



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA DE COMPUTACIÓN**

**CHATBOT INTELIGENTE PARA SOPORTE TI BASADO EN RASA Y TELEGRAM  
CON NLP**

Trabajo de titulación previo a la obtención del  
Título de Ingenieros en Ciencias de la Computación

**AUTORES: ELKIN DAMIAN CHULCA PASPUEL  
GADIEL SANTIAGO PAHUANQUIZA IZA**

**TUTOR: JOSÉ LUIS AGUAYO MORALES**

Quito - Ecuador  
2026

## **CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

Nosotros, Elkin Damian Chulca Paspuel con documento de identificación N° 1729083103 y Gadiel Santiago Pahuanquiza Iza con documento de identificación N° 1726733627; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 27 de febrero de 2026

Atentamente,



---

Elkin Damian Chulca Paspuel  
1729083103



---

Gadiel Santiago Pahuanquiza Iza  
1726733627

## CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Elkin Damian Chulca Paspuel con documento de identificación N° 1729083103 y Gadiel Santiago Pahuanquiza Iza con documento de identificación N° 1726733627; expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: **“Chatbot inteligente para soporte TI basado en rasa y Telegram con NLP”**, el cual ha sido desarrollado para optar por el título de: Ingenieros en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 27 de febrero de 2026

Atentamente,



---

Elkin Damian Chulca Paspuel  
1729083103



---

Gadiel Santiago Pahuanquiza Iza  
1726733627

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, José Luis Aguayo Morales con documento de identificación N° 1709562597, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **CHATBOT INTELIGENTE PARA SOPORTE TI BASADO EN RASA Y TELEGRAM CON NLP**, realizado por Elkin Damian Chulca Paspuel con documento de identificación N° 1729083103 y por Gadiel Santiago Pahuanquiza Iza con documento de identificación N° 1726733627, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 27 de febrero de 2026

Atentamente,



---

Ing. José Luis Aguayo Morales, MSc.  
1709562597

## **DEDICATORIA**

Dedico este trabajo, en primer lugar, a mis padres, Luis Chulca y Gloria Paspuel, por su amor incondicional, por su esfuerzo durante estos años, por ser mi mayor ejemplo de superación y perseverancia. Todo lo que soy y lo que he logrado se debe a ustedes.

A mi novia, por su apoyo, su paciencia, por cuidarme en estos años, y su acompañamiento en cada etapa de este proceso. Tu confianza, motivación y, sobre todo, el amor fue fundamental para llegar hasta aquí.

A mis familiares, por su respaldo permanente y por creer en mí en todo momento.

Y a mis amigos, por su amistad, por los ánimos y por estar presentes a lo largo de este camino académico.

Elkin Damian Chulca Paspuel

## **DEDICATORIA**

Dedico esta tesis primeramente a Dios, que me da la sabiduría para seguir adelante y la fortaleza para cumplir cada uno de los obstáculos que he encontrado a lo largo de este tiempo.

A mis padres Javier y Daysi, con su gran apoyo y amor me han ayudado día a día a seguir adelante, sobre todo en días donde sentía que no podía.

A mi hermana Annette, que con su alegría me sacaba risa en los momentos más tristes y me hacía olvidar de los momentos malos.

A mis familiares y amigos, con sus palabras me ayudaban a mejorar como persona y estuvieron en toda esta etapa universitaria.

Gadiel Santiago Pahuanquiza Iza

## AGRADECIMIENTOS

A mis padres, mi más profundo agradecimiento por ser fundamental en mi vida y de mi formación. Gracias por cada esfuerzo, por el apoyo constante, por los consejos y por el amor que siempre me han brindado. Su sacrificio y ejemplo me dieron la fortaleza necesaria para seguir adelante y no rendirme ante las dificultades. Este logro es también de ustedes.

A mi novia, gracias por tu paciencia, comprensión y apoyo sincero durante todo este proceso. Gracias por estar presente en los momentos de cansancio y estrés. Tu compañía fue una motivación constante para llegar hasta el final.

A mis profesores, expreso mi agradecimiento por los conocimientos compartidos, la orientación brindada y la dedicación demostrada. Sus enseñanzas, consejos y exigencia contribuyeron de manera significativa a mi crecimiento profesional y al desarrollo de este trabajo de titulación.

A mis amigos, gracias por su amistad, por el apoyo, los consejos y los momentos compartidos durante esta etapa. Su compañía hizo que este camino fuera más llevadero y enriquecedor.

Finalmente, a mis familiares, gracias por su respaldo permanente, por creer en mí y por acompañarme de distintas maneras en este proceso. Este trabajo representa el apoyo y la confianza de todos ustedes.

Elkin Damian Chulca Paspuel

## AGRADECIMIENTOS

A mis padres, por todo el amor que he recibido, por permitirme estudiar en una universidad donde siempre he querido, por su apoyo en los momentos difíciles y por ser un pilar muy importante en mi educación, sin ellos no habría estado donde estoy ahora, cumpliendo mis objetivos y logrando mis sueños, porque muchas veces no entendía sus castigos, pero ahora sé que era para ayudarme a ser un mejor hijo, hermano, persona y estudiante.

A mi hermana, por el cariño y su alegría única, por acompañarme en muchas desveladas cuando teníamos muchos deberes, pero en medio de eso por las risas y las anécdotas únicas que pasamos día a día, con su risa particular me alegraba el día sin importar que no le haya contado como me sentía en cada momento.

A mis abuelitos, por sus palabras de ánimo, por sus consejos sabios, por siempre preguntar cómo me va yendo en mi vida, por ser personas que siempre me hacían sentir seguro y había la confianza suficiente para contar algún problema y aprender de sus experiencias.

A mis amigos Elkin, Erik, David y Anahí, por su acompañamiento dentro de esta universidad, donde día a día luchamos para cumplir nuestros objetivos, junto con muchos obstáculos, con pruebas imposibles, con tareas difíciles, pero me enseñaron que todo es más fácil cuando te rodeas de amistades donde no existe la envidia y el apoyo es mutuo.

Finalmente, a Daniela, Sami y Lía, tres personas que llegaron en momentos menos pensados, pero me enseñaron que a veces las cosas inesperadas son las que más cambian a tu vida para bien, personas que nunca pensé tenerlas, pero me hace muy feliz que forme parte de mi vida, por su apoyo y cariño hacia mi sin importar cuantas veces falle.

Gadiel Santiago Pahuanquiza Iza

## ÍNDICE DE CONTENIDOS

### ***CAPÍTULO I***

#### ***ANTECEDENTES Y GENERALIDADES..... 1***

1.1 Introducción ..... 1

1.2 Problema de estudio ..... 2

1.2.1 Antecedentes ..... 3

1.2.2 Importancia..... 4

1.2.3 Delimitación (geográfica-espacial) ..... 4

1.3 Justificación..... 6

1.4 Objetivos ..... 7

1.4.1 Objetivo general ..... 7

1.4.2 Objetivos específicos..... 8

1.5 Alcance..... 9

### ***CAPÍTULO II***

#### ***MARCO TEÓRICO..... 10***

2.1 Soporte de Tecnologías de la Información (Soporte TI).....	11
2.2 Chatbots Inteligentes .....	13
2.3 Procesamiento de Lenguaje Natural (NLP).....	15
2.4 Framework Rasa para IA Conversacional.....	17
2.5 Modelos de Aprendizaje Automático en Rasa .....	21
2.6 Arquitectura del Chatbot Inteligente .....	25
2.7 Arquitectura Basada en Microservicios .....	29
2.8 Contenerización y Orquestación (Docker y Kubernetes).....	30
2.9 Integración con GLPI (Gestión de Tickets) .....	42
2.10 Canales de Comunicación: Telegram.....	44
2.11 Base de Conocimiento y Recuperación de Información .....	46
2.12 Monitoreo y Observabilidad.....	48
 <b><i>CAPÍTULO III</i></b>	
<b><i>METODOLOGÍA</i></b> .....	<b><i>51</i></b>
3.1. Recolección de datos .....	51
3.1.1 Entrevistas con el personal técnico .....	51
3.1.2 Encuestas a usuarios del soporte TI .....	52

3.1.3	Análisis documental de tickets de soporte .....	53
3.1.4	Integración de la información recolectada .....	53
3.2	Análisis de la información.....	53
3.2.1	Análisis cualitativo de entrevistas .....	54
3.2.2	Análisis cuantitativo de encuestas.....	54
3.2.3	Análisis de tickets de soporte .....	55
3.2.4	Determinación de flujos conversacionales .....	55
3.3	Diseño del chatbot.....	55

## ***CAPÍTULO IV***

<b><i>RESULTADOS</i></b> .....	<b>56</b>
4.1 Resultados de la implementación del chatbot .....	56
4.1.1 Funcionamiento general del sistema .....	56
.....	57
4.1.2 Atención de requerimientos frecuentes de soporte TI.....	58
4.1.3 Precisión en la interpretación de las consultas .....	59
4.1.4 Impacto en los tiempos de atención .....	61
4.1.5 Percepción de los usuarios sobre el sistema.....	61
4.2 Resultados de la evaluación del desempeño del chatbot.....	62
4.2.1 Métricas de precisión en la interpretación de consultas .....	63
4.2.2 Cobertura funcional del sistema.....	64

4.2.3 Análisis comparativo de tiempos de respuesta.....	64
4.2.4 Evaluación de la satisfacción del usuario.....	65
4.2.5 Síntesis del desempeño del sistema.....	65
4.3 Impacto del chatbot en el soporte TI del Bloque D.....	66
4.4 Análisis global de resultados y discusión.....	67
<b><i>CONCLUSIONES</i></b> .....	<b>68</b>
<b><i>RECOMENDACIONES</i></b> .....	<b>70</b>
<b><i>REFERENCIAS BIBLIOGRÁFICAS</i></b> .....	<b>72</b>
<b><i>Anexos</i></b> .....	<b>76</b>

## ÍNDICE DE TABLAS

<b>Tabla 1</b> Tipos de incidencias frecuentes en el soporte TI .....	12
<b>Tabla 2</b> Comparación entre chatbots basados en reglas y chatbots inteligentes .....	15
<b>Tabla 3</b> Tareas principales del Proceso de lenguaje Natural .....	17
<b>Tabla 4</b> Componentes principales del framework Rasa .....	20
<b>Tabla 5</b> Modelos de aprendizaje automático utilizados en Rasa .....	23
<b>Tabla 6</b> Herramientas y versiones .....	33
<b>Tabla 7</b> Componentes de la Base de Conocimiento .....	47
<b>Tabla 8</b> Métricas de la Base de Conocimiento .....	48

## ÍNDICE DE FIGURAS

<b>Figura 1</b>	Croquis de la ubicación de la Universidad Politécnica Salesiana .....	5
<b>Figura 2</b>	Fotografía del edificio donde se encuentra el Datacenter .....	6
<b>Figura 3</b>	Evolución de los chatbots: de sistemas basados en reglas a chatbots inteligentes ...	14
<b>Figura 4</b>	Flujo general del Procesamiento de lenguaje Natural (NLP) .....	16
<b>Figura 5</b>	Arquitectura general del framework Rasa .....	19
<b>Figura 6</b>	Configuración en config.yml .....	24
<b>Figura 7</b>	Arquitectura general del sistema.....	28
<b>Figura 8</b>	Arquitectura de los microservicios .....	30
<b>Figura 9</b>	Despliegue en Kubernetes .....	32
<b>Figura 10</b>	Dockerfile de Rasa (Custom).....	34
<b>Figura 11</b>	Dockerfile del Action Server .....	35
<b>Figura 12</b>	Docker Compose (fragmento principal) .....	36
<b>Figura 13</b>	Kubernetes Deployment - Rasa .....	37
<b>Figura 14</b>	Kubernetes Service - LoadBalancer .....	38
<b>Figura 15</b>	Kubernetes ConfigMap.....	39
<b>Figura 16</b>	Kubernetes Secrets.....	40
<b>Figura 17</b>	Kubernetes HorizontalPodAutoscaler.....	41
<b>Figura 18</b>	Flujo de Creación de Tickets .....	43

<b>Figura 19</b> Flujo de usuario - Telegram - Chatbot.....	45
<b>Figura 20</b> Flujo de monitoreo.....	50
<b>Figura 21</b> Conversación Chatbot - usuario.....	57
<b>Figura 22</b> Problemas comunes realizados al Chatbot.....	59
<b>Figura 23</b> Métricas .....	60
<b>Figura 24</b> Ingreso de caso donde se puede visualizar el tiempo en resolución.....	61
<b>Figura 25</b> Calificación del usuario después de su respectiva ayuda .....	63
<b>Figura 26</b> Gráficas de los casos en Elastic .....	66

## RESUMEN

El aumento del uso de tecnologías en las instituciones de educación superior ha incrementado la demanda de servicios de soporte técnico. En la Universidad Politécnica Salesiana, esta situación se manifiesta en un elevado número de solicitudes relacionadas con la conectividad, el acceso a sistemas y el uso de plataformas académicas, las cuales son atendidas de manera manual, lo que genera tiempos de respuesta prolongados, saturación del personal técnico y disminución de la percepción de calidad del servicio.

Debido a esta problemática, el objetivo fue desarrollar e implementar un chatbot inteligente para soporte de TI, basado en el framework Rasa e integrado con la plataforma Telegram, que permita automatizar la atención de requerimientos frecuentes y mejorar la experiencia de los usuarios.

La metodología aplicada corresponde a una investigación de tipo aplicado con enfoque mixto. Se utilizaron entrevistas al personal de soporte, encuestas a estudiantes y docentes, y análisis de registros históricos de incidencias para identificar patrones recurrentes. Posteriormente, se diseñó la arquitectura conversacional del chatbot mediante la definición de intents, entidades y flujos de diálogo, y se entrenó el modelo de Procesamiento de Lenguaje Natural con datos contextualizados.

Los resultados obtenidos evidenciaron una reducción considerable a los tiempos de atención de las consultas frecuentes, una disminución de la carga operativa del personal de soporte y una mayor satisfacción de los usuarios. El sistema demostró capacidad para interpretar adecuadamente las solicitudes y ofrecer respuestas coherentes y oportunas.

**Palabras clave:** chatbot inteligente, tecnologías de la información, procesamiento de lenguaje natural, inteligencia artificial aplicada, Rasa framework.

## Abstract

The growing dependence of higher education institutions on information technology has significantly increased the demand for technical support services. At the Salesian Polytechnic University, this situation manifests itself in a high number of requests related to connectivity, access to systems, and use of academic platforms, which are handled manually, resulting in long response times, overburdened technical staff, and a decline in the perception of service quality.

Given this problem, the objective of the research was to develop and implement an intelligent chatbot for IT support, based on the Rasa framework and integrated with the Telegram platform, which would automate the handling of frequent requests and improve the user experience.

The methodology applied corresponds to applied research with a mixed approach. Interviews with support staff, surveys of students and teachers, and analysis of historical incident records were used to identify recurring patterns. Subsequently, the chatbot's conversational architecture was designed by defining intents, entities, and dialogue flows, training the Natural Language Processing model with contextualized data.

The results obtained showed a significant reduction in response times for frequent queries, a decrease in the operational workload of support staff, and greater user satisfaction. The system demonstrated its ability to accurately interpret requests and provide consistent and timely responses.

**Keywords:** Intelligent chatbot, information technologies, natural language processing, applied artificial intelligence, Rasa framework.

# CAPÍTULO I

## ANTECEDENTES Y GENERALIDADES

### 1.1 Introducción

Hoy en día, el ámbito de las instituciones de educación superior está claramente marcado por el uso de tipos tecnológicos digitales enfocadas tanto en el desarrollo de las actividades académicas, así como en las de carácter administrativo. Las plataformas de formación virtual, los sistemas de gestión académica, las redes de comunicación, las prestaciones informáticas, se han convertido en elementos básicos para pasar a ser las condiciones de funcionamiento de la Universidad. Esta dependencia de las tecnologías ha suscitado a la vez un elenco de grandes desafíos, los cuales tienen que ver con la gestión efectiva del soporte técnico de las Tecnologías de Información (TI).

El crecimiento del uso de servicios tecnológicos ha provocado un incremento sostenido en la demanda de atención por parte de las áreas de soporte TI. De acuerdo con IBM (2024), la transformación digital exige no solo la adopción de nuevas tecnologías, sino también la implementación de mecanismos que permitan garantizar la continuidad operativa y una atención oportuna a los usuarios finales. En el contexto universitario, esta necesidad se vuelve aún más relevante debido a la diversidad de usuarios y a la criticidad de los servicios tecnológicos para el proceso educativo.

En este escenario, los sistemas con inteligencia artificial, particularmente los chatbots inteligentes apoyados en Procesamiento de Lenguaje Natural (NLP), han demostrado ser herramientas eficaces para automatizar la atención de consultas frecuentes y optimizar los procesos de soporte técnico. Según Gartner (2022), los asistentes conversacionales se han establecido como una solución viable para optimizar la eficacia de la operación y la experiencia de las personas usuarias en las organizaciones.

La presente investigación se orienta hacia el desarrollo de un chatbot inteligente para el soporte TI basado en el framework Rasa e integrado con la plataforma Telegram, aplicado en la Universidad Politécnica Salesiana, Campus Sur, específicamente en el Bloque D. El proyecto busca contribuir a la optimización del servicio de soporte técnico en función de reducir los tiempos de respuesta y mejorando la experiencia del usuario, utilizando tecnologías emergentes.

## **1.2 Problema de estudio**

El crecimiento sostenido del uso de tecnologías en el ámbito universitario ha provocado un aumento considerable en la cantidad de solicitudes dirigidas al área de soporte técnico. Estas solicitudes abarcan desde problemas de acceso a plataformas académicas hasta dificultades de conectividad y uso de software institucional. Cuando este volumen de requerimientos es atendido exclusivamente de manera manual, se generan tiempos de espera prolongados y una utilización ineficiente de los recursos humanos disponibles (Stair & Reynolds, 2018).

Se considera que el sistema de soporte TI del Bloque D del Campus Sur de la Universidad Politécnica Salesiana está expuesto a una alta carga de trabajo por la concentración de actividades académicas relacionadas con carreras tecnológicas. Se considera que el sistema

de soporte TI del Bloque D del Campus Sur de la Universidad Politécnica Salesiana está expuesto a una alta carga de trabajo por la concentración de actividades académicas relacionadas con carreras tecnológicas. La falta de gestión por cuenta de la misma para canalizar preguntas frecuentes obstaculiza excesivamente la atención al soporte técnico encuadrado en la búsqueda de imposibilidades de resolver incidencias, disminuyendo la calidad proporcionada en la atención al cliente formado por la comunidad universitaria.

### **1.2.1 Antecedentes**

En el Campus Sur de la Universidad Politécnica Salesiana, especialmente en el Bloque D, que concentra esta alta actividad académica y que está vinculada a carreras tecnológicas, se da una utilización intensificada de recursos de informática. Desde el inicio, el soporte TI ha estado considerado el hecho de que la atención debía hacerse de forma presencial, mediante envíos de correos y mediante registros manuales de incidencias, modelos que, con el actual volumen de solicitudes, son insuficientes.

Diversas investigaciones han evidenciado que una gran proporción de los requerimientos de soporte técnico son repetitivos y pueden resolverse mediante sistemas automatizados (Sebastián, 2025). Estos antecedentes demuestran que la incorporación de chatbots inteligentes en servicios de soporte no solo es técnicamente viable, sino también beneficiosa para mejorar la eficiencia del servicio.

### **1.2.2 Importancia**

La relevancia del presente trabajo se justifica por la exigencia de reforzar y optimizar el servicio de soporte de las Tecnologías de la Información de la Universidad Politécnica Salesiana, ofreciendo una atención más rápida, puntual, efectiva para el usuario. Por tal motivo, la implementación de un chatbot inteligente se dibuja como una buena alternativa que procura la automatización de la atención a las consultas frecuentes y la reducción de los tiempos de respuesta y, como consecuencia, mejorar la atención del usuario final.

En el campo del ejercicio académico, se resalta el presente proyecto, que sustenta el trabajo aplicado en aspectos referidos a la inteligencia artificial, el procesamiento de lenguaje natural y el desarrollo de software, que contribuye a la formación profesional en Ciencias de la Computación y enseña a crear las bases para futuras extensiones del sistema a partir de las cuales permitir la generación de innovación tecnológica y mejora de los servicios universitarios.

### **1.2.3 Delimitación (geográfica-espacial)**

La investigación tiene como marco a la institución universitaria Politécnica Salesiana de la ciudad de Quito, concretamente a la del Campus Sur. El sistema universitario se centra en el del Bloque D, espacio académico en el que llevan a cabo la realización de las actividades de desarrollo formativos, prácticas intrínsecas a la carrera actualmente vigente de Ciencias de la Computación, y en el que se hace un uso Intensivo de los recursos tecnológicos y de la prestación de los servicios que alberga la estancia.



## Figura 2

*Fotografía del edificio donde se encuentra el Datacenter*



*Nota.* Bloque D, edificio donde se encuentra el Datacenter. Elaborado por: Los Autores

### **1.3 Justificación**

La investigación es importante en áreas como la tecnología, la academia y la institución, ya que las universidades tienen una creciente dependencia con respecto a los servicios de Tecnologías de la Información, en particular, la Universidad Politécnica Salesiana, Campus Sur. El uso intensivo de la digitalización, redes informáticas y sistemas académicos nos lleva a tener un alto requerimiento de soporte de TI, en especial en el Bloque D, que es donde se llevan a cabo las actividades formativas vinculadas a carreras del tipo tecnológico.

Desde el ámbito institucional, la gestión de las necesidades manuales en requerimientos de soporte TI provoca unos tiempos de respuesta largos y una alta carga para el personal técnico, lo que supone una disminución de la eficiencia del servicio en el usuario final, y por lo tanto, la implementación de un chatbot inteligente es una alternativa para gestionar el soporte técnico, para automatizar la atención de consultas reiteradas cumpliendo además la función de liberar recursos humanos para la atención de problemas más complejos.

Desde el enfoque académico, la investigación contribuye al fortalecimiento de la formación profesional en Computación mediante la articulación del conocimiento teórico con el desarrollo de una solución real ante una problemática concreta, así mismo, el proyecto propuesto establece una base sólida para la investigación futura e incluso las posibles extensiones del sistema hacia otros bloques o campus de la universidad, de forma tal que permite la innovación tecnológica y la mejora continua de los servicios universitarios.

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

Implementar un Chatbot inteligente basado en Rasa con técnicas de NLP, que brinde soporte técnico automatizado al Data Center de la carrera de Computación de la UPS, con el fin de optimizar la atención de requerimientos tecnológicos y mejorar la atención del usuario.

### **1.4.2 Objetivos específicos**

Indagar las problemáticas frecuentes en los servicios de soporte TI, mediante entrevistas, encuestas y revisión de tickets técnicos, con el fin de identificar los requerimientos más comunes y repetitivos.

Analizar los flujos conversacionales típicos en atención técnica y determinar los puntos críticos que pueden ser automatizados mediante un sistema de chatbot.

Definir la arquitectura del chatbot, la organización de los intents, entidades, acciones y base de conocimiento, de acuerdo con los requerimientos funcionales identificados.

Implementar el modelo de procesamiento de lenguaje natural (NLP) utilizando Rasa para poder interpretar y manejar correctamente las preguntas de los usuarios.

Realizar pruebas funcionales, de usabilidad y de rendimiento para el chatbot inteligente basado en Rasa, con el fin de evaluar su capacidad para brindar soporte técnico automatizado, asegurando la precisión en la atención de requerimientos tecnológicos y la satisfacción del usuario final.

## **1.5 Alcance**

La limitación del presente trabajo se ciñe a la elaboración, desarrollo e implementación de un chatbot inteligente para el soporte técnico de las Tecnologías de la Información (TI) en la Universidad Politécnica Salesiana, Campus Sur, en el Bloque D. En este sentido, este proyecto contempla la automatización de la atención a las constantes y recurrentes consultas realizadas por parte del estudiantado, profesorado y el personal administrativo relacionadas con el uso de los servicios tecnológicos institucionales.

De forma particular, el sistema propuesto está orientado a la resolución de incidencias de baja y mediana complejidad, y comprendiendo problemas de acceso a plataformas académicas, problemas de conectividad a las redes institucionales, la recuperación de credenciales, consultas sobre el uso de software académico y la orientación sobre el uso general de los servicios de TI. Para esto, se implementan las funcionalidades descritas en un chatbot creado sobre el framework Rasa, que a su vez se integra a la plataforma de mensajería Telegram que se fijará como principal canal de interacción y de comunicación con los usuarios.

Mientras que el alcance del proyecto incluye un análisis de los requerimientos actuales del soporte técnico, seguido del diseño de la arquitectura conversacional y del modelo de procesamiento de lenguaje natural, así como, del entrenamiento utilizando datos contextualizados al entorno universitario; también se incorpora una evaluación a través de pruebas funcionales y de usabilidad, en cuanto a indicadores como son el tiempo de respuesta, la precisión en la interpretación de las consultas y la satisfacción de los usuarios.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

El presente capítulo elabora los fundamentos teóricos y conceptuales que sustentan el diseño e implementación del chatbot inteligente para soporte de Tecnologías de la Información (TI) apoyado en Rasa y Telegram, introduciendo el procesamiento de lenguaje natural (PLN), así en este sentido, se le prestará atención a los noetos, tecnologías y enfoques más representativos con el soporte TI, los sistemas conversacionales inteligentes y las herramientas que hacen posible su desarrollo e integración a la empresa.

Igualmente se revisan los principios relativos al procesamiento de lenguaje natural y sobre el aprendizaje automático contexto de los sistemas de chatbot, las modernas arquitecturas de microservicios, la contenerización y la orquestación de aplicaciones, en la misma línea, se revisan los sistemas de gestión de tickets, los canales de comunicación y los mecanismos de monitorización, que permiten determinar el funcionamiento y la eficiencia del modelo propuesto.

De este modo, el marco teórico se ofrece como el soporte conceptual preciso para poder comprender las decisiones técnicas que se han ido tomando a lo largo del desarrollo de la presente propuesta, y por tanto, se configura una buena base para poder realizar el desarrollo metodológico, la posterior implementación del chatbot inteligente de soporte TI, el cual se dirige a la automatización de requerimientos tecnológicos y a la mejora de la atención al usuario final.

## **2.1 Soporte de Tecnologías de la Información (Soporte TI)**

El soporte para las Tecnologías de la Información, esto es, soporte TI, constituye el conjunto de actividades que tiene como propósito garantizar el adecuado funcionamiento de los sistemas informáticos, de las aplicaciones y de los servicios tecnológicos en una determinada organización, con lo que la finalidad del soporte TI radica en atender y resolver los incidentes, requerimientos o problemas técnicos que puedan interrumpir la normal operativa y la productividad de sus usuarios (Torres et al., 2025).

En la actualidad, las entidades sociales del siglo XXI normalmente gestionan el soporte técnico TI a través de las mesas de ayuda o help desk, mediante las cuales las demandas de los usuarios son registradas y clasificadas a través de sistemas de tickets, organizando así los incidentes en función de la prioridad, el impacto o el grado de urgencia, lo cual favorece el seguimiento y la trazabilidad de cada uno de los casos; sin embargo, existe una parte relativamente importante de las demandas del soporte TI que son de tipo repetitivo, pudiendo incluir: problemas de acceso a sistemas, caída de la conectividad, fallas de software o incluso consultas de información básica (de Jesús, 2024)

La atención manual de este tipo de requerimientos provoca una importante carga operativa para la plantilla de técnicos y, en consecuencia, incrementa el tiempo de respuesta e incluso puede afectar la calidad del servicio ofrecido; por ello, la automatización del soporte TI ha sido vista como una alternativa que permite mejorar la eficiencia en el servicio, reducir costes operativos y poder ofrecer atención a los usuarios de manera continua (Torres et al., 2025).

En este contexto, el uso de herramientas inteligentes, como son los chatbots basados en procesamiento del lenguaje natural, posibilita la atención automática de consultas frecuentes, la entrega de respuestas inmediatas y la escalabilidad de los casos que realmente requieren intervención humana, de esta manera, el soporte TI evoluciona hacia un modelo más ágil, proactivo y orientado a la satisfacción del usuario final (de Jesús, 2024).

**Tabla 1**

*Tipos de incidencias frecuentes en el soporte TI*

<b>Tipo de incidencia</b>	<b>Descripción</b>
<b>Problemas de acceso</b>	Dificultades para iniciar sesión en sistemas, bloqueo de cuentas o recuperación de contraseñas.
<b>Fallas de software</b>	Errores en aplicaciones, programas que no responden o problemas de instalación y actualización.
<b>Incidencias de red</b>	Problemas de conectividad a internet, red local o acceso a recursos compartidos.
<b>Solicitudes de información</b>	Consultas sobre procedimientos, uso de sistemas o políticas tecnológicas internas.
<b>Problemas de hardware</b>	Fallas en equipos de cómputo, periféricos o dispositivos físicos.
<b>Requerimientos de servicio</b>	Solicitudes de creación de usuarios, permisos, configuraciones o accesos especiales.

*Nota.* Descripción de todas las incidencias encontradas. Elaborado por: Los Autores

## **2.2 Chatbots Inteligentes**

Los chatbots inteligentes son sistemas informáticos que pueden comunicarse con sus usuarios a través del lenguaje natural, con el objetivo de simular una conversación similar a la que se mantiene entre personas. De acuerdo con Torres et al. (2025), estos sistemas de chatbot utilizan técnicas de Inteligencia Artificial y de Procesamiento del Lenguaje Natural para reconocer las preguntas de los usuarios, identificar su intención y generar respuestas acordes con el contexto de la interacción, lo que resulta en mejorar el nivel de comunicación hombre-máquina.

En sus primeras etapas, los chatbots se basaban en reglas predefinidas, donde las respuestas estaban limitadas a patrones específicos y opciones cerradas, lo que reducía considerablemente su capacidad de adaptación ante consultas no previstas. Este enfoque restringía la flexibilidad del sistema y afectaba la experiencia del usuario. Con el avance del aprendizaje automático y de los modelos de lenguaje natural, surgieron en cambio los chatbots inteligentes con los que pueden aprender a partir de los datos y gestionar conversaciones más sofisticadas de manera muy diferente a los sistemas a partir de normas (Garzón-Quiroz et al., 2025).

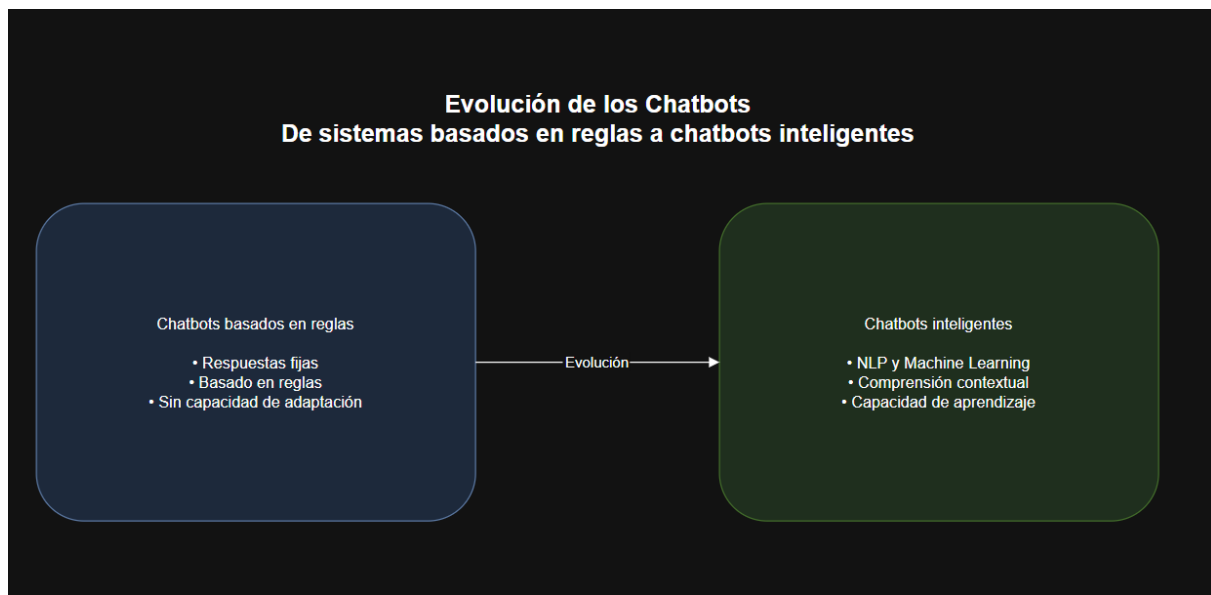
En el ámbito del soporte de las Tecnologías de la Información, los chatbots inteligentes han sido sin duda una de las herramientas clave para automatizar la atención a las consultas frecuentes, la orientación a los usuarios frente a la dificultad de resolución de problemas comunes y la provisión de soporte inmediato sin depender de la disponibilidad del personal técnico. Asimismo, estos sistemas pueden integrarse con plataformas empresariales como

sistemas de tickets y bases de conocimiento, ayudando de este modo a mejorar la atención o eficiencia del servicio (Maruri et al., 2025).

En conjunto, el uso de chatbots inteligentes representa una evolución significativa en la atención al usuario, ya que combinan rapidez, disponibilidad continua y capacidad de aprendizaje. Estas características permiten ofrecer una experiencia de soporte ágil y eficiente, alineada a las demandas actuales de las organizaciones y orientada a incrementar la satisfacción del usuario final (Torres et al., 2025).

### Figura 3

*Evolución de los chatbots: de sistemas basados en reglas a chatbots inteligentes*



*Nota.* Diferencias entre chatbot basado en reglas e inteligentes. Elaborado por: Los Autores

**Tabla 2**

*Comparación entre chatbots basados en reglas y chatbots inteligentes*

<b>Característica</b>	<b>Chatbots basados en reglas</b>	<b>Chatbots inteligentes</b>
<b>Tipo de lógica</b>	Reglas predefinidas	Aprendizaje automático
<b>Comprensión del lenguaje</b>	Limitada a palabras clave	Lenguaje natural (NLP)
<b>Flexibilidad</b>	Baja	Alta
<b>Capacidad de aprendizaje</b>	No aprende	Aprende de interacciones
<b>Manejo de consultas nuevas</b>	Deficiente	Adaptativo
<b>Aplicación en soporte TI</b>	Básica	Avanzada

*Nota.* Diferencias entre chatbot basado en reglas e inteligentes. Elaborado por: Los Autores

### **2.3 Procesamiento de Lenguaje Natural (NLP)**

El Procesamiento de Lenguaje Natural, también conocido como Natural Language Processing (NLP), constituye una de las ramas de la Inteligencia Artificial enfocada en que los sistemas informáticos sean capaces de poder entender, analizar y procesar automáticamente el lenguaje humano. Su principal objetivo o propósito es disminuir la diferencia existente entre la forma como se comunican las personas y la manera en la que las máquinas comprenden/anotan la información para poder proporcionar una interacción más natural y eficiente entre las personas y las computadoras (Torres et al., 2025).

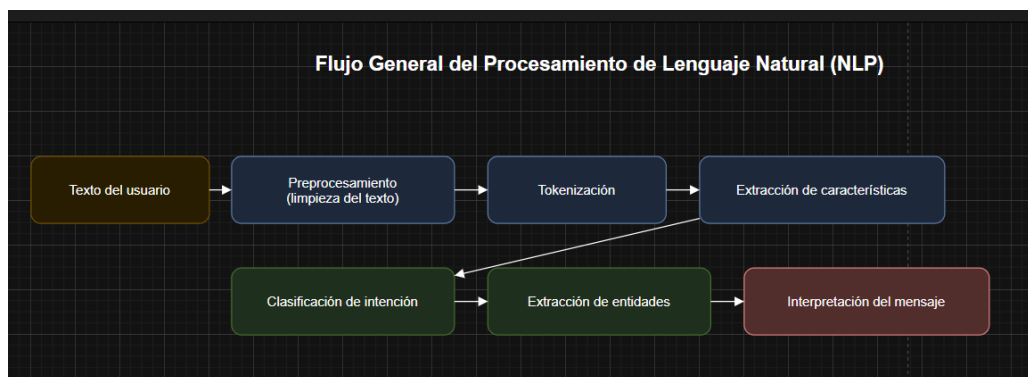
En el marco de los chatbots inteligentes se incorpora el NLP como una parte fundamental, ya que le permite analizar los mensajes introducidos por los usuarios para poder identificar la intención de este y extraer información relevante a partir del texto introducido. Para conseguirlo, el lenguaje natural ha de ser procesado por distintas fases que

transformándose el texto bruto en datos estructurados que pueda procesar el machine learning, cubriendo así las carencias de los sistemas esencialmente basados en reglas de decisión escritas de antemano (Garzón-Quiroz et al., 2025).

En el marco de un sistema de soporte de las Tecnologías de la Información, el NLP permite integrar la automatización de las consultas habituales o la categoría de las peticiones, pero también se puede entender problemas técnicos que pueden ser expresados de otras maneras por parte de los usuarios. Por este motivo, la aplicación de técnicas de NLP implicadas en los chatbots inteligentes puede ayudar a dar respuestas precisas y contextualizadas, beneficiando la eficiencia operativa de los servicios y el usuario final (Maruri et al., 2025).

#### Figura 4

*Flujo general del Procesamiento de lenguaje Natural (NLP)*



*Nota.* Flujo del proceso desde el usuario hasta la recepción del mensaje. Elaborado por: Los Autores

**Tabla 3**

*Tareas principales del Proceso de lenguaje Natural*

<b>Tarea de NLP</b>	<b>Descripción</b>
<b>Preprocesamiento de texto</b>	Limpieza del texto eliminando caracteres especiales, signos innecesarios y normalizando el contenido.
<b>Tokenización</b>	División del texto en partes más pequeñas como palabras.
<b>Lematización</b>	Reducción de las palabras a su forma base o raíz.
<b>Extracción de características</b>	Transformación del texto en representaciones numéricas para su análisis.
<b>Clasificación de intenciones</b>	Identificación del propósito del mensaje del usuario.
<b>Extracción de entidades</b>	Detección de información clave dentro del texto, como nombres, fechas o categorías.
<b>Análisis del contexto</b>	Interpretación del significado considerando el historial de la conversación.

*Nota.* Funcionamiento del NLP. Elaborado por: Los Autores

## **2.4 Framework Rasa para IA Conversacional**

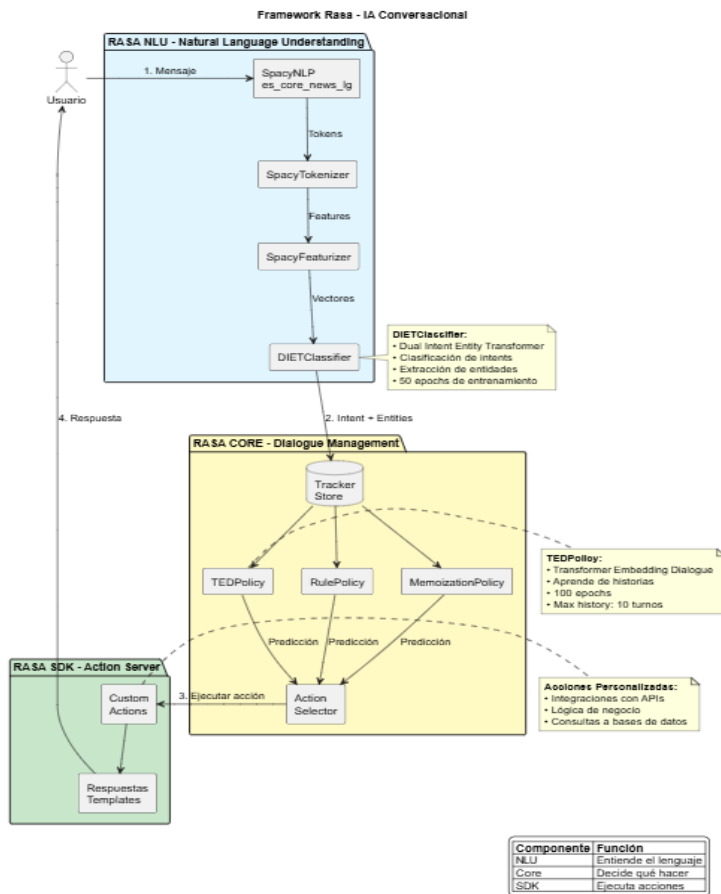
Rasa es un framework de código abierto orientado al desarrollo de asistentes conversacionales inteligentes, el cual permite construir chatbots personalizados mediante el uso de técnicas de procesamiento de lenguaje natural y aprendizaje automático. A diferencia de las plataformas cerradas, Rasa ofrece un control total sobre los datos, los modelos y la lógica conversacional, lo que lo convierte en una alternativa para entornos académicos y

organizacionales donde la privacidad y la capacidad de personalización constituyen aspectos prioritarios (Gujjar & Kumar, 2022).

Rasa se organiza a partir de diversos componentes principales que funcionan de manera integrada con el objetivo de interpretar los mensajes enviados por los usuarios, gestionar la conversación y ejecutar acciones de acuerdo con el contexto de cada interacción. En este sentido, el framework permite entrenar modelos capaces de reconocer intenciones, extraer entidades relevantes y determinar la respuesta más adecuada en cada escenario conversacional, facilitando el desarrollo de sistemas conversacionales aplicados a servicios de información académica y organizacional (Ruindungan & Jacobus, 2021).

**Figura 5**

*Arquitectura general del framework Rasa*



*Nota.* Flujo RASA NLU con RASA CORE y RASA SDK. Elaborado por los Autores.

**Tabla 4**

*Componentes principales del framework Rasa*

Componente	Descripción
<b>Rasa NLU</b>	Interpreta el mensaje de los usuarios, identificando intenciones y entidades mediante modelos de lenguaje.
<b>Rasa Core</b>	Gestiona el flujo de la conversación y decide la siguiente acción basada en el estado del diálogo.
<b>Rasa SDK</b>	Permite definir acciones personalizadas e integraciones externas mediante código en Python.

*Nota.* Tecnologías importantes para el entrenamiento con NLP. Elaborado por: Los Autores

**Figura 6**

*Fragmento de intent (nlu.yml).*

```
nlu:
  # Saludos y despedidas
  - intent: greet
    examples: |
      - hola
      - buenas
      - buen día
      - hey
      - buenos días
      - buenas tardes
      - buenas noches
      - qué tal
      - saludos
      - hola, necesito ayuda
      - /start
      - /restart
      - inicio
      - empezar
      - reiniciar
      - comenzar de nuevo
```

*Nota.* Intenciones que entiende el chatbot. Elaborado por: Los Autores

## Figura 7

*Fragmento de dominio (domain.yml)*

```
responses:  
  utter_greet:  
    - text: "¡Hola! Soy tu asistente de soporte TI. ¿En qué puedo ayudarte hoy?"  
    - text: "¡Buen día! Estoy aquí para ayudarte con cualquier problema técnico. ¿Qué necesitas?"
```

*Nota.* Respuesta del chatbot Elaborado por: Los Autores

Este fragmento ilustra cómo Rasa permite definir de forma estructurada las intenciones del usuario y las respuestas del sistema, lo que facilita el diseño y el mantenimiento del asistente conversacional.

### **2.5 Modelos de Aprendizaje Automático en Rasa**

Rasa emplea diversos modelos de aprendizaje automático con el propósito de interpretar el lenguaje natural y gestionar el flujo del diálogo dentro de los sistemas conversacionales. Según Gujjar y Kumar (2022).

En el componente Rasa NLU, los modelos se utilizan principalmente para la clasificación de intenciones y el reconocimiento de entidades; lo que se puede denominar como modelo de aprendizaje automático más importante es el clasificador DIET (Dual Intent and Entity Transformer), ue emplea una arquitectura basada en transformadores para entrenar de forma conjunta tanto la detección de intenciones y la extracción de entidades. Este enfoque reduce la complejidad del pipeline y mejora el rendimiento del sistema al compartir representaciones internas del lenguaje, lo que ha sido evidenciado en implementaciones prácticas de Rasa en distintos dominios (Ruindungan & Jacobus, 2021).

Estas representaciones permiten capturar el significado contextual del texto, incrementando la capacidad del sistema para manejar variaciones lingüísticas, ambigüedades y diferentes formas de expresión, aspecto fundamental en aplicaciones de atención al usuario y servicios de información (Pradana et al., 2022).

En el componente Rasa Core, los modelos se orientan a la gestión del diálogo. Uno de los más utilizados es la política TED (Transformer Embedding Dialogue Policy), la cual analiza el historial de la conversación junto con el estado actual del diálogo para predecir la siguiente acción que debe ejecutar el sistema. Este modelo se apoya en redes neuronales y técnicas de aprendizaje supervisado, permitiendo que el asistente aprenda patrones conversacionales complejos a partir de historias de entrenamiento y contextos dinámicos (Cesila et al., 2023).

Es posible adaptar los modelos a diferentes dominios, mejorar su desempeño mediante procesos de entrenamiento continuo y mantener un equilibrio entre reglas explícitas y aprendizaje automático, lo que resulta útil en entornos como el soporte técnico y los servicios organizacionales (Gujjar & Kumar, 2022).

**Tabla 5**

*Modelos de aprendizaje automático utilizados en Rasa*

<b>Componente</b>	<b>Tipo</b>	<b>Función</b>	<b>Parámetros clave</b>
<b>DIETClassifier</b>	NLU	Clasificación de intenciones y extracción de entidades mediante aprendizaje profundo.	Epochs: 50, embedding_dim: 256
<b>TEDPolicy</b>	Core	Predicción de la siguiente acción del diálogo usando modelos tipo Transformer.	Epochs: 100, max_history: 10
<b>RulePolicy</b>	Core	Ejecución de reglas fijas definidas en el archivo rules.yml.	core_fallback_threshold: 0.3
<b>MemoizationPolicy</b>	Core	Memorización de conversaciones exactas observadas durante el entrenamiento.	Max_history: 10

*Nota.* Entrenamiento del chatbot. Elaborado por: Los Autores

## Figura 8

*Configuración en config.yml*

```
language: es
pipeline:
  - name: SpacyNLP
    model: "es_core_news_lg"
  - name: SpacyTokenizer
  - name: SpacyFeaturizer
  - name: RegexFeaturizer
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
  - name: DIETClassifier
    epochs: 50
    constrain_similarities: true
    drop_rate: 0.2
    entity_recognition: false
    use_masked_language_model: false
    evaluate_on_num_examples: 50
    evaluate_every_num_epochs: 10
policies:
  - name: RulePolicy
  - name: MemoizationPolicy
  - name: TEDPolicy
    max_history: 10
    epochs: 100
    constrain_similarities: true
    drop_rate: 0.25
    evaluate_on_num_examples: 50
    evaluate_every_num_epochs: 20
```

*Nota.* Archivo de configuración para los modelos de aprendizaje automático. Elaborado por:  
Los Autores

Las configuraciones de los modelos de machine learning que ha de emplear el chatbot construido sobre Rasa estan en el archivo de configuración config.yml, de este modo, las técnicas de procesamiento del lenguaje natural basadas en spaCy se combinan en el pipeline de NLU con técnicas de aprendizaje profundo con el objetivo de que el lenguaje tenga una representación semántica y haya una clasificación de intenciones por medio del clasificador DIET, la configuración también incluye información léxica, sintáctico y sobre n-gramas de modo que se optimiza la capacidad de capturar variaciones lingüísticas del lengua español, además el modelo se entrena utilizando un número de épocas moderado junto con una dimensión del embeddings alto, beneficiando de esta forma un balance entre precisión y capacidad de generalización para el sistema

En lo que respecta a la gestión del diálogo, se utilizan políticas híbridas que combinan reglas determinísticas, aprendidas de registros de conversación y políticas de machine learning por medio de la política TED, de este modo se garantiza que el sistema dé respuestas coherentes frente a situaciones que ya haya aprendido al mismo tiempo que se adapta a contextos de la conversación más complejos manteniendo un comportamiento estable y robusto del chatbot

## **2.6 Arquitectura del Chatbot Inteligente**

La arquitectura del chatbot inteligente fue concebida bajo un enfoque modular y escalable, lo que permite la integración eficiente del procesamiento de lenguaje natural, gestión del diálogo y comunicación con sistemas externos. Este tipo de diseño arquitectónico es ampliamente recomendado en la literatura, ya que contribuye a mejorar la mantenibilidad, la

evolución del sistema y su capacidad de adaptación frente a nuevos requerimientos funcionales (Park et al., 2022).

El chatbot se fundamenta en el framework Rasa, el cual actúa como el núcleo central del sistema. A través de los diferentes medios de comunicación, como plataformas de mensajería, los mensajes enviados por los usuarios son procesados por el módulo de procesamiento de lenguaje natural. Este componente es responsable de transformar el texto en representaciones comprensibles para el sistema mediante la intención del usuario y la extracción de entidades relevantes, tal como se describe en diversas implementaciones basadas en Rasa (Gujjar & Kumar, 2022; Pradana et al., 2022).

Una vez procesada la entrada, la información es gestionada por el componente encargado del control del diálogo, el cual mantiene el estado de la conversación y determina la siguiente acción a ejecutar. Esta toma de decisiones se fundamenta en el historial conversacional, reglas previamente definidas y modelos de aprendizaje automático entrenados, lo que permite al chatbot generar respuestas coherentes, consistentes y contextualizadas, aspecto clave en los sistemas conversacionales modernos (Adamopoulou & Moussiades, 2020).

El servidor de acciones constituye un componente dentro de la arquitectura, ya que posibilita la ejecución de lógica personalizada y la interacción con sistemas externos. Mediante este módulo, el chatbot puede acceder a bases de datos, consultar sistemas de gestión de servicios, recuperar información desde bases de conocimiento y registrar eventos relevantes

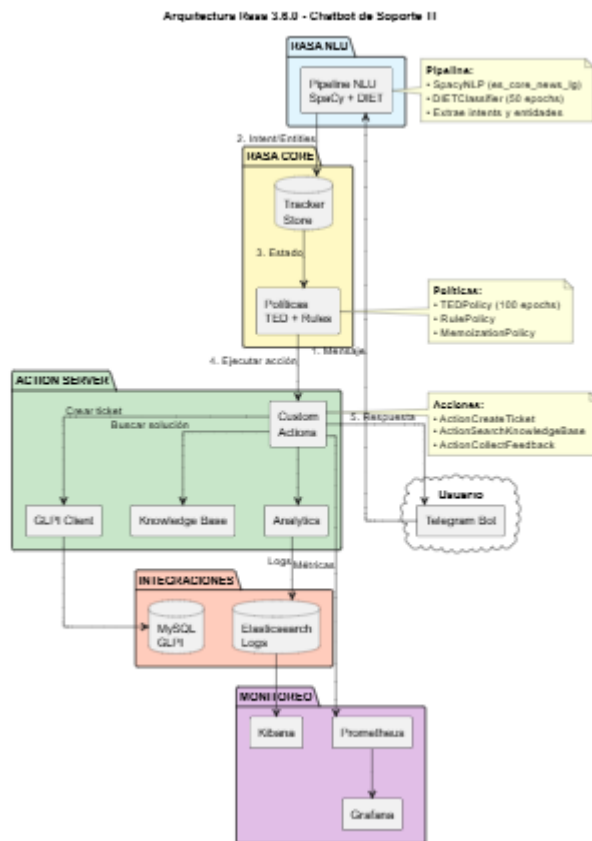
para su análisis posterior. Esta separación entre la lógica conversacional y las acciones externas favorece una arquitectura más limpia, modular y desacoplada, como se evidencia en soluciones avanzadas implementadas con Rasa en distintos dominios (Gujjar & Kumar, 2022; Cesila et al., 2023).

Adicionalmente, la arquitectura contempla mecanismos de persistencia y monitoreo, los cuales permiten almacenar el estado de las conversaciones y recopilar métricas de desempeño. Estos elementos resultan fundamentales para evaluar el comportamiento del chatbot, identificar oportunidades de mejora y garantizar la disponibilidad y confiabilidad del sistema en entornos productivos (Pradana et al., 2022; Adamopoulou & Moussiades, 2020).

En conjunto, la arquitectura propuesta proporciona una base que puede ser válida para el desarrollo de un chatbot inteligente de soporte técnico, integrando técnicas de aprendizaje automático con una estructura flexible que favorezca la escalabilidad, la seguridad y la eficiencia operativa que vayan en línea con las buenas prácticas referenciadas en la literatura (Gujjar & Kumar, 2022; Cesila et al., 2023).

Figura 9

Arquitectura general del chatbot inteligente



Nota. Flujo del chatbot. Elaborado por: Los Autores

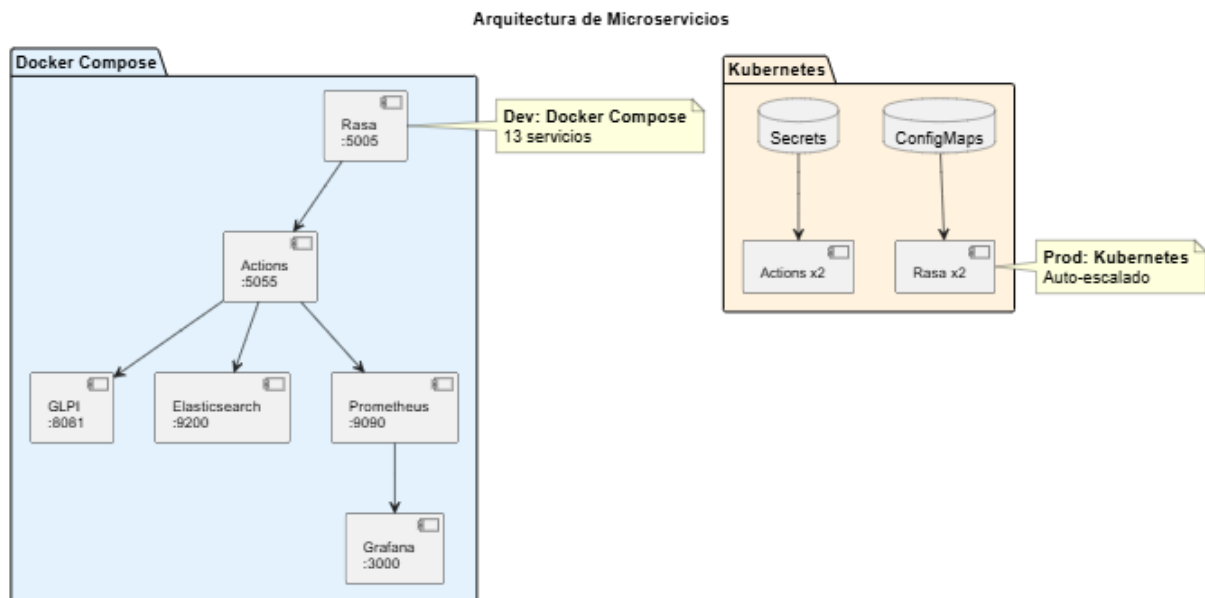
## **2.7 Arquitectura Basada en Microservicios**

La arquitectura microservicio permite diseñar los sistemas software a partir de una división de la aplicación en servicios específicos, pequeños, autónomos e independientes y que, además se comunican entre sí por medio de interfaces y protocolos ligeros (HTTP, mensajes asíncronos ...), brindando flexibilidad y escalabilidad al sistema (Mendez, 2026; Adamopoulou & Moussiades, 2020). La implementación de un chatbot inteligente nos ayuda a aplicar este modelo por medio de separación de diferentes componentes, tales como el procesador de lenguaje natural, el gestor de diálogos, el servidor de acciones y la integración con el sistema/canal externo, permitiendo su despliegue, actualización y el mantenimiento de manera independiente; tal y como se puede observar en distintas implementaciones con Rasa (Ruindungan & Jacobus, 2021; Gujjar & Kumar, 2022).

Una característica importante de esta arquitectura consiste en la escalabilidad de los servicios de forma individual, en caso de caídas, ya que si un microservicio se detiene, el sistema no tiene por qué detenerse, lo cual es importante en entornos donde se producen exigencias de alta carga de dialogo; también permite incorporar unas tecnologías heterogéneas y el despliegue y monitoreo automático, contribuyendo a la creación de sistemas conversacionales más robustos y acordes a como se requiere para el soporte técnico (Gartner, 2022; Torres et al., 2025).

## Figura 10

### Arquitectura de los microservicios



*Nota.* Conexión entre contenedores y orquestador. Elaborado por: Los Autores

## 2.8 Contenerización y Orquestación (Docker y Kubernetes)

La contenerización es un método que hace posible envolver aplicaciones con todas sus dependencias en un todo aislado denominado contenedor que puede ejecutarse en entornos diferentes y variados. Dentro de este contexto, ha ido surgiendo Docker como una de las plataformas más utilizadas, gracias a su sencillez y facilidad en la portabilidad, despliegue y

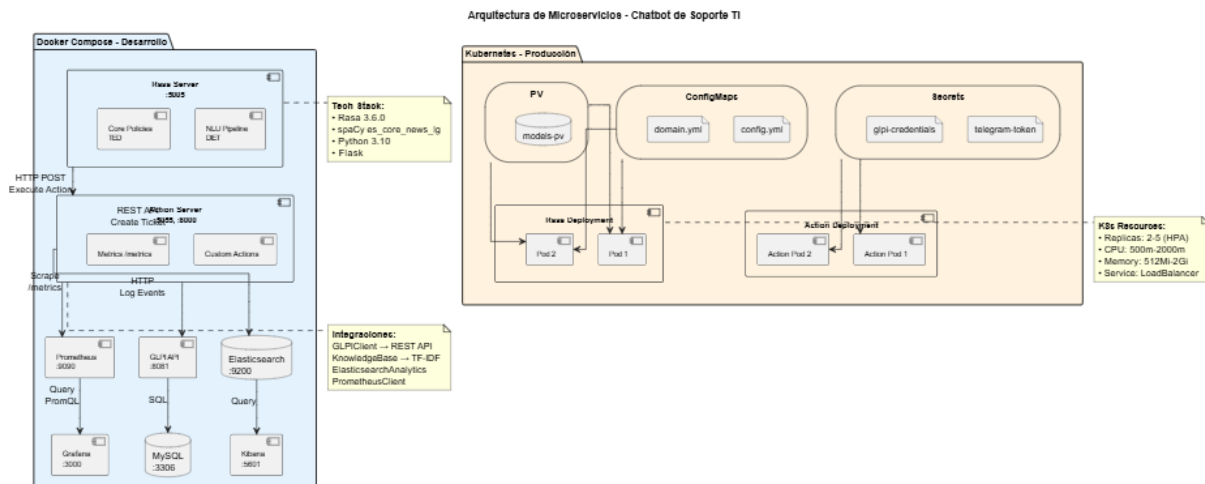
replicación de aplicaciones. En el desarrollo de un chatbot inteligente, la contenerización posibilita encapsular componentes como el servidor de Rasa, el servidor de acciones y los servicios de integración en contenedores independientes, reduciendo problemas de compatibilidad entre los espacios de desarrollo, prueba y producción, y simplificando la vida útil del software (Chen et al., 2021).

La combinación de Docker y Kubernetes facilita la implementación de arquitecturas basadas en microservicios, permitiendo escalar componentes bajo demanda, reiniciar contenedores ante fallos y mejorar la observabilidad del sistema mediante mecanismos de monitoreo y registro de eventos (Chen et al., 2021; Vasireddy et al., 2023).

En conjunto, la integración de contenerización y orquestación constituye un pilar fundamental para el despliegue moderno de sistemas inteligentes, ya que aporta eficiencia operativa, resiliencia y capacidad de escalado, asegurando un funcionamiento estable y confiable del chatbot en entornos productivos (Vasireddy et al., 2023).

## Figura 11

### Despliegue en Kubernetes



Nota. Despliegue con Docker Compose y Kubernetes. Elaborado por: Los Autores

**Tabla 6***Herramientas y versiones*

<b>Herramienta</b>	<b>Versión</b>	<b>Propósito</b>
<b>Docker</b>	28.4.0	Motor de contenedores
<b>Docker Compose</b>	2.	Orquestación multi-contenedor local
<b>Kubernetes</b>	1.34.0	Orquestación en producción
<b>Minikube</b>	1.37.0	Cluster Kubernetes local
<b>kubectl</b>	1.34.0	CLI de Kubernetes
<b>Rasa Image</b>	3.6.0	Contenedor de Rasa personalizado
<b>Action Server Image</b>	latest	Contenedor de acciones personalizado
<b>MySQL</b>	8.0	Base de datos
<b>Elasticsearch</b>	8.11.0	Motor de búsqueda
<b>Prometheus</b>	latest	Recolección de métricas
<b>Grafana</b>	latest	Visualización de métricas

*Nota.* Requerimiento funcional de salidas. Elaborado por: Los Autores

## Figura 12

*Dockerfile de Rasa (Custom)*

```
FROM rasa/rasa:3.6.0

USER root

RUN pip install --no-cache-dir \
    'spacy>=3.4,<3.6' \
    && python -m spacy download es_core_news_lg \
    && python -m spacy link es_core_news_lg es

USER 1001

WORKDIR /app
```

*Nota.* Archivo de creación del contenedor Rasa. Elaborado por: Los Autores

## Figura 13

### *Dockerfile del Action Server*

```
FROM rasa/rasa-sdk:3.2.2

# Instalar dependencias del sistema para python-ldap
USER root
RUN apt-get update && apt-get install -y \
    gcc \
    libldap2-dev \
    libsasl2-dev \
    && rm -rf /var/lib/apt/lists/*

# Copiar requirements
COPY requirements.txt /tmp/requirements.txt

# Instalar dependencias de Python como root
RUN pip install --no-cache-dir -r /tmp/requirements.txt

# Volver al usuario rasa
USER 1001

# El resto de la configuración viene del docker-compose
WORKDIR /app
```

*Nota.* Requerimiento funcional. Elaborado por: Los Autores

## Figura 14

*Docker Compose (fragmento principal)*

```
services:
  > Run Service
  rasa:
    # Imagen personalizada con spaCy y es_core_news_lg
    build:
      context: ./rasa_custom
      dockerfile: Dockerfile
    restart: always
    env_file:
      - .env
    ports:
      - "5007:5005"
    volumes:
      - ./models:/app/models
      - ./data:/app/data
      - ./actions:/app/actions
      - ./config:/app/config
      - ./config.yml:/app/config.yml:ro
      - ./domain.yml:/app/domain.yml:ro
      - ./credentials.yml:/app/credentials.yml:ro
      - ./endpoints.yml:/app/endpoints.yml:ro
    command: run --enable-api --cors "*" --debug

  > Run Service
  interceptor:
    image: python:3.10-slim
    restart: always
    env_file:
      - .env
    ports:
      - "5005:5005" # Este será el puerto que ngrok ve
    volumes:
      - ./interceptor.py:/app/interceptor.py
      - ./proxy_requirements.txt:/app/requirements.txt
    working_dir: /app
```

*Nota.* Levantamiento de los contenedores. Elaborado por: Los Autores

## Figura 15

### *Kubernetes Deployment - Rasa*

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rasa-deployment
  labels:
    app: rasa
spec:
  replicas: 2 # Puedes aumentar según la carga
  selector:
    matchLabels:
      app: rasa
  template:
    metadata:
      labels:
        app: rasa
    spec:
      containers:
        - name: rasa
          image: rasa-chatbot:latest
          imagePullPolicy: Never
          command: ["rasa"]
          args: ["run", "--enable-api", "--cors", "*", "--debug"]
          ports:
            - containerPort: 5005
              name: http
          env:
            - name: RASA_MODEL
              value: "/app/models"
            - name: RASA_ACTION_ENDPOINT
              value: "http://action-server-service:5055/webhook"
          volumeMounts:
            - name: config-volume
              mountPath: /app/config.yml
              subPath: config.yml
            - name: domain-volume
              mountPath: /app/domain.yml
              subPath: domain.yml
```

*Nota.* Archivo del orquestador de contenedores. Elaborado por: Los Autores

## Figura 16

### *Kubernetes Service - LoadBalancer*

```
apiVersion: v1
kind: Service
metadata:
  name: rasa-service
spec:
  selector:
    app: rasa
  ports:
  - protocol: TCP
    port: 5005
    targetPort: 5005
  type: LoadBalancer

---
apiVersion: v1
kind: Service
metadata:
  name: action-server-service
spec:
  selector:
    app: action-server
  ports:
  - protocol: TCP
    port: 5055
    targetPort: 5055
  type: ClusterIP
```

*Nota.* Balanceador de cada uno de los servicios implementados. Elaborado por: Los Autores

## Figura 17

### *Kubernetes ConfigMap*

```
# ConfigMap para config.yml
apiVersion: v1
kind: ConfigMap
metadata:
  name: rasa-config
data:
  config.yml: |
    # Se copiará el contenido de tu config.yml aquí
    # kubectl create configmap rasa-config --from-file=config.yml

---
# ConfigMap para domain.yml
apiVersion: v1
kind: ConfigMap
metadata:
  name: rasa-domain
data:
  domain.yml: |
    # Se copiará el contenido de tu domain.yml aquí
    # kubectl create configmap rasa-domain --from-file=domain.yml
```

*Nota.* Archivo para guardar datos no sencibles. Elaborado por: Los Autores

## Figura 18

### *Kubernetes Secrets*

```
apiVersion: v1
kind: Secret
metadata:
  name: rasa-credentials
type: Opaque
stringData:
  credentials.yml: |
    telegram:
      access_token: "8415781433:AAHLg54YTZcK8EEOL-x38fcXsbzdYiktXIo"
      verify: "Soporteupsbot"
      webhook_url: "https://samella-overfunctioning-overserenely.ngrok-free.dev/webhooks/telegram/v1"

---
# Secret para Action Server (GLPI, LDAP)
apiVersion: v1
kind: Secret
metadata:
  name: action-server-secrets
type: Opaque
stringData:
  glpi-url: "http://glpi/apirest.php"
  glpi-app-token: "b255497de9998f9c4322a9304cac0377"
  glpi-user-token: "B5i7nrCuVEIMq8Ta6nvt0pIAQAq7y35Gr2Zaqmn"
```

*Nota.* Datos sensibles. Elaborado por: Los Autores

## Figura 19

### *Kubernetes HorizontalPodAutoscaler*

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: rasa-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: rasa-deployment
  minReplicas: 2
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 70 # Escala si CPU > 70%
  - type: Resource
    resource:
      name: memory
      target:
        type: Utilization
        averageUtilization: 80 # Escala si Memoria > 80%
  behavior:
    scaleDown:
      stabilizationWindowSeconds: 300 # Espera 5 min antes de reducir
      policies:
      - type: Percent
        value: 50 # Reduce máximo 50% de pods por vez
        periodSeconds: 60
    scaleUp:
      stabilizationWindowSeconds: 0 # Escala inmediatamente
      policies:
      - type: Percent
        value: 100 # Duplica pods si es necesario
        periodSeconds: 15
```

*Nota.* Permite escalar a futuro de manera horizontal. Elaborado por: Los Autores

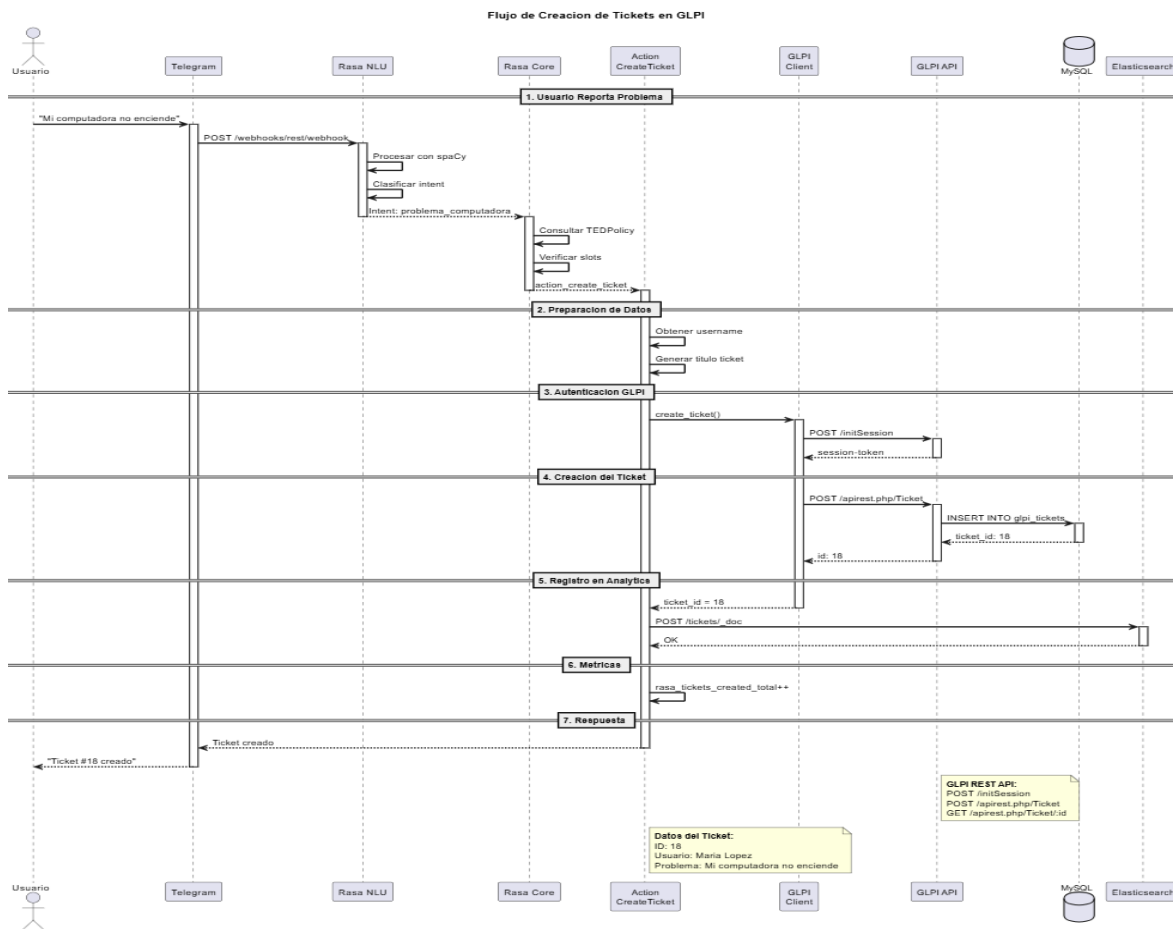
## **2.9 Integración con GLPI (Gestión de Tickets)**

El objetivo de integrar el chatbot inteligente con GLPI es automatizar y optimizar la gestión de tickets de soporte técnico. GLPI se consolida como una herramienta para gestionar servicios de TI, ya que permite el registro, clasificación, asignación y seguimiento de incidentes y solicitudes, favoreciendo una administración eficiente de los implementos de soporte disponibles (Marin, 2023).

El chatbot actúa como punto de entrada para los usuarios, recopilando datos relevantes como la descripción del problema, la categoría del incidente y la información básica del solicitante, los cuales son enviados a GLPI para su registro formal. Esta integración mejora la calidad y trazabilidad de los incidentes, reduce la intervención manual y optimiza los tiempos de respuesta del soporte técnico. Además, el chatbot puede proporcionar retroalimentación automática al usuario, como el número de ticket y su estado actual, contribuyendo a una mayor transparencia del proceso y a una mejor experiencia del usuario (Marin, 2023).

**Figura 20**

*Flujo de Creación de Tickets*



*Nota.* Sistema de gestión de tickets. Elaborado por: Los Autores

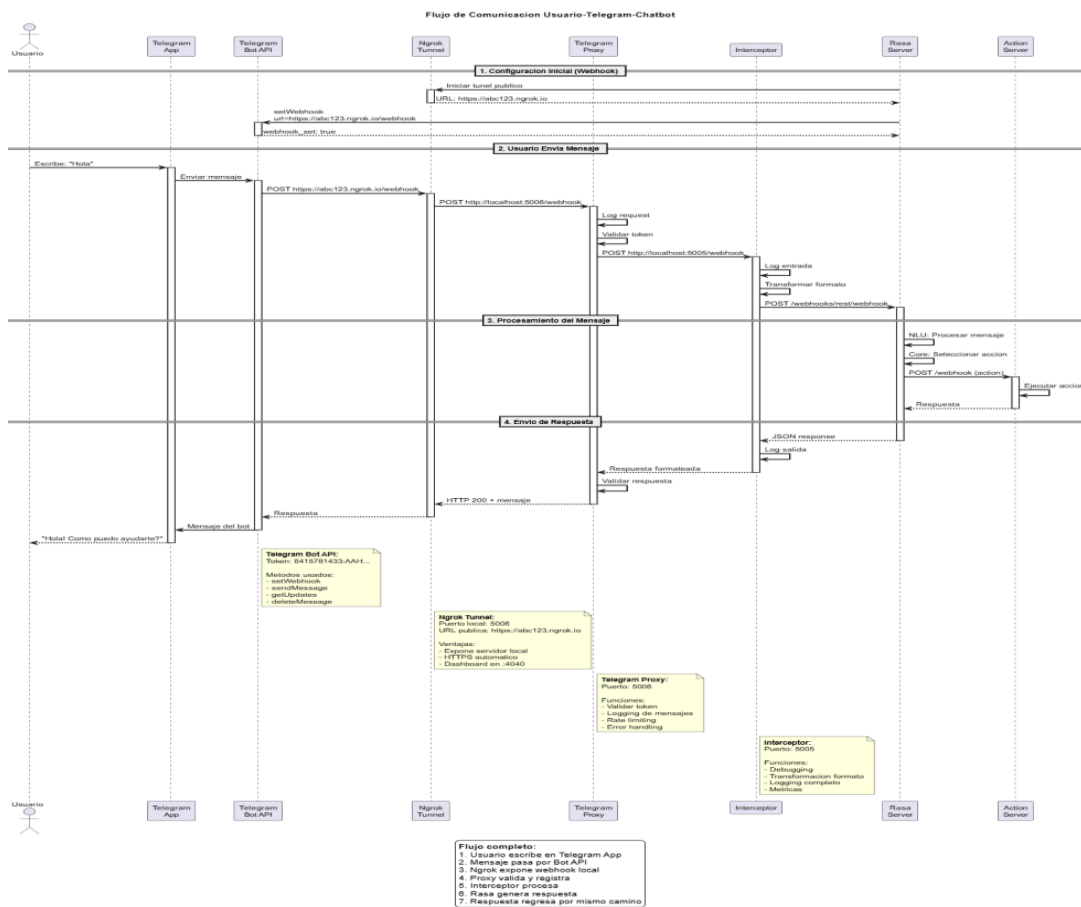
### **2.10 Canales de Comunicación: Telegram**

De esta forma, los mensajes del usuario son procesados por el procesamiento del lenguaje natural y el diálogo antes de generar una respuesta adecuada, asegurando coherencia y contexto en la interacción (Gunawan et al., 2021).

El uso de Telegram favorece la separación entre la lógica interna del chatbot y la interfaz de usuario, permitiendo extender el sistema a otros canales de mensajería en el futuro sin modificar su funcionalidad principal, lo que promueve la reutilización de componentes y la escalabilidad del sistema (Khaund et al., 2020).

**Figura 21**

*Flujo de usuario – Telegram - Chatbot*



*Nota.* Flujo de interacción del usuario con telegram. Elaborado por: Los Autores

### **2.11 Base de Conocimiento y Recuperación de Información**

Esta base reúne documentos validados, preguntas frecuentes, procedimientos y soluciones, lo que permite al asistente proporcionar respuestas consistentes y alineadas con las políticas y directrices de la organización, aspecto clave en la administración de los servicios de TI y la automatización del soporte al usuario (Ltd, 2025; Adamopoulou & Moussiades, 2020).

Estos enfoques contribuyen a mejorar las respuestas del chatbot, reducir la carga del personal de soporte técnico y disminuir los tiempos de respuesta, tal como se evidencia en estudios sobre eficiencia comunicativa en sistemas conversacionales (Garzón-Quiroz et al., 2025).

Este proceso de mejora continua fortalece la capacidad del chatbot para ampliar progresivamente el alcance de las respuestas automatizadas y adaptarse a las necesidades recurrentes de la organización, contribuyendo a una mayor eficiencia operativa y a una experiencia de usuario más satisfactoria (Torres et al., 2025; Maruri et al., 2025; Gartner, 2022).

**Tabla 7***Componentes de la Base de Conocimiento*

<b>Componente</b>	<b>Tecnología</b>	<b>Descripción</b>
<b>Índice</b>	JSON	Metadatos, títulos, rutas y vectores TF-IDF
<b>Motor de búsqueda</b>	TF-IDF + Scikit-learn	Algoritmo de recuperación de información
<b>Vectorizador</b>	TfidfVectorizer	Transforma texto a vectores numéricos
<b>Similitud</b>	Cosine Similarity	Mide relevancia entre consulta y documentos
<b>Threshold</b>	0.5	Umbral mínimo de similitud para retornar resultado
<b>Logging</b>	Elasticsearch	Registro de búsquedas y resultados

<b>Componente</b>	<b>Tecnología</b>	<b>Descripción</b>
<b>Acción Rasa</b>	ActionSearchKnowledgeBase	Ejecuta búsqueda desde el chatbot

*Nota.* Base de conocimiento. Elaborado por: Los Autores

## **Tabla 8**

*Métricas de la Base de Conocimiento*

<b>Métrica</b>	<b>Valor</b>	<b>Descripción</b>
<b>Total de artículos</b>	2	Artículos disponibles en la KB
<b>Búsquedas realizadas</b>	21	Total de consultas procesadas
<b>Tasa de éxito</b>	85%	Búsquedas que retornaron resultados (similitud > 0.5)
<b>Promedio de similitud</b>	0.72	Similitud promedio de resultados encontrados
<b>Tiempo de búsqueda</b>	< 50ms	Tiempo promedio de procesamiento

*Nota.* Métricas en base al conocimiento pasado al chatbot. Elaborado por: Los Autores

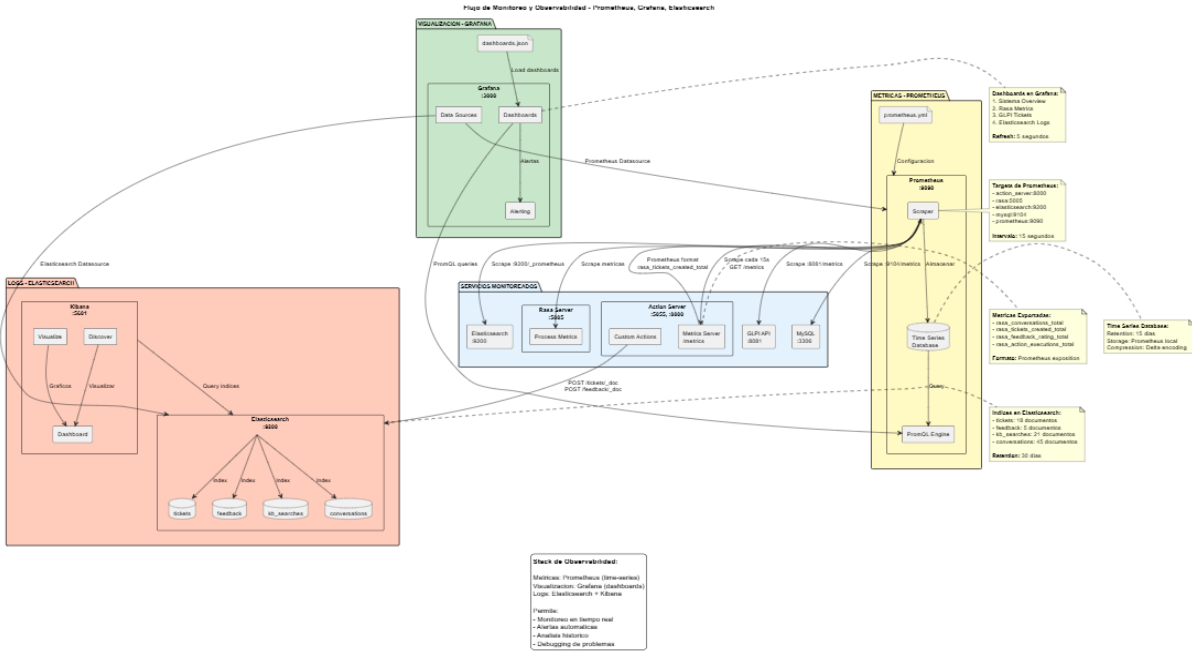
### **2.12 Monitoreo y Observabilidad**

La observabilidad, por otro lado, amplía el alcance del monitoreo al permitirnos comprender el funcionamiento interno del sistema a través del estudio de registros, métricas y eventos. Este método permite la identificación de cuellos de botella, la evaluación de los modelos de aprendizaje automático y el análisis de patrones de uso del chatbot , lo cual es

particularmente útil en configuraciones en contenedores y arquitecturas basadas en microservicios. La integración de herramientas de monitoreo y observabilidad mejora la estabilidad, confiabilidad y escalabilidad del chatbot inteligente, maximiza la utilización de recursos y ayuda a brindar un servicio de soporte más efectivo y consistente que es consistente con las mejores prácticas en la gestión de los servicios de TI y el uso estratégico de chatbots en atención al cliente (Garzón-Quiroz et al., 2025).

# Figura 22

## Flujo de monitoreo



Nota. Sistema de monitoreo del chatbot. Elaborado por: Los Autores

## **CAPÍTULO III**

### **METODOLOGÍA**

#### **3.1. Recolección de datos**

La etapa de la recolección de información fue el corazón del trabajo, ya que nos permitió acceder a insumos que estaban directamente relacionados con la realidad del soporte técnico de las Tecnologías de la Información en la Universidad Politécnica Salesiana, Campus Sur, especialmente en el Bloque D. Por tanto, el proceso de levantamiento de información intentaba hacer transparentes situaciones reales del contexto universitario para que el diseño y el funcionamiento del chatbot respondiese coherentemente a la dinámica existente en el servicio de soporte TI.

Para perseguir este objetivo, se decidió adoptar un enfoque de método mixto que articulara métodos cualitativos y cuantitativos. Con esta estrategia permitió, por un lado, obtener un conocimiento de la experiencia del personal técnico responsable de la gestión de incidentes y, por otro lado, analizar datos cuantitativos estructurados de encuestas y registros históricos de entradas. Así que se adoptó una perspectiva más integradora del problema, que articularía las percepciones y los patrones medibles de las consultas de apoyo.

##### **3.1.1 Entrevistas con el personal técnico**

Se realizaron entrevistas con el personal técnico responsable del soporte de TI en el Bloque D del Campus Sur, que tiene experiencia en la gestión de incidentes en el día a día y un

conocimiento profundo de los tiempos de respuesta promedio y los problemas reportados con más frecuencia por los usuarios de la comunidad. La aplicación de la entrevista como técnica permitió información sobre los procedimientos establecidos y sobre las fortalezas y debilidades del modelo de atención basado principalmente en la intervención manual.

La técnica empleada fue la de entrevista semiestructurada, pues permitía contar con una guía temática clara al tiempo que no se limitaba la posibilidad de profundizar sobre aspectos relevantes que aparecían en la conversación. La selección de la muestra fue por muestra intencionada en la que se priorizaron a los técnicos con entrevistas de experiencia en las áreas de redes, plataformas académicas y software institucional.

El análisis de la información permitió identificar los problemas estructurales, entre ellos la consulta recurrente de las mismas solicitudes, la ausencia de herramientas automáticas de priorización y la dependencia del trabajo humano en la resolución de incidencias de baja complejidad.

### **3.1.2 Encuestas a usuarios del soporte TI**

Con el fin de complementar la información cualitativa obtenida, se aplicaron encuestas dirigidas a estudiantes, docentes y el resto del personal que utilizan los servicios tecnológicos del Bloque D. Esta técnica tuvo como propósito principal reconocer la percepción de los usuarios sobre el servicio de soporte TI, así como identificar los problemas más frecuentes y los medios empleados para solicitar asistencia.

### **3.1.3 Análisis documental de tickets de soporte**

El análisis documental se concretó a partir de los registros históricos de tickets de soporte generados en el Bloque D. Esta técnica propició el acceso a información objetiva y estructurada de las incidencias generadas, considerando variables como el tipo de problema, el área afectada, los tiempos de resolución y la frecuencia con la cual apareció algunos requisitos. El análisis de los tickets se centró en la identificación de patrones recurrentes y la clasificación de las incidencias en función de su nivel de complejidad. La información obtenida de este análisis permitió distinguir las solicitudes que son susceptibles de ser resueltas mediante la automatización de aquellas que requieren obligatoriamente la intervención directa del personal técnico.

### **3.1.4 Integración de la información recolectada**

La información recolectada de las entrevistas, encuestas y del análisis de la documentación fueron estructurados y sistematizados de forma organizada, lo que permitió que posteriormente se pudiera proceder al análisis. El proceso de integración ha propiciado una visión completa del problema, interrelacionando la experiencia del técnico personal, la percepción de los usuarios, y los datos históricos de incidencias.

## **3.2 Análisis de la información**

Aquí es donde se interpreta, de forma sistemática, la información recibida a partir de las entrevistas personales, las encuestas y el examen de la información documental, con la vista

puesta en la identificación de patrones, la aparición continua de problemas o el descubrimiento de oportunidades para la automatización del soporte TI del Bloque D.

### **3.2.1 Análisis cualitativo de entrevistas**

La información obtenida fue procesada a partir de un proceso de categorización a partir de los temas idénticos obtenidos en las respuestas. En primer lugar, se realizó la revisión de las respuestas, con el fin de detectar las ideas comunes relacionadas con los tipos de incidencias, los flujos de atención, los tiempos de respuesta y las principales dificultades operativas. Posteriormente, las ideas fueron clasificados por categorías, tales como consultas recurrentes, incidencias graves, restricciones de recursos, o fortalezas para la automatización.

### **3.2.2 Análisis cuantitativo de encuestas**

Los datos obtenidos a partir de las encuestas se procesaron mediante técnicas de estadística descriptiva, lo que permitió calcular frecuencias, porcentajes y promedios de las variables analizadas. Este procedimiento facilitó la identificación de los problemas más frecuentes y la percepción general de los usuarios ante el servicio de soporte TI.

Los resultados reflejaron una alta recurrencia de solicitudes relacionadas con el acceso a plataformas académicas, la conectividad a redes institucionales y el uso de software académico. Además, se evidenció un nivel de insatisfacción con los tiempos de respuesta, así como una actitud favorable a la implementación de un sistema automatizado de atención en un chatbot.

### **3.2.3 Análisis de tickets de soporte**

El procesamiento de los datos se lleva a cabo de acuerdo con las técnicas de estadística descriptiva, que proporcionan la posibilidad de calcular frecuencias, porcentajes y promedios de las variables que fueron analizadas. Este procedimiento permitió identificar los problemas más solicitados y, en general, la idea que tienen los usuarios del servicio de soporte TI.

### **3.2.4 Determinación de flujos conversacionales**

A partir del análisis integral de la información recogida en el proceso levantado por las encuestas, se definieron los principales flujos conversacionales del mismo chatbot y las necesidades de automatización. En efecto, el análisis permitió conocer los tipos de consultas que pueden resolverse de manera automática y cuáles deben ser atendidas por el soporte humano.

## **3.3 Diseño del chatbot**

Una de las etapas significativas de la metodología es el diseño del chatbot, ya que en esta fase se materializan las conclusiones a las que se llegó desde la recolección de la información hasta su análisis. A través del reconocimiento de patrones, los problemas recurrentes y las necesidades de automatización, se erigió un modelo conversacional orientado a ofrecer soporte técnico eficiente y contextualizado a los usuarios del Bloque D.

## **CAPÍTULO IV**

### **RESULTADOS**

La presente sección expone los resultados alcanzados a raíz de la implementación del chatbot inteligente creado con el objetivo de poder proporcionar soporte técnico para las Tecnologías de la Información en el Bloque D del Campus Sur de la Universidad Politécnica Salesiana. Las conclusiones planteadas tienen su origen, por un lado, en las etapas metodológicas llevadas a cabo y presentadas en el momento del Capítulo III, en lo que se refiere a la implementación técnica del sistema y al posterior proceso de evaluación de la misma.

Para ello se tuvo en cuenta tanto los indicadores de desempeño del sistema como la propia apreciación de las personas que intervinieron en la herramienta durante el uso de esta.

#### **4.1 Resultados de la implementación del chatbot**

##### **4.1.1 Funcionamiento general del sistema**

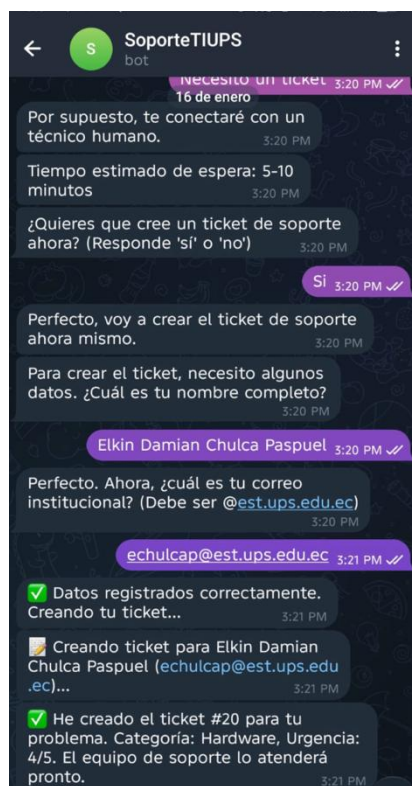
Una vez que fue implementado en la plataforma Telegram, se experimentó la funcionalidad del chatbot como primer canal de atención de requisitos de soporte de tecnologías de la información.

El sistema mostró también la capacidad de iniciar de forma automática la interacción, determinar el motivo de la consulta y dónde guiar al usuario en flujos conversacionales que han sido definidos con anterioridad, de modo que, en aquellos casos en que la información fue

insuficiente, el chatbot pidió información adicional de modo claro y ordenado, de manera que allí no se otorgan respuestas genéricas o ambiguas.

### Figura 23

*Conversación Chatbot - usuario*



*Nota.* Asignación de ticket al a través del telegram. Elaborado por: Los Autores

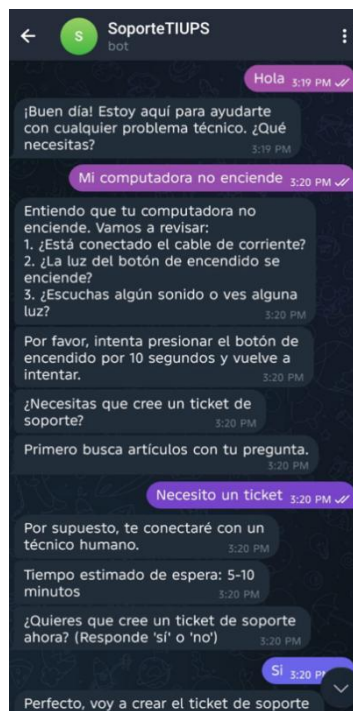
#### **4.1.2 Atención de requerimientos frecuentes de soporte TI**

Uno de los hallazgos más relevantes que se pusieron de manifiesto fue la competencia de un chatbot para atender de manera autónoma los requisitos más frecuentes que se habían decidido en la etapa de análisis. Las cuestiones relacionadas con el acceso a las plataformas académicas, las dificultades relacionadas con el acceso a las redes institucionales, la recuperación de credenciales y la orientación del uso del software académico constituyen una parte de estos.

En estos casos, el sistema respondería instantáneamente con instrucciones claras y secuenciales para ayudar a resolver incidencias habituales. Gran parte de los casos, los usuarios en casi todas las interacciones lograron resolver sus problemas sin tener que contactar directamente con el técnico personal, lo que se interpretó como un indicio de una disminución importante de la carga operativa del área de soporte TI.

## Figura 24

*Problemas comunes realizados al Chatbot.*



*Nota.* Reconocimiento de la intención. Elaborado por: Los Autores

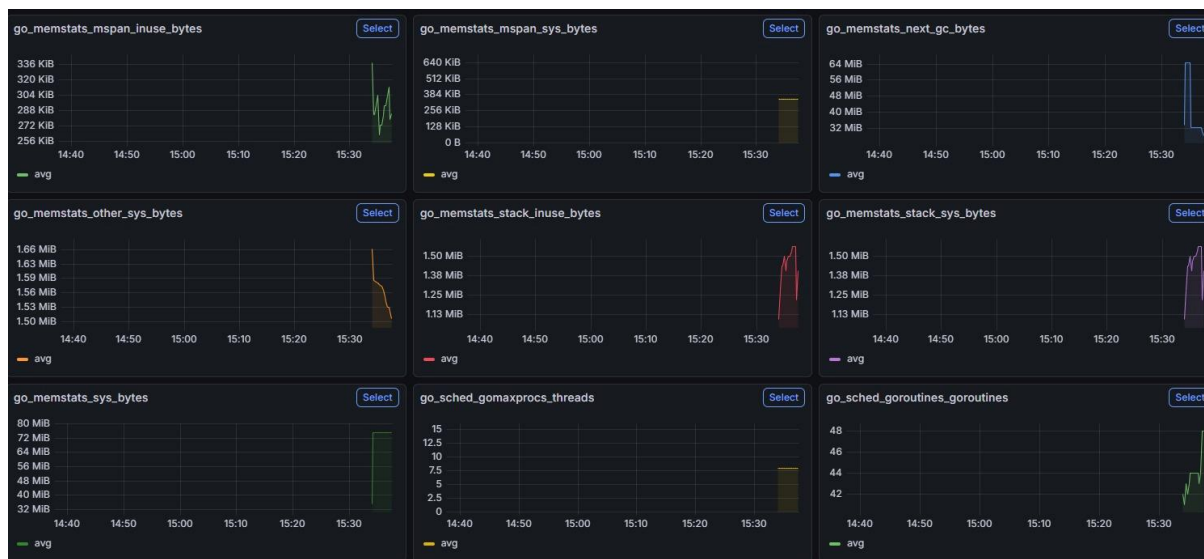
### 4.1.3 Precisión en la interpretación de las consultas

El chatbot adquirió un altísimo estatus en cuanto a la interpretación de las consultas, ya que pudo identificar intenciones y entidades de forma realmente sobresaliente. El empleo de

datos reales para entrenar el modelo fue esencial, ya que permitió al sistema acercarse a los aspectos de la práctica comunicativa universitaria muy nuestros, como son expresiones informales, variaciones lingüísticas y descripciones poco técnicas. La de pruebas fue de un número reducido de interpretaciones incorrectas.

## Figura 25

### Métricas



*Nota.* Sistema de monitoreo del chatbot a través de las métricas. Elaborado por: Los Autores

#### 4.1.4 Impacto en los tiempos de atención

Otro resultado significativo fue la drástica reducción de los tiempos de atención en las consultas de soporte TI. Dado que el modelo tradicional dependía de la disponibilidad del personal técnico, en el caso del chatbot, se podía hacer atención continua e ininterrumpida, sin tiempos de espera.

Esta disponibilidad irrestricta fue considerada una ventaja clara para los usuarios, especialmente fuera del horario académico habitual. Por lo tanto, la reducción de los tiempos de respuesta mejoró la percepción del servicio de soporte de TI y robustez la confianza en las herramientas tecnológicas institucionales.

#### Figura 26

*Ingreso de caso donde se puede visualizar el tiempo en resolución*

ID	TÍTULO	ESTADO	ÚLTIMA MODIFICACIÓN	FECHA DE APERTURA	PRIORIDAD	SOLICITANTE - SOLICITANTE	ASIGNADO A - TÉCNICO	CATEGORÍA	TIEMPO EN RESOLV
20	Mi computadora no enciende - Elkin Damian Chulca Paspuel	<span style="color: green;">●</span> Nuevos	2026-01-16 14:21	2026-01-16 14:21	Media				
19	Mi computadora no enciende - Elkin Damian Chulca Paspuel	<span style="color: green;">●</span> Nuevos	2026-01-15 23:01	2026-01-15 23:01	Media				
18	Mi computadora no enciende - Elkin Damian Chulca Paspuel	<span style="color: green;">●</span> Nuevos	2026-01-15 22:30	2026-01-15 22:30	Media				
17	Mi computadora no enciende y necesito ayuda urgente - Maria Lopez	<span style="color: green;">●</span> Nuevos	2026-01-15 21:35	2026-01-15 21:35	Media				

*Nota.* Asignación de tickets al personal de soporte. Elaborado por: Los Autores

#### 4.1.5 Percepción de los usuarios sobre el sistema

Las impresiones de los usuarios con respecto al chatbot fueron obtenidas a través de encuestas conducidas en el transcurso de las interacciones. Los resultados fueron bastante

positivos con respecto a la aceptación del sistema, poniendo de manifiesto no solo el grado de facilidad con la que el sistema podría ser utilizado, así como la claridad de las respuestas a los usuarios y la velocidad en la atención de las interacciones.

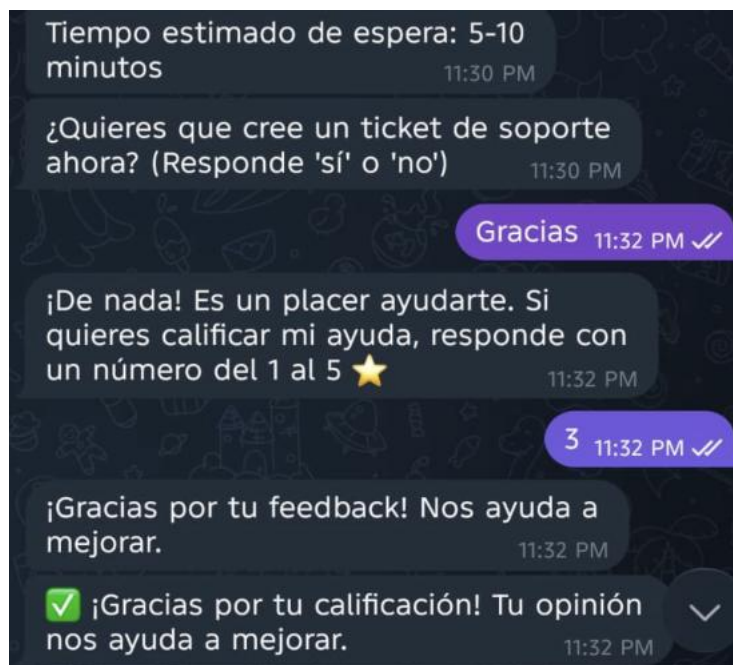
#### **4.2 Resultados de la evaluación del desempeño del chatbot**

La evaluación del chatbot se llevó a cabo con el fin de determinar su eficacia como herramienta de asistencia al soporte técnico de Tecnologías de la Información.

Se tuvieron en cuenta métricas de carácter cuantitativo y cualitativo que se definieron en la metodología que permitió analizar el comportamiento del sistema desde la perspectiva técnica, y la experiencia del usuario, los resultados obtenidos se analizan en un sentido comparativo, apoyándose en el modelo tradicional de atención manual en base a la valoración de factores como la precisión en la interpretación de las consultas, la cobertura funcional, los tiempos de respuesta y el nivel de satisfacción de las personas.

## Figura 27

*Calificación del usuario después de su respectiva ayuda*



*Nota.* Feedback para el chatbot donde se agrega la calificación. Elaborado por: Los Autores

### 4.2.1 Métricas de precisión en la interpretación de consultas

La exactitud del chatbot fue evaluada mediante el análisis de las interacciones capturadas durante la fase de pruebas funcionales, los resultados mostraron una alta tasa de reconocimientos correctos de los intentos correspondientes a las consultas relacionadas con los

problemas más frecuentes, que fueron el acceso a las plataformas académicas, la conectividad institucional y la pérdida de contraseñas.

#### **4.2.2 Cobertura funcional del sistema**

La cobertura del chatbot se analizó en función de su capacidad para atender la diversidad de requerimientos durante la fase de análisis. Los resultados evidenciaron que el sistema logró cubrir la mayoría de las incidencias de baja y mediana complejidad definidas en su alcance.

En comparación con el modelo tradicional, el chatbot demostró una mayor eficiencia al actuar como primer punto de contacto, filtrando y resolviendo consultas frecuentes. Esta cobertura funcional contribuyó a disminuir la carga operativa del área de soporte de TI y a optimizar el uso de los recursos disponibles.

#### **4.2.3 Análisis comparativo de tiempos de respuesta**

El tiempo de respuesta fue uno de los indicadores más relevantes en la evaluación del desempeño. En el modelo tradicional, los tiempos de atención dependían directamente de la disponibilidad del personal técnico, lo que generaba demoras variables. En contraste, el chatbot ofreció respuestas inmediatas, independientemente del horario o del número de usuarios concurrentes.

#### **4.2.4 Evaluación de la satisfacción del usuario**

La satisfacción del usuario se evaluó mediante encuestas aplicadas tras la interacción con el chatbot. Los resultados reflejaron valoraciones positivas en términos de facilidad, claridad de las respuestas y rapidez en la atención.

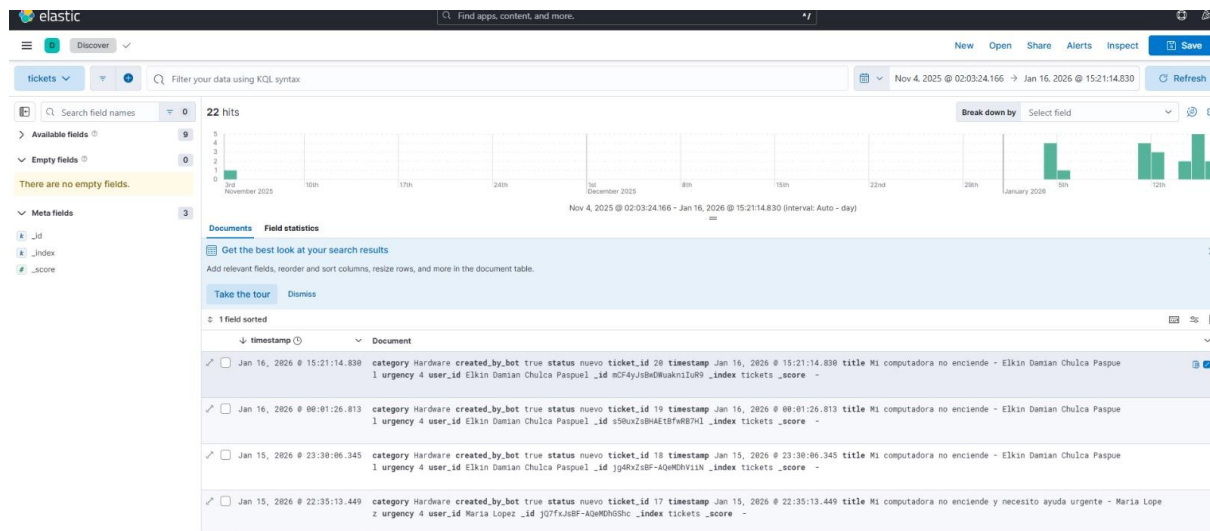
Los usuarios destacaron la utilidad del sistema como una herramienta accesible y eficiente para resolver problemas comunes, lo que reforzó su aceptación dentro del entorno universitario.

#### **4.2.5 Síntesis del desempeño del sistema**

En general, los resultados de la evaluación del desempeño demuestran que el chatbot cumple con los objetivos establecidos. La combinación de métricas técnicas y del análisis de la experiencia del usuario permite afirmar que el sistema representa una mejora significativa frente al modelo tradicional de soporte TI.

## Figura 28

### Gráficas de los casos en Elastic



*Nota.* Sistema de monitoreo a los chats de telegram. Elaborado por: Los Autores

### 4.3 Impacto del chatbot en el soporte TI del Bloque D

Uno de los principales efectos observados fue la reorganización del flujo de atención de incidencias, ya que el chatbot asumió el rol de primer punto de contacto. Esto permitió resolver consultas de baja complejidad y liberar al personal técnico para que se concentrara en incidencias que requieren intervención especializada. Desde la perspectiva del usuario, la

disponibilidad continua del sistema mejoró el acceso al soporte de TI y redujo la dependencia de horarios específicos.

#### **4.4 Análisis global de resultados y discusión**

El análisis global de los resultados permite integrar los hallazgos derivados de la implementación y la evaluación del chatbot, ofreciendo una visión consolidada sobre su efectividad como herramienta de apoyo al soporte de TI en el Bloque D.

Desde una perspectiva técnica, el uso de datos reales para entrenar el modelo NLP se consolidó como un factor clave para lograr una comprensión adecuada del lenguaje natural. Con esto podemos observar de forma general los casos creados para soporte de forma general después de realizar el caso y que se anexe al GLPI.

## CONCLUSIONES

El estudio llevado a cabo en torno a las principales problemáticas vinculadas al soporte de Tecnologías de la Información permitió comprender con mayor detalle el comportamiento de las peticiones generadas en el Bloque D. Gracias a la aplicación de entrevistas al personal técnico, encuestas dirigidas a los usuarios y el análisis de registros históricos de entradas se comprobó que una parte importante de los requisitos son consultas muy repetidas y de escasa complejidad. Las observaciones de este estudio señalan que el modelo tradicional del soporte no saca el rendimiento suficiente de las capacidades de recursos, esto apoya la necesidad de integrar mecanismos de automatización en el soporte técnico.

Complementariamente, el estudio de los flujos de interacción que se utilizaba en la gestión de las incidencias ha dado como resultado la identificación de patrones bien definidos en la comunicación entre los usuarios y el soporte. Estos patrones han facilitado la definición de etapas del proceso que pueden ser automatizadas sin que se pierdan las garantías de la calidad del servicio. En este sentido, se identifica que todo el primer nivel de soporte técnico puede ser gestionado a través de un sistema conversacional, lo que facilita una gestión más ordenada, ágil y consistente en la atención a la comunidad universitaria.

Con respecto a esa misma necesaria definición de la arquitectura del chatbot que haciendo posible estructurar la solución con el diseño de un sistema coherente en función de los requisitos funcionales establecidos en las fases anteriores de la investigación, la adecuada delimitación de intenciones, entidades y acciones más la creación de una base de conocimiento

adaptada a la planificación de la institución verificaron su modelado en un chatbot claro y consistente. Queda muy claro que todo ello no solo vino a satisfacer el soporte TI, sino que estableció las bases para una solución flexible, prefigurando futuras ampliaciones o cambios en los servicios tecnológicos de la institución, es decir, para un diseño un chatbot que podría adaptarse a las necesidades futuras.

Al final de la implementación del modelo de procesamiento de lenguaje natural, el uso del marco Rasa permitirá la obtención de un nivel satisfactorio de comprensión de las consultas de los usuarios, de la misma manera, el entrenamiento del modelo con datos reales de un entorno universitario fue determinante para la comprensión de las expresiones coloquiales y variaciones lingüísticas más frecuentes; en consecuencia el chatbot consiguió identificar correctamente las intenciones de los usuarios gestionando las solicitudes a partir de ellas, incrementando la calidad del soporte automatizado y reduciendo la intervención del técnico personal.

Finalmente, las pruebas y ensayos funcionales, de usabilidad y rendimiento, de los cuales se deriva la evaluación funcional del chatbot ya en producción, evidenciaron la mejora en el tiempo de respuesta al haber hecho frente a los requisitos más frecuentes de los usuarios, así como una buena percepción de los usuarios sobre la funcionalidad del servicio asociación de este chatbot. Por lo tanto, por el dicho, se puede confirmar que el sistema cumple las condiciones fijadas para la prestación de soporte técnico automatizado, lo que demuestra la viabilidad de una herramienta complementaria dentro del servicio de soporte institucional.

## RECOMENDACIONES

Comenzando a partir de la identificación de una alta cuota de consultas repetitivas y de baja complejidad, la Universidad Politécnica Salesiana debe continuar con el avance de la automatización progresiva del soporte técnico de Tecnologías de la Información, concediendo preferencia a las demandas que no requieren un asesoramiento especializado.

Esto permitirá mejorar el uso de los recursos humanos que efectuarán el servicio el cual, a su vez, reducirá la carga de trabajo del personal técnico, logrando mejorar la eficiencia general del servicio, manteniendo en todo caso el nivel de atención que recibe el usuario final. Debido a los patrones de interacción claramente definidos de los usuarios y del soporte, se recomienda a su vez consolidar el uso del chatbot como primer nivel de la atención técnica. En consecuencia, se trata de estandarizar los flujos conversacionales identificados y pasarlos formalmente al proceso institucional de soporte TI con la intención de garantizar una atención inicial más estructurada, más rápida y más coherente como filtro a seguir antes de pasar a la atención especial cuando sea necesario.

En cuanto a la arquitectura del sistema que hemos diseñado, se sugiere que continúe manteniendo y estableciendo un enfoque modular y flexible en el chatbot, que permita en el futuro ampliar más fácilmente el servicio de soporte TI; por otro lado, también se sugiere documentar continuamente las modificaciones que sufren la arquitectura y la base de conocimiento para facilitar así la escalabilidad del sistema a los requisitos futuros de tipo tecnológico o institucional.

Visto el buen rendimiento del modelo de procesamiento de lenguaje natural ejecutado con Rasa, sugerimos formalizar un proceso continuo de retroalimentación y de reentrenamiento del modelo a partir de nuevas interacciones reales que se lleven a cabo con los usuarios; esta práctica puede ayudar a progresar en la precisión de la interpretación de consultas, a adaptarse a los cambios que se producen en el lenguaje utilizado por la comunidad universitaria y ayudar a mejorar la calidad de la atención automática de modo que esta dependa cada vez menos de la intervención directa del personal técnico.

En virtud de todos los buenos resultados de las pruebas funcionales, pruebas de usabilidad y pruebas de rendimiento, se propone considerar su puesta en marcha como herramienta permanente dentro del servicio de soporte institucional. Asimismo, se considera necesarias la realización de las pruebas periódicas que permitan controlar el funcionamiento del chatbot, para conocer las oportunidades de mejorarlo, así como asegurar que su utilización se mantenga dentro de unos niveles óptimos de eficiencia, disponibilidad y satisfacción del usuario en un contexto de operación eficaz.

## REFERENCIAS BIBLIOGRÁFICAS

- Adamopoulou, E., & Moussiades, L. (2020, May). An overview of chatbot technology. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 373–383). Springer International Publishing.
- Cesila, C. H., Pinto, R. P., Mayer, K. S., Escallón-Portilla, A. F., Mello, D. A., Arantes, D. S., & Rothenberg, C. E. (2023, June). Chat-ibn-rasa: Building an intent translator for packet-optical networks based on Rasa. In *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)* (pp. 534–538). IEEE.
- Chen, C. C., Hung, M. H., Lai, K. C., & Lin, Y. C. (2021). Docker and Kubernetes. In *Industry 4.1: Intelligent Manufacturing with Zero Defects* (pp. 169–213).
- Gartner. (2022, July 27). Gartner predicts chatbots will become a primary customer service channel within five years. Gartner. <https://www.gartner.com/en/newsroom/press-releases/2022-07-27-gartner-predicts-chatbots-will-become-a-primary-customer-service-channel-within-five-years>

- Garzón-Quiroz, M., Campo-Saltos, G. D., & Loor-Ávila, B. (2025). Análisis sistemático sobre la eficiencia comunicativa entre chatbots basados en reglas y modelos de lenguaje natural. *Universitas-XXI, Revista de Ciencias Sociales y Humanas*, (42), 167–192.
- Gujjar, J. P., & Kumar, V. N. (2022). Open source chatbot development framework—RASA. *Asian Journal of Advances in Research*, 5(1), 451–453.
- Gunawan, T. S., Babiker, A. B. F., Ismail, N., & Effendi, M. R. (2021, August). Development of intelligent Telegram chatbot using natural language processing. In *2021 7th International Conference on Wireless and Telematics (ICWT)* (pp. 1–5). IEEE.
- Jesús, A. L. (2024). Implementación de un chatbot de soporte en la plataforma Aulas Virtuales. *Cuadernos Técnicos Universitarios de la DGTIC*, 2(4).
- Khaund, T., Hussain, M. N., Shaik, M., & Agarwal, N. (2020, October). Telegram: Data collection, opportunities and challenges. In *Annual International Conference on Information Management and Big Data* (pp. 513–526). Springer International Publishing.

- Ltd, A. (2025). What is IT service management? Axelos.  
<https://uat2.axelos.com/certifications/itil-service-management/what-is-it-service-management>
- Marin, D. J. (2023). *Manual de uso general software glpi*.
- Maruri, N. R. B., Reina, A. G. R., Lara, F. R. O., & Martínez, M. P. E. (2025). Automatización del soporte al cliente mediante un chatbot con IA integrado al CRM: Caso de estudio empresa EDITRATECH. *GADE: Revista Científica*, 5(3), 206–219.
- Mendez, J. (2026). Principios de SI: Stair & Reynolds, Capítulo I. Scribd.  
<https://es.scribd.com/document/521023200/1-Principios-de-SI-Stair-Reynolds-Cap-I-recognized>
- Park, D. M., Jeong, S. S., & Seo, Y. S. (2022). Systematic review on chatbot techniques and applications. *Journal of Information Processing Systems*, 18(1), 26-47.
- Pradana, Z. H., Nafi'ah, H., & Rochmanto, R. A. (2022). Chatbot-based information service using RASA open-source framework in Prambanan temple tourism object. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6(4), 656–662.

- Ruindungan, D. G., & Jacobus, A. (2021). Chatbot development for an interactive academic information services using the Rasa open source framework. *Jurnal Teknik Elektro dan Komputer*, 10(1), 61–68.
- Torres, M. J. S., Espin, W. R. N., Hurtado, H. Z., López, R. A. P., & Silva, D. H. Z. (2025). Uso de chatbots inteligentes como herramienta estratégica para la mejora de la atención al cliente en empresas de telecomunicaciones en Ecuador. *ARANDU UTIC*, 12(2), 3910–3919.
- Vasireddy, I., Ramya, G., & Kandi, P. (2023). Kubernetes and Docker load balancing: State-of-the-art techniques and challenges. *International Journal of Innovative Research in Engineering and Management*, 10(6), 49–54.

## Anexo

Fotografía en el Datacenter, donde recopilamos la información necesaria para realizar todo el proyecto técnico.

