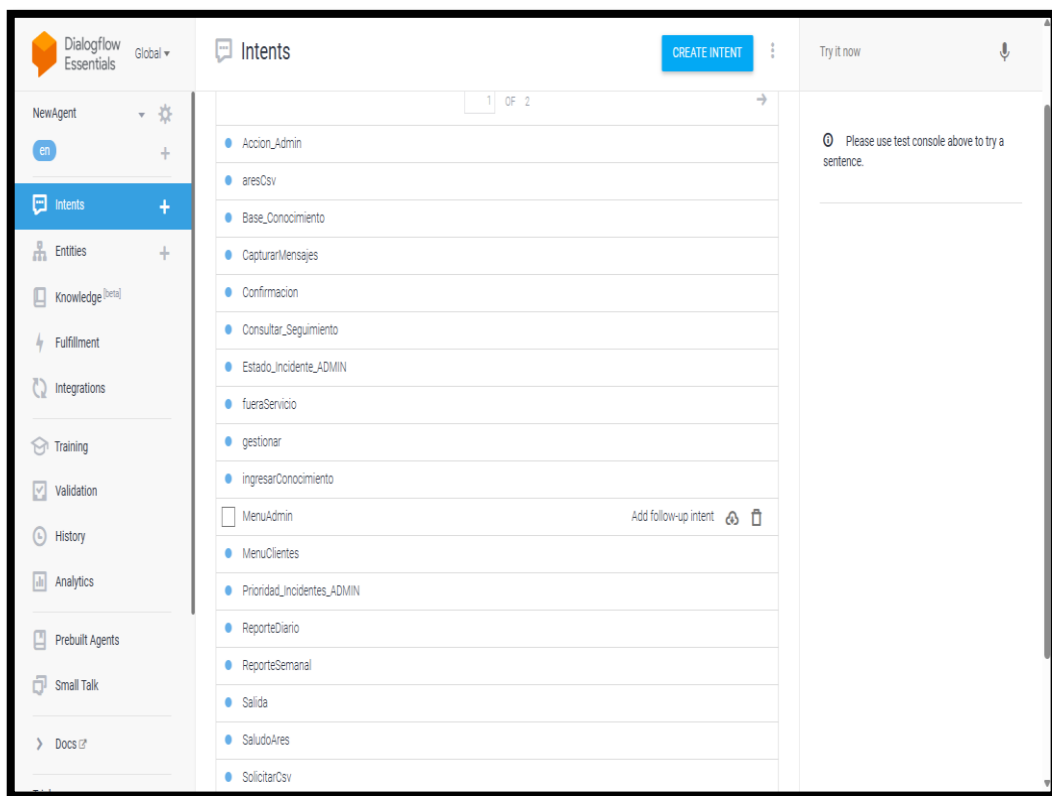


Ilustración 30 Interfaz DialogFlow.

Notificaciones de Notify bot

Mediante el uso de un bot de notificaciones llamado NotifyBot, se notificará al usuario el estado del ticket que reportó, proporcionándole información sobre el número de ticket y un mensaje de agradecimiento por su colaboración. Esto se observa en la ilustración 31.

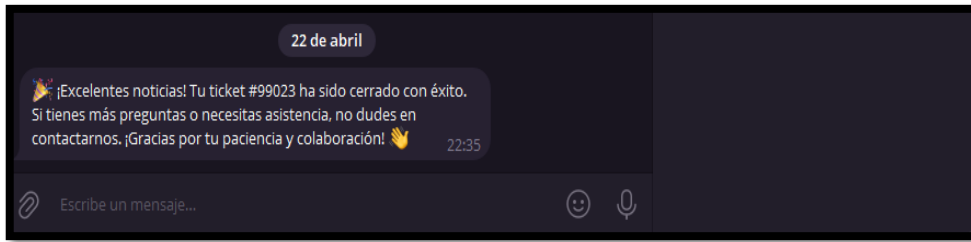


Ilustración 31 Bot de notificaciones.

Comunicación entre DialogFlow y el Webhook Node

En la ilustración 32 se muestra un fragmento del código empleado para la comunicación entre el webhook Node y Dialogflow. Se implementa una solicitud `app.post` que permanece a la espera de los intentos del usuario. Dentro de esta solicitud, se obtiene la fecha y la zona horaria actuales con el propósito de validar y garantizar que el chatbot esté disponible únicamente durante los días laborables y dentro del horario operativo establecido. Cuando se cumplen estas condiciones, se procesan los intentos y se ejecutan las acciones correspondientes.

```

app.post("/", express.json(), (request, response) => {
  const horaActual = new Date();
  const diaSemana = horaActual.getDay();
  let horaFormateada = horaActual.toLocaleString('es-EC', { timeZone: 'America/Guayaquil', hour12:
false, hour: 'numeric' });

  if (horaFormateada === '0') {
    horaFormateada = '24';
  }
  console.log(horaFormateada);
  const esDialaboral = (diaSemana >= 1 && diaSemana <= 5);
  if (horaFormateada >= 8 && horaFormateada < 17 && esDialaboral) {
    const agent = new WebhookClient({ request, response });
    console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
    console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
    const body = request.body;
    if (body &&
        body.originalDetectIntentRequest &&
        body.originalDetectIntentRequest.payload &&
        body.originalDetectIntentRequest.payload.data &&
        body.originalDetectIntentRequest.payload.data.chat &&
        body.originalDetectIntentRequest.payload.data.chat.id) {
      const chatId = body.originalDetectIntentRequest.payload.data.chat.id;
      console.log('Chat ID:', chatId);
    } else {
      console.log('La estructura de la solicitud no es la esperada. No se pudo obtener el chat ID.');
```

Ilustración 32 Comunicación Webhook con DialogFlow.

En el fragmento de código se ilustra el proceso de realizar una petición a la API de ChatGPT. Inicialmente, se valida que la descripción del incidente no esté vacía ni nula. Posteriormente, se define el prompt junto con la descripción del incidente para obtener una respuesta utilizando el modelo GPT-3.5 Turbo. Finalmente, se retorna una respuesta que será enviada al usuario a través de Telegram. El código correspondiente se muestra en la ilustración 33.

```
async function buscarSolucionBaseConocimientos(descripcionInciGlobal) {
  try {
    if (!descripcionInciGlobal) {
      console.error('La descripción del incidente es nula o está vacía');
      return null;
    }
    const prompt = `Quiero que me des soluciones sobre esto "${descripcionInciGlobal}" quiero que me des soluciones con un identificador, es decir, una ID. Por el momento, solo quiero la ID que sea numerica y corta y un título de solución para que pueda elegir mediante el ID el que desee y que sean máximo 5 y que tengan una linea de espacio entre sí, y que sean solo IDs del 1 al 5 (ejemplo) ID: 1, ID:2, etc`;

    const response = await openaiClient.chat.completions.create({
      model: 'gpt-3.5-turbo',
      messages: [{ role: 'user', content: prompt }],
    });

    const aiResponse = response['choices'][0]['message']['content'] || '';
    globalGpt = aiResponse;
    return aiResponse;
  } catch (error) {
    console.error('Error al buscar solución con OpenAI:', error);
    throw error;
  }
}
```

Ilustración 33 Solicitud a la API de ChatGpt.

Comunicación Webhook con Zammad

La comunicación entre Zammad y el webhook que controla al chatbot es esencial. En la ilustración 34 se muestra cómo se gestionan estas interacciones para la creación de un ticket. Se envía una solicitud POST al servidor de Zammad utilizando las credenciales adecuadas. Una vez que las credenciales son validadas, se envía un cuerpo JSON con los datos del ticket mediante Axios, y se espera la respuesta del servidor Zammad para la confirmación del ticket.

```
app.post('/crearTicket', async (req, res) => {
  try {
    // Configura la URL de la API de Zammad y tu token de autenticación
    const apiUrl = 'http://34.83.15.14/api/v1/tickets';
    const authToken = 'OWnZYRTmPPFiE_QAuHu8PcGxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx';

    // Datos del nuevo ticket a crear
    const nuevoTicket = {
      title: tituloZammad,
      group_id: 1,
      customer_id: usuario_cedula, // Cliente
      organization_id: 1, // MIES
      owner_id: 3, // encargado de los tickets jordybernabe

      article: {
        type: 'web',
        internal: false,
        customer_id: usuario_cedula,
        subject: 'Incidentes',
        body: descripcionTickets,
        origin_by_id: idRegistroTickets,
      }
    };

    const response = await axios.post(apiUrl, nuevoTicket, {
      headers: {
        'Authorization': `Bearer ${authToken}`,
        'Content-Type': 'application/json',
      },
    });

    res.json(response.data);
  } catch (error) {
    console.error('Error al crear el ticket:', error);
    res.status(500).json({ error: 'Error al crear el ticket' });
  }
});
```

Ilustración 34 Comunicación con Zammad.

Creación de usuarios para el chatbot

En la Ilustración 35 se presenta de manera destacada la capacidad del administrador de Zammad para crear usuarios destinados al chatbot, ofreciendo una experiencia intuitiva y completa. La interfaz de creación de usuarios en Zammad solicita una variedad de datos relevantes, incluyendo nombre, apellido, correo electrónico, teléfono, organización y contraseña. Además, se proporciona la opción de asignar privilegios, permitiendo que los usuarios puedan ser designados como clientes o agentes de tickets según corresponda. Este enfoque detallado y completo garantiza que

los usuarios sean correctamente registrados en el sistema, con todos los datos necesarios para una interacción con el chatbot.

Una vez que los usuarios son creados exitosamente, el acceso al chatbot se facilita gracias a la integración fluida con Zammad. Esto garantiza que los usuarios recién creados puedan iniciar sesión sin problemas y comenzar a utilizar las funcionalidades del chatbot de inmediato. En conjunto, esta capacidad de Zammad para administrar usuarios de manera eficiente y proporcionar un acceso sin problemas al chatbot contribuye significativamente a una experiencia de usuario positiva y a una gestión eficaz de la plataforma de asistencia.

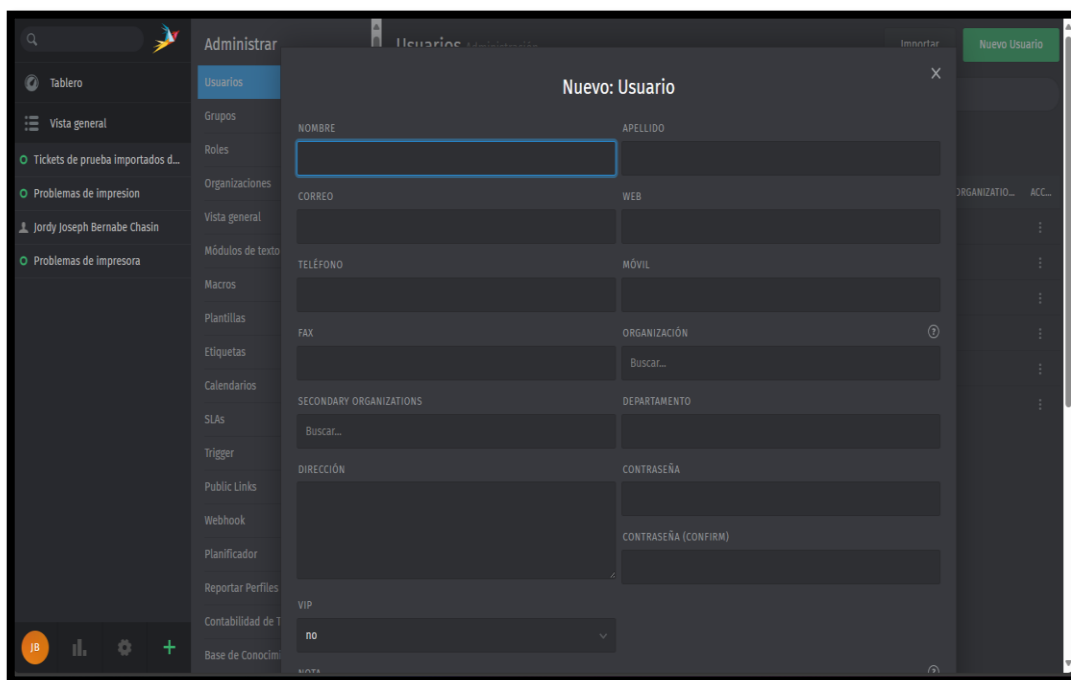


Ilustración 35 Creación de Usuarios para el chatbot.

2.6 Pruebas

Para asegurar que el chatbot funcione de manera óptima, es crucial realizar pruebas de funcionalidad. Estas pruebas tienen como objetivo verificar que todas las funcionalidades del asistente se ejecuten correctamente y conforme a los requerimientos especificados. Además, es esencial realizar pruebas de integración para garantizar que el chatbot interactúe adecuadamente con otros sistemas y servicios. Esta fase de pruebas es fundamental para identificar y corregir cualquier fallo.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	001	
CASO DE USO	Identificar usuario con su rol	
OBJETIVO DE LA PRUEBA	Controlar las opciones que se muestran a los usuarios en función de los roles asignados dentro de la empresa.	
CONDICIONES	El usuario debe ingresar su número de cédula, el cual será verificado para coincidir con los registros existentes en la base de datos.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • Invocar al chatbot Ares. • Ares solicita al usuario que ingrese su número de cédula. • El usuario ingresa su número de cédula. • Ares verifica el número de cédula en la base de datos. • Una vez identificado, Ares saluda al usuario por su nombre. • Ares muestra las opciones disponibles según el rol del usuario. 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
El chatbot autentica al usuario al iniciar sesión, utiliza su nombre para un saludo personalizado y luego presenta un menú de opciones adaptado a su rol dentro del sistema.	Exitoso	Fallido
		

Tabla 18 Prueba de funcionalidad – Identificar usuario con su rol.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	002	
CASO DE USO	Informar un incidente.	
OBJETIVO DE LA PRUEBA	Confirmar que el usuario pueda reportar su incidente y recibir sugerencias relacionadas.	
CONDICIONES	El usuario debe proporcionar una descripción detallada del incidente.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • El usuario seleccionará la opción de informar un incidente. • Ares solicitará al usuario que ingrese el detalle del incidente. • El usuario ingresará la descripción del incidente. • Ares consultará a la API de ChatGPT y obtendrá posibles soluciones. • Ares presentará al usuario varias opciones de solución. • El usuario elegirá una opción y leerá la solución a su incidente. 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
El chatbot guía al usuario a través de un proceso, para capturar los detalles del incidente. Posteriormente, emplea la API de ChatGPT para generar respuestas que se ajustan al contexto del incidente y ofrecen soluciones relevantes para el usuario.	Exitoso	Fallido
		

Tabla 19 Prueba de funcionalidad – Informar Incidente.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	003	
CASO DE USO	Generar un ticket en Zammad.	
OBJETIVO DE LA PRUEBA	Verificar que Ares cree correctamente un ticket en Zammad cuando el incidente no ha sido resuelto utilizando las sugerencias proporcionadas.	
CONDICIONES	Cuando no hay solución se crea un ticket.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • Ares consulta al usuario si las sugerencias han solucionado el incidente. • El usuario confirma que el incidente no ha sido solucionado. • Ares solicita al usuario que asigne un título al ticket. • El usuario ingresa el título del ticket. • Ares presenta un mensaje de confirmación junto con el número de ticket. • Ares finaliza la sesión. 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
Ares se integra con Zammad para crear un ticket cuando el usuario no logra resolver el incidente con las sugerencias proporcionadas previamente.	Exitoso	Fallido
		

Tabla 20 Generar un ticket en Zammad.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	004	
CASO DE USO	Consultar incidentes.	
OBJETIVO DE LA PRUEBA	Validar que el usuario pueda consultar los incidentes que ha reportado.	
CONDICIONES	Para activar la opción de consultar incidentes, el usuario selecciona esta función específica.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • Ares presenta el menú de opciones. • El usuario selecciona la opción "Consultar incidentes". • Ares consulta la base de datos en busca de los últimos incidentes reportados por el usuario. • Ares muestra un mensaje que contiene la descripción de los últimos incidentes reportados. 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
Los resultados evidencian una interacción efectiva con Ares, permitiendo al usuario acceder fácilmente a la opción de consulta de incidentes. Ares recuperó con éxito los últimos incidentes reportados del usuario, demostrando así la capacidad de respuesta del chatbot.	Exitoso	Fallido
		

Tabla 21 Prueba de funcionalidad – Consultar Incidentes.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	005	
CASO DE USO	Solicitar Reportes.	
OBJETIVO DE LA PRUEBA	Validar que el usuario pueda solicitar reportes mensuales y diario de incidentes.	
CONDICIONES	Para habilitar la función de reportes, el usuario debe seleccionar la opción correspondiente y tener privilegios de administrador.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • Ares muestra el menú de opciones disponibles. • El usuario elige la opción "Solicitar reportes". • Ares presenta dos opciones de reportes: diario y mensual. • El usuario selecciona la opción deseada. • Ares envía un archivo PDF con el reporte solicitado. • Además, Ares proporciona un enlace de Grafana para acceder al reporte completo. 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
Ares envía el archivo PDF del reporte solicitado, además de ofrecer un enlace de Grafana para acceder al reporte completo.	Exitoso	Fallido
		

Tabla 22 Prueba de funcionalidad – Solicitar Reportes.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	006	
CASO DE USO	Solicitar Archivo Csv.	
OBJETIVO DE LA PRUEBA	Validar que el usuario pueda solicitar archivo CSV de incidentes.	
CONDICIONES	Para habilitar la función de reportes, el usuario debe seleccionar la opción correspondiente y tener privilegios de administrador.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • Ares muestra el menú de opciones disponibles. • El usuario elige la opción "Solicitar CSV". • El usuario selecciona la opción deseada. • Ares envía un archivo CSV de incidentes. 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
	Exitoso	Fallido
Los resultados obtenidos muestran una experiencia satisfactoria para el usuario al interactuar con Ares para solicitar un archivo CSV de incidentes. A través de un menú claro y opciones fáciles de entender, el usuario pudo seleccionar la opción deseada sin dificultad. Ares respondió eficazmente enviando el archivo CSV solicitado.		

Tabla 23 Prueba de funcionalidad – Solicitar CSV.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	007	
CASO DE USO	Cerrar Sesión	
OBJETIVO DE LA PRUEBA	Validar que el usuario una vez que cierra sesión no pueda acceder a las opciones.	
CONDICIONES	El usuario ha cerrado sesión correctamente en el sistema.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • Ares muestra opción de salir o cierra sesión cuando el proceso de registro y creación de tickets a finalizado. • Usuario elije salir de la sesión • Ares no permite ingresar a las opciones una vez que se cerró la sesión • Usuario debe volver a validar sus credenciales 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
Los resultados confirman que el usuario puede cerrar sesión de manera exitosa al finalizar el proceso de registro y creación de tickets. De igual manera puede seleccionar la opción para salir de la sesión, Ares impide el acceso a las opciones, lo que demuestra una correcta restricción de acceso una vez cerrada la sesión.	Exitoso	Fallido
		

Tabla 24 Prueba de funcionalidad – Cerrar Sesión del chatbot.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	008	
CASO DE USO	Creación de usuarios para Ares.	
OBJETIVO DE LA PRUEBA	Validar que el administrador pueda crear usuarios para que sean reconocidos por el chatbot.	
CONDICIONES	El administrador tiene acceso y privilegios para crear nuevos usuarios.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • El administrador inicia sesión en Zammad. • Navega hacia la sección de usuarios. • Crea un nuevo usuario ingresando los datos correspondientes. • Zammad registra y crea al nuevo usuario. • Ares identifica automáticamente al usuario gracias a la integración con Zammad. • El usuario recién creado puede iniciar sesión en el chatbot sin problemas. 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
Los resultados obtenidos demuestran que el proceso de creación de usuarios por parte del administrador en Zammad es exitosa. Una vez creado el usuario, Ares lo identifica automáticamente gracias a la integración con Zammad.	Exitoso	Fallido
		

Tabla 25 Prueba de funcionalidad – Creación de usuarios para Ares.


DATOS DEL CASO DE PRUEBA		
CASO DE PRUEBA N°	009	
CASO DE USO	Tiempos de resolución de incidentes con el chatbot	
OBJETIVO DE LA PRUEBA	Medir el tiempo que tarda el chatbot en resolver incidentes, desde el inicio de sesión del usuario hasta la confirmación de resolución.	
CONDICIONES	Los tiempos se tomarán desde que el usuario inicia sesión en el chat bot hasta que confirme la resolución.	
PASOS DE LA PRUEBA		
<ul style="list-style-type: none"> • El usuario invoca al chatbot • Ares comienza a registrar los tiempos de interacción del usuario con el chatbot. • El usuario confirma que el incidente ha sido solucionado. • Ares cierra y registra el ticket, incluyendo el tiempo total de resolución. • Los tiempos registrados se guardan en la base de datos. 		
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
<i>En la prueba de tiempos de resolución de incidentes con el chatbot, Ares tarda en promedio 8 minutos por incidente, con variaciones entre menos y más de 8 minutos.</i>	Exitoso	Fallido
		

Tabla 26 Prueba de funcionalidad – Tiempos de Resolución Ares.

2.7 Resultados

En este trabajo de titulación, se desarrolló un asistente virtual tipo chatbot para la gestión de incidencias, denominado Ares. Este chatbot permite a los trabajadores del MIES informar sobre incidentes de primer nivel que ocurren a diario y recibir sugerencias de resolución para buscar resolver estos incidentes. Los principales resultados obtenidos en este proyecto son los siguientes:

El chatbot fue desarrollado aprovechando tecnologías emergentes como Zammad, Grafana y Elasticsearch. Estas integraciones permiten la creación de tickets en Zammad y la generación de dashboards estadísticos en Grafana utilizando los datos almacenados en Elasticsearch. Esto posibilita que el encargado de TICs pueda monitorear y analizar los incidentes reportados de manera eficiente.

Además, el chatbot interactúa de manera efectiva con ChatGPT para obtener soluciones a los problemas planteados. Las soluciones proporcionadas por ChatGPT alimentan una base de conocimientos propia de incidentes del MIES, que puede ser utilizada en el futuro en lugar de depender exclusivamente de ChatGPT. Esta integración inteligente permite no solo resolver incidentes de manera eficiente en el presente, sino también construir una base sólida de conocimientos para abordar desafíos futuros con mayor autonomía y precisión.

Un aspecto importante del desarrollo es la notificación automática al encargado de tickets sobre nuevos incidentes, lo que reduce significativamente los tiempos de respuesta. Esto es especialmente relevante dado que el MIES enfrenta el desafío de contar con un único encargado en el área de TICs, quien no siempre está disponible para monitorear los incidentes en tiempo real. Gracias a esta funcionalidad, el encargado recibe notificaciones inmediatas cada vez que se registra un nuevo incidente.

Utilizando el método de recolección de información a través de entrevistas dirigidas al encargado del área de TICs (ver Anexo 1), se recopilaron datos relevantes, incluyendo los tiempos de resolución de incidentes básicos, urgentes y normales. Para incidentes con prioridad alta, el tiempo de resolución oscila entre 2 y 4 horas. Los incidentes de prioridad normal tienen un tiempo de resolución estimado entre 8 y 15

horas. Por último, los incidentes de prioridad baja se resuelven en un lapso de 20 a 24 horas. Estos tiempos se detallan en la Tabla 26.

TIEMPOS DE RESOLUCION DE INCIDENTES “MESA DE AYUDA”

<i>PRIORIDAD</i>	<i>TIEMPO</i>
<i>ALTO</i>	2 - 4 horas
<i>NORMAL</i>	8 - 15 horas
<i>BAJO</i>	20 – 24 horas

Tabla 27 Recolección de datos – Tiempos de Resolución “Mesa de ayuda”

Con la ayuda de Ares, se han reducido significativamente los tiempos de resolución de incidentes básicos de primer nivel, logrando resolver estos incidentes en tiempos que van desde 6 a 8 minutos por incidente, siempre y cuando las sugerencias sean útiles para el usuario que reporta el ticket. Si las sugerencias no resultan efectivas, el ticket se transfiere a Zammad, donde se clasifica como un ticket normal y los tiempos de resolución varían entre 2 y 3 horas.

Gracias a las recomendaciones de Ares y a la capacidad del usuario para seleccionar la mejor solución, los tiempos de resolución para los incidentes de prioridad alta, que ahora se escalan a Zammad mediante la creación de un ticket, se han reducido notablemente a un intervalo de entre 1 y 2 horas. Ares notifica de manera inmediata al encargado cuando se produce un incidente, lo que garantiza una respuesta ágil y permite al encargado estar atento a los tickets desde su reporte inicial. Esto se refleja en la Tabla 27.

TIEMPOS DE RESOLUCION DE INCIDENTES “ARES Y ZAMMAD”

<i>PRIORIDAD</i>	<i>TIEMPO</i>
<i>ALTO</i>	1 - 2 horas
<i>NORMAL</i>	2 - 3 horas

PRIORIDAD

TIEMPO

BAJO

~8 minutos

Tabla 28 Recolección de datos – Tiempos de Resolución “Ares y Zammad”

CONCLUSIONES

- En conclusión, el uso de herramientas de software libre representa una alternativa valiosa para las empresas. En el proyecto se emplearon tres herramientas de software libre que cumplen eficazmente con el primer objetivo de resolver problemas relacionados con TIC. La implementación de Zammad como sistema de gestión de tickets tuvo un impacto positivo, ya que es una solución completa alineada con los estándares de ITIL V4. Telegram, al ser de código libre, facilitó la creación e integración de bots, desempeñando un rol crucial en la comunicación. Así mismo, Grafana permitió la visualización de reportes y gráficos estadísticos, lo que fue esencial para el monitoreo y análisis del proyecto.
- El asistente virtual de tipo chatbot se desarrolló utilizando ocho módulos interconectados, cada uno diseñado para abordar aspectos específicos de la interacción con el usuario. La integración de servicios y el uso de APIs, junto con la potencia y flexibilidad de Node.js, facilitaron significativamente el desarrollo del chatbot. Además, el uso de lenguajes de programación fáciles de usar como JavaScript permitió una rápida iteración y adaptación a los requisitos cambiantes del proyecto.
- Durante las pruebas, se midió la capacidad de respuesta del chatbot, demostrando que, gracias a la ayuda de la API de GPT, siempre respondió adecuadamente a los incidentes reportados. Esto redujo significativamente los tiempos de resolución de problemas básicos y aseguró la correcta asignación y envío de tickets al sistema de gestión Zammad. Además, con las respuestas de los incidentes resueltos, se creó una base de conocimientos para manejar futuros problemas de manera más efectiva.
- Finalmente, el chatbot se integró exitosamente con el sistema de gestión de tickets Zammad mediante el uso de su robusta API. Esto permitió que todos los incidentes reportados se registraran correctamente en la base de datos de Zammad, garantizando un acceso eficiente y preciso gracias a la naturaleza de código libre del sistema.

RECOMENDACIONES

- El proyecto fue desarrollado utilizando el servicio de interfaces conversacionales conocido como DialogFlow. Sin embargo, se recomienda migrar a Rasa, una alternativa de código abierto que ofrece características similares. A diferencia de DialogFlow, que impone un límite de 2000 intentos mensuales, Rasa no presenta esta restricción y también proporciona integración directa con Telegram, facilitando así la creación y despliegue del bot.
- En el presente proyecto se utilizó Render como servicio de hosting. Sin embargo, podría ser ventajoso considerar el uso de Heroku debido a su madurez y estabilidad en el mercado, un ecosistema robusto con una amplia variedad de add-ons e integraciones nativas, y una documentación extensa respaldada por una gran comunidad de usuarios y desarrolladores. Heroku también proporciona un flujo de trabajo DevOps refinado, permitiendo despliegues continuos y una infraestructura de escalado altamente eficiente. Para proyectos que buscan una plataforma fiable y un ecosistema de desarrollo más completo, Heroku podría ofrecer mayores beneficios a largo plazo.
- Es crucial implementar autenticación de dos factores (2FA) en el proyecto. Inicialmente, solo se utilizó el número de cédula de los usuarios para la autenticación. Sin embargo, sería más seguro requerir tanto el número de cédula como un código de verificación enviado al teléfono celular del usuario. No se implementó esta característica debido a los costos potenciales asociados con el envío de SMS.
- Además, se recomienda integrar las funcionalidades del chatbot en la aplicación de mensajería WhatsApp, dada su popularidad y amplia aceptación comercial. En este proyecto, no se implementó esta integración debido a los posibles costos adicionales asociados, ya que WhatsApp es propiedad de Meta.
- Con los datos guardados en la base de conocimientos basados en los incidentes resueltos, se recomienda reemplazar la API de GPT para utilizar esta base de conocimientos específica de la empresa. Esto permitiría manejar los datos y resolver incidentes de manera más precisa y autónoma, utilizando soluciones ya implementadas en situaciones reales de la empresa. Al dejar de depender de

la API de GPT, se reducirían los costos asociados con su uso continuo. Estas mejoras no solo incrementarían la seguridad y eficiencia del sistema, sino que también ampliarían su alcance y accesibilidad.

REFERENCIAS

- [1] A. A. Loayza-Uyehara, «Modelo de gestión de incidentes para una entidad estatal,» Revista de la Carrera de Ingeniería de Sistemas Universidad de Lima, Perú, 2016.
- [2] R. Correa, 23 10 2007. [En línea]. Available: <https://www.inclusion.gob.ec/wp-content/uploads/downloads/2012/07/Decreto-mbs-mies.pdf>. [Último acceso: 23 09 20].
- [3] J. Carpio, «Desarrollo de un Asistente Virtual (Chatbot) para la Automatización de la Atención al Cliente,» Universidad de Guayaquil Facultad de Ciencias Matemáticas , Guayaquil, 2019.
- [4] E. A. Llacta, «Implementación de un Asistente Virtual para la Atención al Cliente en Electrocentro S.A de Huancayo,» Universidad Continental Facultad de Ingeniería, Huancayo Peru, 2020.
- [5] L. C. Caba, «Asistente Virtual (chatbot) para la Web de la Facultad de Informática,» Universidad Complutense de Madrid Facultad de Informatica, Madrid España, 2014.
- [6] IBM, «IBM,» [En línea]. Available: <https://www.ibm.com/mx-es/topics/it-infrastructure-library>. [Último acceso: 26 09 2023].
- [7] Microsoft, «Visual Studio,» [En línea]. Available: <https://code.visualstudio.com/docs>. [Último acceso: 23 09 2023].
- [8] Google, «Google Cloud,» [En línea]. Available: <https://cloud.google.com/dialogflow>. [Último acceso: 26 09 2023].
- [9] PostgreSQL, «PostgreSQL,» [En línea]. Available: <https://www.postgresql.org/>. [Último acceso: 26 09 2023].


- [10] render, «render,» [En línea]. Available: <https://render.com/blog>. [Último acceso: 26 09 2023].
- [11] GitGub, «GitHub,» [En línea]. Available: <https://docs.github.com/es/get-started/using-git/about-git>. [Último acceso: 26 09 2023].
- [12] Zammad, «Zammad,» [En línea]. Available: <https://docs.zammad.org/en/latest/>. [Último acceso: 03 04 2024].
- [13] Nodejs, «Nodejs,» [En línea]. Available: <https://nodejs.org/es/about>. [Último acceso: 27 09 2023].
- [14] Expressjs, «Expressjs,» [En línea]. Available: <https://expressjs.com/>. [Último acceso: 27 09 2023].
- [15] M. W. Docs, «MDN Web Docs,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/JavaScript>. [Último acceso: 27 09 2023].
- [16] Telegram, «Telegram FAQ,» [En línea]. Available: <https://telegram.org/faq>. [Último acceso: 04 03 2024].
- [17] OpenAI, «OpenAI docs,» [En línea]. Available: <https://platform.openai.com/docs/quickstart?context=python>. [Último acceso: 24 03 2024].
- [18] Grafana, «Grafana Labs,» [En línea]. Available: <https://grafana.com/oss/grafana/>. [Último acceso: 20 05 2024].
- [19] Elastic, «Elastic,» [En línea]. Available: <https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html>. [Último acceso: 20 05 2024].
- [20] ITIL, ITIL FOUNDATION ITIL 4 EDITION, 2021.

- [21] S. PLANIFICACION, «Plan de Creacion de Oportunidades 2021-2025,» Quito, 2021-2025.
- [22] R. H. Sampieri, Metodologia de la investigacion 5ta Edicion, Mexico: McGrawHill.
- [23] Ministerio de Inclusión Económica y Social, «Ministerio de Inclusión Económica y Social,» [En línea]. Available: <https://www.inclusion.gob.ec/misionvision/>. [Último acceso: 06 05 2024].
- [24] J. M. R.-F. Raúl Oltra-Badenes, «HERRAMIENTA PARA LA EVALUACION DE LA ADECUACION DE SOFTWARE AL PROCESO DE GESTION DE INCIDENTES ITIL,» 2014.
- [25] G. L. Huertas Camacho, «ITIL.V4 en el proceso de Gestión de Incidencias en el área de TI,» Lima, 2023.
- [26] F. Itil, de *ITIL 4 Edición*, Axelos, pp. 163-164.
- [27] «ambit,» 03 11 2020. [En línea]. Available: <https://www.ambit-bst.com/blog/cu% C3%A1 les-son-los-niveles-de-soporte-it>. [Último acceso: 27 05 2024].
- [28] J. E. M. Díaz, «Tecnologías emergentes, reto para la educación superior colombiana,» Barranquilla, 2017.
- [29] J. J. Fernández Santiago, «Desarrollo de chat-bot para plataforma basada en moodle orientada a innovación educativa,» Madrid, 2022.
- [30] J. R. Q. F. F. M. Susana Graciela Pérez Ibarra, «HERRAMIENTAS Y TECNOLOGÍAS PARA EL DESARROLLO WEB DESDE,» 2021.

- [31] S. V. M. Edilson Londoño Castañeda, «DISEÑO E IMPLEMENTACIÓN DE APLICACIÓN TIPO WEBHOOK COMO BACKEND,» UNIVERSIDAD TECNOLÓGICA DE PEREIRA , Pereira, 2020.
- [32] Oracle, «Oracle,» [En línea]. Available: <https://www.oracle.com/mx/database/what-is-database/>. [Último acceso: 21 05 2024].
- [33] G. E. D. Vega, «Arquitectura para diseñar e implementar Web Services,» 2015.
- [34] M. S. A.-L. Luis Fernando Bravo-Encalada, «ITIL v4 en la gestión de solicitudes e incidentes de la mesa de ayuda de la,» 2020.
- [35] J. S. C. GONZALEZ, «Implementacion en la nube de un sistema de Monitoreo de Eventos de Fallas para Infraestructura de Redes y de Seguridad Informática Utilizando la Integración de Zabbix, Grafana y Zammad,» Guayaquil, 2020.
- [36] H. J. Ibáñez, «Desarrollo de un chatbot para la recomendación de eventos o lugares de interés,» 2020.

ANEXOS

Anexo 1. Formato de entrevista.

	<p>UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES TECNOLOGÍAS DE LA INFORMACIÓN</p>
<p>ENTREVISTA DIRIGIDA A DIRECTIVO DE TI DEL MIES</p>	
<p>Objetivo: Conocer la situación actual de la gestión de incidentes de TI en el departamento de Tics del Ministerio de Inclusión Económica y Social (MIES)</p>	
<p>¿Cuál es el enfoque actual del MIES en cuanto a la gestión de incidentes de TI y cuáles son los principales objetivos en este campo?</p>	
<p>Que el flujo de trabajo y los servicios implementados no se vean interrumpidos o afectados durante tiempos prolongados causando problemas en la atención a usuarios y el desempeño de funciones de los servidores públicos. Entre los principales objetivos está el de prevenir o restaurar en el menor tiempo posible cualquier incidente que afecte la calidad de servicio y minimizar el riesgo o impacto en las transacciones efectuadas. Otro objetivo es el de gestionar como máxima prioridad la atención de la recuperación de un servicio caído o la operatividad de un recurso tecnológico utilizado, aplicando una solución temporal hasta implementar una solución permanente para optimizar el tiempo de respuesta de un incidente presentado</p>	
<p>¿Qué políticas y procedimientos se han implementado en el MIES para gestionar y responder a incidentes de seguridad de la información?</p>	
<p>Se han implementado normas y políticas de seguridad que minimicen posibles impactos de riesgos de seguridad ocasionados por usuarios debido a la falta de conocimiento. A nivel de nuestra plataforma de red, se han implementado</p>	

herramientas y directivas de seguridad a través del software de seguridad que controlan y monitorean todos los equipos conectados en el dominio institucional.

¿Cuáles son los principales desafíos que enfrenta el MIES en la gestión de incidentes de TI en la actualidad?

La falta de recurso humano en el área de TICS para dar atención oportuna y eficaz a todos los incidentes presentados, que son atendidos por tickets a través de la mesa de servicios institucionales, es un desafío. El tiempo de atención se prolonga cuando las incidencias se escalan a niveles superiores. Los incidentes de baja prioridad pueden tardar entre 20 a 24 horas en resolverse, los de prioridad normal suelen ser resueltos en 8 a 15 horas, y los de alta prioridad reciben atención inmediata, resolviéndose en un promedio de 2 a 4 horas.

¿Cómo se mide la efectividad de la respuesta a incidentes en su organización, y qué métricas clave se utilizan para evaluar el rendimiento?

La efectividad se mide con indicadores como el tiempo de respuesta del soporte, el KPI que mide los tickets resueltos, y la calidad del servicio que identifica fallas y propone mejoras. Las métricas incluyen: tecnológicas para evaluar la infraestructura de TI, de procesos para medir tiempos de respuesta y cumplimiento, y de servicios para evaluar al personal de TICS mediante encuestas de satisfacción realizadas tres veces al año.

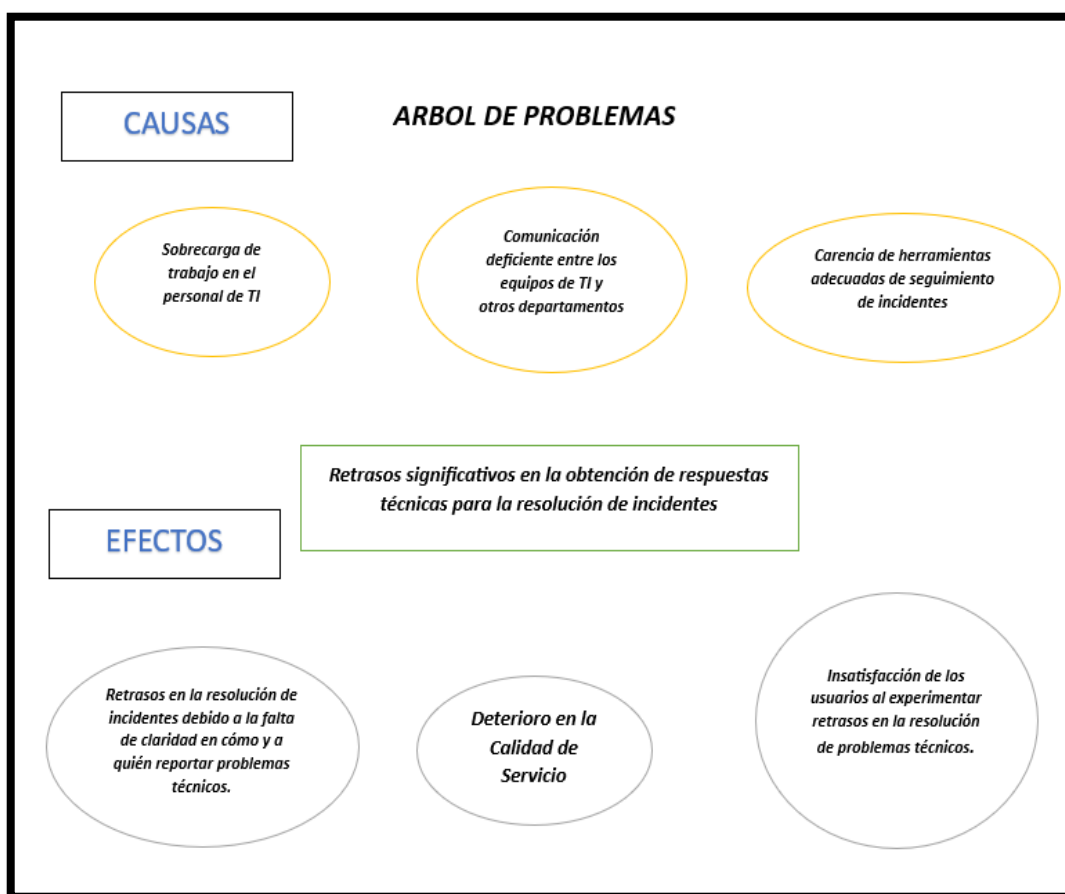
¿Qué roles y responsabilidades específicos existen en su equipo o departamento para abordar y resolver incidentes de TI?

Nuestra dirección distrital no tiene un equipo de TI y solo cuenta con una persona responsable de TI, cuyo rol abarca múltiples funciones: soporte a usuarios mediante herramientas help desk, resolución de problemas en la infraestructura de red local, escalamiento de incidentes de alta prioridad, elaboración de informes técnicos, instalación y análisis de herramientas tecnológicas, configuración y mantenimiento de equipos y redes, y gestión de cuentas de acceso a plataformas

<p>institucionales. Esta única persona debe abordar todos los incidentes reportados a través de la mesa de servicios institucional.</p>
<p>¿Cuál es el proceso típico que sigue el MIES al recibir un reporte de incidente de seguridad de TI? ¿Cómo se priorizan y clasifican los incidentes?</p>
<p>Empieza con un diagnóstico inicial, luego se escala el incidente a un nivel funcional o jerárquico y se clasifica en los niveles de prioridad bajo, normal y alto.</p>
<p>¿Qué tecnologías y herramientas se utilizan en la detección y respuesta a incidentes de TI en el MIES?</p>
<p>Se utiliza la mesa de servicios para la respuesta a incidentes.</p>
<p>¿Cómo se mantiene actualizado el personal en cuanto a las últimas amenazas de seguridad y las mejores prácticas en la gestión de incidentes de TI?</p>
<p>Mediante cápsulas informativas enviadas por la mesa de servicios a todo el personal institucional a través de los canales de comunicación de la empresa, y mediante las normativas y políticas implementadas, se da a conocer y se monitorea el fiel cumplimiento a nivel local a través del responsable de TI.</p>
<p>¿Se realiza algún tipo de ejercicio o simulacro de respuesta a incidentes de forma regular para evaluar la preparación del equipo? ¿Cuáles han sido los resultados más destacados de estos ejercicios?</p>
<p>No se realizan ejercicios o simulacros de respuesta a incidentes. Antes de implementar un recurso tecnológico, ya sea hardware o software, se realizan pruebas piloto para identificar vulnerabilidades, fallas de servicio o procesos. Esto permite realizar mejoras y regulaciones necesarias para asegurar que el equipo o servicio entre en producción con todas las seguridades implementadas.</p>
<p>En el contexto de la transformación digital y los avances tecnológicos, ¿cuáles son las estrategias futuras que el MIES está considerando para mejorar aún más la gestión de incidentes de TI?</p>

Primero, mejorar el flujo de datos a un nivel de seguridad estratégico en toda nuestra plataforma de red. Implementar herramientas de seguridad de software para controlar de manera más eficiente los recursos y los incidentes a nivel nacional. Capacitar al personal mediante medios digitales en tareas y funciones para prevenir amenazas externas que puedan infiltrarse en nuestra red.

Anexo 2. Árbol de Problemas



Anexo 3. Formato de encuesta de satisfacción



**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA
ELENA FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES TECNOLOGÍAS DE
LA INFORMACIÓN**

ENCUESTA DIRIGIDA A DIRECTIVOS DE AREAS

Objetivo: Conocer la satisfacción del usuario, con respecto al proceso de gestión de incidentes que lleva actualmente el MIES.

1.- ¿Qué tan fácil fue reportar un incidente a la mesa de ayuda?

Muy fácil

Facil

Neutral

Difícil

Muy difíci

2.- ¿Entre qué rangos de tiempo, se logra resolver un incidente?

2 - 4 hora

8 - 15 hora

20 - 24 hora

3.- ¿Qué tan satisfecho estuvo con el tiempo de respuesta de la mesa de ayuda?

Muy satisf

Satisfecho

Neutral

Insatisfecho

Muy insat

4.- ¿Qué rápido recibió una respuesta a su solicitud de incidente?

- Muy rápido
- Rápido
- Neutral
- Lento
- Muy lento

5.- ¿Qué tan satisfecho estuvo con la comunicación del agente de la mesa de ayuda?

- Muy satisf
- Satisfecho
- Neutral
- Insatisfecho
- Muy insat

6.- ¿Qué tan satisfecho estuvo con la comunicación del agente de la mesa de ayuda?

- Muy satisf
- Satisfecho
- Neutral
- Insatisfecho
- Muy insat

7.- ¿Qué tan satisfecho estuvo con la resolución de su incidente?

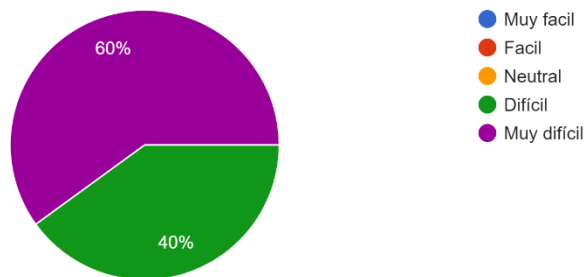
- Muy satisf
- Satisfecho
- Neutral
- Insatisfecho
- Muy insat

8.- ¿En general, ¿qué tan satisfecho estuvo con el proceso de gestión de incidencias?

- Muy satisf
- Satisfecho
- Neutral
- Insatisfecho
- Muy insat

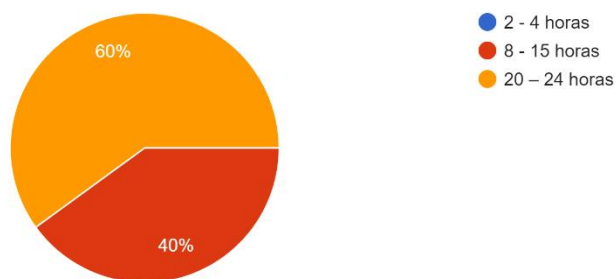
¿Qué tan fácil fue reportar un incidente a la mesa de ayuda?

5 respuestas



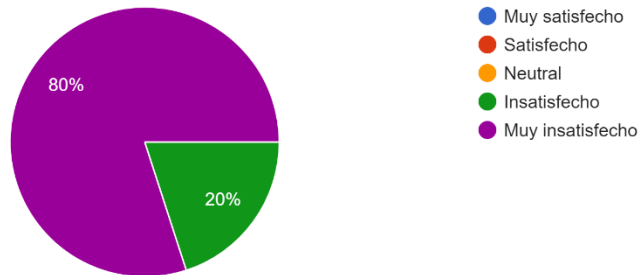
¿Entre qué rangos de tiempo, se logra resolver un incidente?

5 respuestas



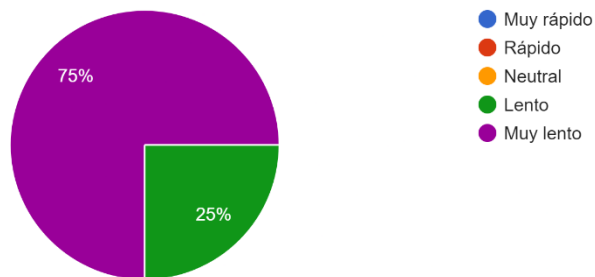
¿Qué tan satisfecho estuvo con el tiempo de respuesta de la mesa de ayuda?

5 respuestas



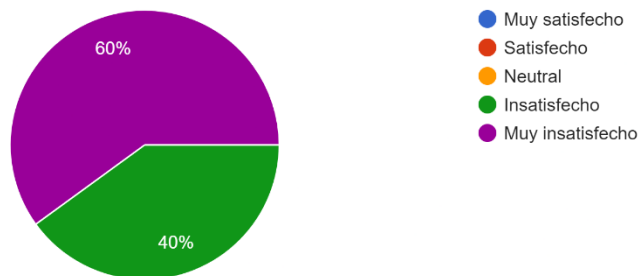
¿Qué tan rápido recibió una respuesta a su solicitud de incidente?

4 respuestas



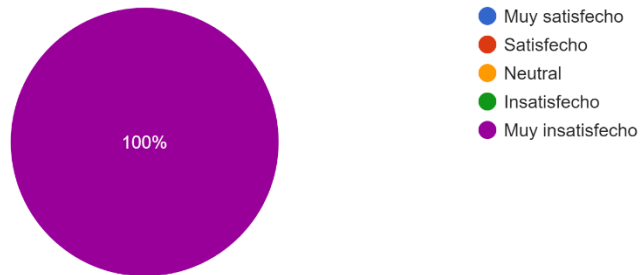
¿Qué tan satisfecho estuvo con la comunicación del agente de la mesa de ayuda?

5 respuestas



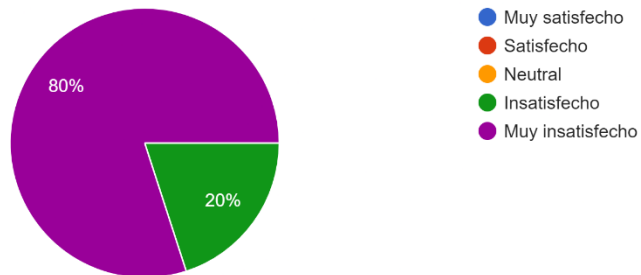
¿En general, qué tan satisfecho estuvo con el proceso de gestión de incidencias?

5 respuestas



¿Qué tan satisfecho estuvo con el tiempo de respuesta de la mesa de ayuda?

5 respuestas



¿Qué tan satisfecho estuvo con la resolución de su incidente?

4 respuestas

