



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
TÍTULO DEL TRABAJO DE TITULACIÓN**

**GESTIÓN DE SERVICIOS DE TI MEDIANTE CLASIFICACIÓN
AUTOMATIZADA DE SOLICITUDES BASADA EN PRINCIPIOS DE ITIL V4
UTILIZANDO MACHINE LEARNING.**

AUTOR

Balón Angel Aníbal Fernando

PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR
Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN
TUTOR

Ing. Coronel Suárez Marjorie Alexandra, Mgt.

Santa Elena, Ecuador

Año 2024



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

TRIBUNAL DE SUSTENTACIÓN

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

Ing. Marjorie Coronel Suárez, Mgt.
TUTOR

Ing. Walter Orozco Iguasnia, Msc.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez, Mgt.
DOCENTE GUÍA UIC



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

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

La Libertad, a los 02 días del mes de diciembre del año 2024

TUTOR



Firmado electrónicamente por:
MARJORIE ALEXANDRA
CORONEL SUAREZ

Ing. Marjorie Coronel Suárez, Mgt.



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

Yo, **Balón Angel Aníbal Fernando**

DECLARO QUE:

El trabajo de Titulación, Gestión de servicios de TI mediante clasificación automatizada de solicitudes basadas en principios de ITIL v4 utilizando Machine Learning previo a la obtención del título de Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

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

La Libertad, a los 02 días del mes de diciembre del año 2024

EL AUTOR

A handwritten signature in blue ink, which appears to read "Balón Angel Aníbal Fernando".

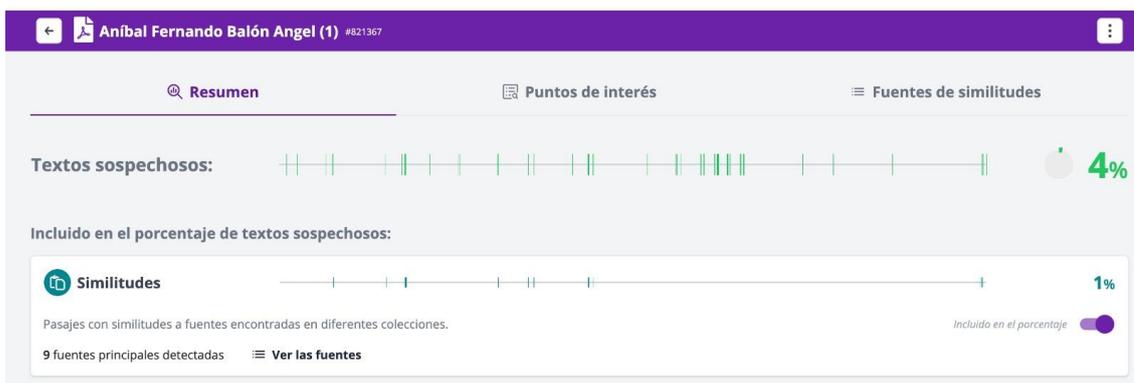
Aníbal Fernando Balón Angel



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

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Gestión de servicios de TI mediante clasificación automatizada de solicitudes basadas en principios de ITIL v4 utilizando Machine Learning, presentado por el estudiante, Aníbal Fernando Balón Angel, fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 4%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTOR



Firmado electrónicamente por:
**MARJORIE ALEXANDRA
CORONEL SUAREZ**

Ing. Marjorie Coronel Suárez, Mgt.



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

Yo, Balón Angel Aníbal Fernando

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

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

La Libertad, a los 02 días del mes de diciembre del año 2024

EL AUTOR

Aníbal Fernando Balón Angel

AGRADECIMIENTO

Mi gratitud en primer lugar es hacia Dios, por darme la fuerza y sabiduría, por estar ahí en todo tiempo, durante todo este proceso, por permitirme llegar hasta este momento muy importante en mi vida que es culminar este período académico. A toda mi familia por ese apoyo incondicional, por esas palabras de aliento que me impulsaban día a día durante toda la etapa de mi carrera. Agradezco a todos los docentes que fueron parte de mi formación académica. A aquellos amigos que pude conocer en el transcurso de la carrera, por su confianza y creer en que se iba a lograr esta meta. A la Ing. Marjorie Coronel por su orientación y conocimiento que fueron guía fundamental para culminar con éxito este proyecto de titulación.

Aníbal Fernando, Balón Angel

DEDICATORIA

Dedico este trabajo a Dios, quien me guio durante este proceso y me permitió culminar con éxito.

A mi madre Teodora Flor Angel Floreano y mi padre Justo German Balón Suárez, por su ayuda en todo momento, ellos fueron parte fundamental y siempre se los agradeceré, también a mis hermanos: Juan Balón, Jefferson Balón, José Balón y Teresa Balón, por apoyarme en toda esta etapa de la carrera.

A eso amigos que conocí al inicio de la carrera: Hugo Bazán y Carlos Labayen quienes fueron de gran ayuda en todo momento.

Aníbal Fernando, Balón Angel

ÍNDICE GENERAL

TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE ILUSTRACIONES	XI
ÍNDICE DE TABLAS	XIV
RESUMEN	XVII
ABSTRACT	XVIII
INTRODUCCIÓN	1
CAPITULO 1. FUNDAMENTACIÓN	3
1.1 Antecedentes	3
1.2 Descripción del Proyecto	4
1.3 Objetivo del Proyecto	7
1.4 Justificación del Proyecto	8
1.5 Alcance del Proyecto	9
1.6 Metodología del Proyecto	10
1.6.1 Metodología de Investigación	10

1.6.2 Beneficiarios del Proyecto	11
1.6.3 Variables	12
1.6.4 Análisis de recolección de datos	12
1.7 Metodología de desarrollo	13
CAPÍTULO 2. PROPUESTA	15
2.1 Marco Contextual	15
2.2 Marco Conceptual	16
2.2.1 ITIL (Information Technology Infrastructure Library)	16
2.2.2 ITIL v4	16
2.2.3 Gestión de solicitudes de servicio.	16
2.2.4 Gestión de catálogos de servicios	18
2.2.5 Machine Learning	19
2.3 Marco Teórico	22
2.3.1 Implementación de las buenas prácticas de ITIL v4: Mejorar la gestión de solicitudes.	22
2.3.2 Aplicación de Machine Learning: Modelos para clasificación de solicitudes.	22
2.3.3 Mejoras en la experiencia de usuario: Análisis del servicio de Help desk Zammad.	23
2.4 Requerimientos	23
2.4.1 Requerimientos Funcionales	23
2.4.2 Requerimientos no Funcionales	27
2.5 Componentes de la propuesta	28
2.5.1 Arquitectura del sistema	28

	29
2.5.3 Diseño del catálogo de solicitudes de servicios	29
2.5.4 Proceso de limpieza de datos	30
2.5.5 Entrenamiento de modelos de aprendizaje supervisados	31
2.5.5 Diagramas de caso de uso	46
2.5.6 Modelado de Datos	53
2.6 Diseño de Interfaces	54
2.7 Pruebas	63
2.8 Resultados	71
CONCLUSIONES	73
RECOMENDACIONES	74
REFERENCIAS	75
Anexo 1. Catálogo de solicitudes de servicios.	79
Anexo 2. Árbol de problemas.	79

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Metodología del desarrollo del proyecto	14
Ilustración 2 Mapa de calor de la contribución de la gestión de solicitudes de servicio al valor [19].	18
Ilustración 3 Mapa de calor de la contribución de la gestión del catálogo de servicios al valor [19].....	19
Ilustración 4 Arquitectura del sistema del proyecto.....	28
Ilustración 5 Diagrama del proceso de solicitudes.....	29
Ilustración 6 Catálogo de solicitudes.	30

Ilustración 7 Limpieza y depuración de datos.....	31
Ilustración 8 Proceso 1 del entrenamiento del modelo SVM.....	33
Ilustración 9 Proceso 2 del entrenamiento del modelo SVM.....	33
Ilustración 10 Proceso 3 del entrenamiento del modelo SVM.....	34
Ilustración 11 Proceso 4 del entrenamiento del modelo SVM.....	34
Ilustración 12 Resultados del modelo Support Vector Machine (SVM).	35
Ilustración 13 Resultados matriz confusión del modelo Support Vector Machine (SVM).	36
Ilustración 14 Proceso 1 del entrenamiento del modelo Naive Bayes.	36
Ilustración 15 Proceso 2 del entrenamiento del modelo Naive Bayes.	37
Ilustración 16 Resultados del modelo Naive Bayes.	37
Ilustración 17 Resultados matriz confusión del modelo Naive Bayes.	38
Ilustración 18 Proceso 1 del entrenamiento del modelo K- Nearest Neighbor (KNN).	39
Ilustración 19 Proceso 2 del entrenamiento del modelo K- Nearest Neighbor (KNN).	39
Ilustración 20 Resultados del modelo K- Nearest Neighbor (KNN).	40
Ilustración 21 Resultados matriz confusión del modelo K- Nearest Neighbor (KNN).	40
Ilustración 22 Proceso 1 del entrenamiento del modelo Logistic Regression.	41
Ilustración 23 Proceso 2 del entrenamiento del modelo Logistic Regression.	41
Ilustración 24 Resultados del modelo Logistic Regression.	42
Ilustración 25 Resultados matriz confusión del modelo Logistic Regression.	43
Ilustración 26 Proceso 1 del entrenamiento del modelo Random Forest.....	43

Ilustración 27 Proceso 2 del entrenamiento del modelo Random Forest.....	44
Ilustración 28 Resultados del modelo Random Forest.....	44
Ilustración 29 Resultados matriz confusión del modelo Random Forest.....	45
Ilustración 30 Métricas de Evaluación de Modelos de Clasificación.	46
Ilustración 31 Diagrama de caso de uso – rol general.	47
Ilustración 32 Diagrama de caso de uso - acceso al sistema.....	48
Ilustración 33 Diagrama de caso de uso – rol del usuario.....	50
Ilustración 34 Diagrama de caso de uso – rol del administrador	52
Ilustración 35 Diseño de base de datos PostgreSQL.....	54
Ilustración 36 Interfaz inicio de sesión de rol usuario.	55
Ilustración 37 Interfaz rol usuario: Crear solicitud.	56
Ilustración 38 Interfaz de creación de Tickets en Zammad.	56
Ilustración 39 Interfaz rol usuario: Ver detalles de tickets.	57
Ilustración 40 Interfaz rol usuario: Ver respuesta de soporte 1.	57
Ilustración 41 Interfaz rol usuario: Ver respuesta de soporte 2.	58
Ilustración 42 Interfaz rol Administrador - Gestionar solicitudes en Zammad. ...	58
Ilustración 43 Interfaz rol Administrador - Ver solicitudes existentes.	59
Ilustración 44 Ver solicitudes existentes - Gestionar solicitudes en Zammad.....	59
Ilustración 45 Ver solicitudes por estado	60
Ilustración 46 Ver reportes – Semanal, mensual y anual.	61
Ilustración 47 Registro de usuario – Mediante el Zammad	61
Ilustración 48 Monitoreo de tickets en grafana.	62
Ilustración 49 Tiempo en clasificación de solicitudes	62

Ilustración 50 Predicción de las solicitudes registradas	63
Ilustración 51 Total de solicitudes correctas	63

ÍNDICE DE TABLAS

Ilustración 1 Metodología del desarrollo del proyecto	14
Ilustración 2 Mapa de calor de la contribución de la gestión de solicitudes de servicio al valor [19].	18
Ilustración 3 Mapa de calor de la contribución de la gestión del catálogo de servicios al valor [19].	19
Ilustración 4 Arquitectura del sistema del proyecto.	28
Ilustración 5 Diagrama del proceso de solicitudes	29
Ilustración 6 Catálogo de solicitudes.	30
Ilustración 7 Limpieza y depuración de datos.	31
Ilustración 8 Proceso 1 del entrenamiento del modelo SVM.	33
Ilustración 9 Proceso 2 del entrenamiento del modelo SVM.	33
Ilustración 10 Proceso 3 del entrenamiento del modelo SVM.	34
Ilustración 11 Proceso 4 del entrenamiento del modelo SVM.	34
Ilustración 12 Resultados del modelo Support Vector Machine (SVM).	35
Ilustración 13 Resultados matriz confusión del modelo Support Vector Machine (SVM).	36
Ilustración 14 Proceso 1 del entrenamiento del modelo Naive Bayes.	36
Ilustración 15 Proceso 2 del entrenamiento del modelo Naive Bayes.	37
Ilustración 16 Resultados del modelo Naive Bayes.	37
Ilustración 17 Resultados matriz confusión del modelo Naive Bayes.	38

Ilustración 18 Proceso 1 del entrenamiento del modelo K- Nearest Neighbor (KNN).	39
Ilustración 19 Proceso 2 del entrenamiento del modelo K- Nearest Neighbor (KNN).	39
Ilustración 20 Resultados del modelo K- Nearest Neighbor (KNN).	40
Ilustración 21 Resultados matriz confusión del modelo K- Nearest Neighbor (KNN).	40
Ilustración 22 Proceso 1 del entrenamiento del modelo Logistic Regression.	41
Ilustración 23 Proceso 2 del entrenamiento del modelo Logistic Regression.	41
Ilustración 24 Resultados del modelo Logistic Regression.	42
Ilustración 25 Resultados matriz confusión del modelo Logistic Regression.	43
Ilustración 26 Proceso 1 del entrenamiento del modelo Random Forest.	43
Ilustración 27 Proceso 2 del entrenamiento del modelo Random Forest.	44
Ilustración 28 Resultados del modelo Random Forest.	44
Ilustración 29 Resultados matriz confusión del modelo Random Forest.	45
Ilustración 30 Métricas de Evaluación de Modelos de Clasificación.	46
Ilustración 31 Diagrama de caso de uso – rol general.	47
Ilustración 32 Diagrama de caso de uso - acceso al sistema	48
Ilustración 33 Diagrama de caso de uso – rol del usuario	50
Ilustración 34 Diagrama de caso de uso – rol del administrador	52
Ilustración 35 Diseño de base de datos PostgreSQL.	54
Ilustración 36 Interfaz inicio de sesión de rol usuario.	55
Ilustración 37 Interfaz rol usuario: Crear solicitud.	56
Ilustración 38 Interfaz de creación de Tickets en Zammad.	56

Ilustración 39 Interfaz rol usuario: Ver detalles de tickets.	57
Ilustración 40 Interfaz rol usuario: Ver respuesta de soporte 1.	57
Ilustración 41 Interfaz rol usuario: Ver respuesta de soporte 2.	58
Ilustración 42 Interfaz rol Administrador - Gestionar solicitudes en Zammad.	58
Ilustración 43 Interfaz rol Administrador - Ver solicitudes existentes.	59
Ilustración 44 Ver solicitudes existentes - Gestionar solicitudes en Zammad.	59
Ilustración 45 Ver solicitudes por estado	60
Ilustración 46 Ver reportes – Semanal, mensual y anual.	61
Ilustración 47 Registro de usuario – Mediante el Zammad	61
Ilustración 48 Monitoreo de tickets en grafana.	62
Ilustración 49 Tiempo en clasificación de solicitudes	62
Ilustración 50 Predicción de las solicitudes registradas	63
Ilustración 51 Total de solicitudes correctas	63

RESUMEN

Este presente trabajo trata sobre la implementación de un sistema de gestión de servicios de TI mediante clasificación automatizada de solicitudes basadas en principios de ITIL v4 con el uso de Machine Learning. El propósito es implementar un sistema capaz de clasificar de manera automática las solicitudes emitidas por los usuarios, mediante la mensajería de Telegram, por lo que para cumplir dicho propósito se llevó un proceso de recopilación de datos de diferentes fuentes externas que tengas relación con descripciones en solicitudes, que luego fueron depurados para su uso en el entrenamiento de modelos de Machine Learning, se aplicaron modelos de aprendizaje supervisados para entrenar el sistema de clasificación de solicitudes, se evaluó el mejor modelo de clasificación y se integró al sistema (chatbot). También se integró Zammad, un software de código abierto y creación de tickets, donde el departamento de TICS se encargará de optimizar las solicitudes de cada usuario. Al mismo tiempo, Zammad permite mantener un registro de todas las interacciones y solicitudes, para una mejora continua en los servicios de TI.

Palabras claves: Gestión de Solicitudes, ITIL v4, Aprendizaje automático.

ABSTRACT

This present work deals with the implementation of an IT service management system through automated classification of requests based on ITIL v4 principles with the use of Machine Learning. The purpose is to implement a system capable of automatically classifying requests issued by users, through Telegram messaging, so to fulfill this purpose, a process of data collection was carried out from different external sources that are related to descriptions in requests, which were then debugged for use in training Machine Learning models, supervised learning models were applied to train the request classification system, the best classification model was evaluated and integrated into the system (chatbot). Zammad, an open source and ticket creation software, was also integrated, where the ICT department will be in charge of optimizing the requests of each user. At the same time, Zammad allows you to keep a record of all interactions and requests, for continuous improvement in IT services.

Keywords: Request Management, ITIL v4, Machine Learning.

INTRODUCCIÓN

En la era actual de la tecnología de la información, el uso de Machine Learning ha emergido como una herramienta clave para mejorar los procesos y la eficiencia en diversas áreas. En particular, la gestión de servicios de tecnologías de la información (TI) ha experimentado una evolución significativa con la adopción de marcos como ITIL (Information Technology Infrastructure Library), que proporciona un conjunto de mejores prácticas para la gestión eficiente de los servicios de TI. Sin embargo, a medida que las organizaciones crecen y el volumen de solicitudes aumenta, surge la necesidad de soluciones automatizadas que permitan optimizar la clasificación y gestión de solicitudes de manera precisa y eficiente. En este sentido, la automatización mediante herramientas como chatbot, combinada con técnicas de machine learning, se ha presentado como una solución prometedora.

El objetivo principal de este trabajo de titulación es diseñar un sistema para gestionar servicios de TI a través de la incorporación del manejo automático de solicitudes y con base en las mejores prácticas del ITIL V4 y machine learning. El proyecto intenta brindar una forma fácil para la gestión y clasificación de solicitudes, donde los usuarios tendrán a su disposición un chatbot que registrará y organizará las solicitudes de los usuarios automáticamente en categorías y servicios según lo que se infiera de los detalles. Adicionalmente, el sistema les permitirá a los administradores monitorear las solicitudes, controlarlas y generar reportes automáticamente todo dentro de un marco de ITIL.

El sistema se basa en el uso de machine learning para agrupar estas solicitudes clasificándolas y categorizándolas de manera precisa y correcta. Para lograr esto, se realizó un procedimiento de limpieza de datos exhaustivo, que involucró la purga de los datos de entrada donde existía duplicación, la corrección de datos mal formateados y la normalización de valores variados. Esta etapa se volvió esencial porque permitió que el modelo de aprendizaje automático fuera entrenado en un conjunto de datos limpios. Además, el modelo fue entrenado utilizando técnicas de clasificación de texto supervisados, lo que permitió que el sistema fuera capaz de entender y procesar las solicitudes de manera efectiva.

El proyecto también integra el uso de un chatbot en Telegram que potencia la interacción entre los usuarios y el sistema. El chatbot permite a los usuarios registrar nuevas solicitudes, consultar el estado de sus tickets, observar en que categoría y servicio fue clasificada la solicitud y recibir respuestas de soporte o del administrador. Los administradores tienen el acceso al sistema para gestionar todas las solicitudes y dar seguimiento a los tickets mediante una interfaz que conecta con Zammad. La integración de Zammad y la generación de reportes automáticos mediante la consulta a la base de datos PostgreSQL donde se registran las solicitudes, usuarios y tickets y el uso de Grafana permite un control más detallado sobre las solicitudes y el rendimiento del sistema.

El sistema diseñado debe facilitar la experiencia tanto de los usuarios como de los administradores al mejorar la eficiencia en la clasificación de solicitudes, la gestión de tickets y la elaboración de reportes. El objetivo del presente trabajo fue proponer un marco de gestión de sistema como servicio utilizando ITIL v4 en conjunto con técnicas avanzadas de aprendizaje automático y análisis de datos con el fin de automatizar la solución para la provisión de la gestión de servicios de TI.

CAPÍTULO 1. FUNDAMENTACIÓN

1.1 Antecedentes

Con el paso de los años, las empresas han demostrado su capacidad de cambio constante al adaptar sus modelos de negocio para satisfacer las expectativas de sus clientes, las exigencias del tiempo han impulsado la adaptación de los procesos internos y externos, las empresas deben adaptarse constantemente a los avances tecnológicos para sobresalir en el mercado competitivo actual [1].

La gestión de servicios de TI en referencia a la madurez digital, las instituciones enfrentan importantes desafíos digitales, según el informe Universitic, el 26% de las universidades han llegado a un nivel de satisfacción en la práctica de gestión de TI, este porcentaje se debe a la falta de recursos humanos, financieros, poca disponibilidad, seguridad de la información en las instituciones u organizaciones y una falta en la planificación de estrategias, esto impide el desarrollo integral y eficaz de la tecnología en las instituciones al no mejorar su eficiencia operativa [2].

Las instituciones latinoamericanas proporcionan un número bastante amplio de servicios de TI, las universidades españolas con un promedio de 49 servicios en TI, Argentina 28, Ecuador 17, México 26, estos servicios, aunque estén un poco elevados, no quiere decir que trabajen de forma óptima. A menudo, estos servicios no están actualizados y tampoco optimizados, la desconexión de los servicios y la interacción con el usuario afecta negativamente en la satisfacción. Además, la falta de planificación y diversos mecanismos de control y acuerdos de nivel de servicio (SLA) presenta una ineficiente gestión en servicios [3].

Las empresas u organizaciones dependen ampliamente de los servicios de TI para llevar a cabo sus operaciones de manera más eficiente y efectiva. En el área de gestión de solicitudes muchas organizaciones enfrentan una serie de desafíos que dificultan su eficacia a nivel global. En este trabajo ‘Analizar, diseñar e implementar un sistema de información que soporte el proceso de Gestión de Solicitudes de Servicio del Ministerio de Comunicaciones de una Iglesia Evangélica’, menciona que los procesos de clasificación de solicitudes son manuales y poco eficientes en la recepción de solicitudes por correo electrónico y su registro en hojas de cálculo resulta un proceso lento y propenso a errores y

perdida de información, esta también dificulta la planificación y entrega de servicios [4].

Johana L. Rosas Flores con el trabajo de titulación ‘Implementación de una solución de inteligencia de negocios para optimizar la gestión de solicitudes del área de preventa en una empresa de telecomunicaciones en la ciudad de Lima’, recalca que enfrentan desafíos en la gestión de solicitudes de implementaciones de proyectos para el segmento de Grandes Empresas. Estas solicitudes, definidas por los asesores comerciales, experimentan incumplimientos de los Acuerdos de Nivel de Servicio (SLA) debido a diversos factores [5].

Entre ellos se encuentran la falta de visibilidad de los casos, demoras en respuestas de otras áreas y errores en la derivación de las solicitudes, esta problemática incide en el aprovechamiento de las oportunidades y ocasiona pérdidas para la empresa, asimismo, la falta de una herramienta para gestionar y rectificar los casos representa una carga para los líderes de proyecto encargados de la supervisión [5].

En el tema ‘Análisis para la Implementación de la herramienta Bayestree’ de Carlos Vicente Roballo, resalta que en la gestión de solicitudes de servicio y por fallas del mismo hay un aumento en las quejas de los clientes y una disminución en su satisfacción, lo que puede resultar en la cancelación de servicios [6]. Se han identificado varios problemas en el proceso de atención a solicitudes de los clientes, como la ausencia de una gestión del conocimiento adecuada, deficiencias en la gestión de solicitudes y comunicación con otras áreas de servicios, estos contratiempos ponen de manifiesto la falta de eficiencia en la clasificación y gestión de solicitudes de servicio, esto ha provocado que la insatisfacción en los clientes aumente.

1.2 Descripción del Proyecto

El proyecto tiene como objetivo plantear un sistema de gestión de solicitudes con la finalidad de clasificar solicitudes y categorizarlas de manera automática, aplicando Machine Learning, optimizando así la sobrecarga de solicitudes para el departamento de TI. Este sistema facilitará la resolución de manera rápida y precisa de las peticiones de los usuarios, mejorando y dando un servicio de calidad. El uso

de Telegram donde los usuarios puedan enviar sus solicitudes de manera sencilla, las que automáticamente serán clasificadas mediante el uso de Machine Learning, este sistema utiliza modelos entrenados donde identifica la categoría respectiva de cada solicitud o descripción de esta y a la vez en ese proceso se genera un ticket en Zammad para su seguimiento, este proceso ayuda a optimizar los registros de las solicitudes.

El proyecto se fundamenta en las prácticas óptimas de la gestión de servicios de TI (ITSM), ITIL es una metodología de administración de TI muy conocida que proporciona un conjunto de procedimientos y prácticas para mejorar la planificación, diseño, entrega y soporte de los servicios de TI. Su meta principal es optimizar y perfeccionar los procedimientos y acciones de Tecnología de la Información en la compañía [7].

El implemento de este principio de ITIL v4 en el proyecto es la Gestión de solicitudes de servicios, esta práctica garantiza que las solicitudes de servicio sean atendidas de manera efectiva, eficiente y consistente.

El chatbot de mensajería de Telegram estará estructurada por los siguientes módulos:

ROL DE USUARIO.

MODULO DE VERIFICACION DE USUARIOS

- Comprueba si el usuario esta registrado en el sistema.
- Permitir uso del sistema al usuario ya registrado.

MODULO DE VALIDACION DE SOLICITUDES

- Permitir a los usuarios ingresar solicitudes.
- Validar solicitudes coherentes y detalladas.

MODULO DE CLASIFICACION DE SOLICITUDES

- Registrar solicitudes y clasificarlas automáticamente en el servicios y categorías según los estándares de ITIL v4.
- Generar tickets.

MODULO DE CONSULTA O DETALLES DE TICKETS

- Proporcionar detalles de los estados de tickets y solicitudes.

MODULO DE CIERRE DE SOLICITUDES

- Actualización del estado de la solicitud.
- Enviar mensaje de la solicitud resuelta.

ROL ADMINISTRADOR

MODULO DE GESTIONAR SOLICITUDES

- Permitir observar todas las solicitudes existentes.
- Redirigir a Zammad.
- Ver solicitudes por estado.

MODULO DE VER REPORTES

- Observar reportes semanales.
- Observar reportes mensuales.
- Observar reportes anuales.

Se detallan también las herramientas que se usaran en el desarrollo del proyecto:

Zammad: Es una plataforma de soporte y gestión de tickets gratuita, de código abierto, que simplifica la comunicación del usuario mediante diversos medios o canales como el correo electrónico, el chat, Telegram, entre otros. Su interfaz y habilidad facilitan su integración con otras herramientas, proporcionando una solución eficiente [8].

Telegram: Es una aplicación de mensajería gratis, rápida y segura, facilita una API para integrar un chatbot y agilizar los procesos, es una herramienta que favorece en la clasificación de gestión de solicitudes [9].

TypeScript: Es un lenguaje de programación de código abierto de JavaScript, es decir, de acceso y uso libre, sirve para aplicar y ejecutar en proyectos, nos permite trabajar de manera estructurada [10].

Postman: Herramienta que permite el desarrollo y pruebas de APIs. Es muy útil para programar porque da la posibilidad crear, enviar solicitudes en HTTP. También nos permite comprobar el correcto funcionamiento de proyectos [11].

Express.js: Es un marco de desarrollo para Node.js que permite estructurar una aplicación de una manera ágil, nos proporciona funcionalidades como el enrutamiento, opciones para gestionar sesiones, entre otros [12].

Python: Es un lenguaje de programación y de alto nivel, ampliamente utilizado en las aplicaciones o desarrollo web, software, ciencia de datos y el machine learning (ML) [13].

Grafana: Plataforma de desarrollo libre e interactiva, permite visualizar métricas e información en tiempo real de la información obtenida. Principalmente se emplea para supervisar infraestructuras y aplicaciones de tecnología de la información [14].

Visual Studio Code: Esta herramienta permite visualizar, modificar, ejecutar y depurar códigos de aplicaciones o proyectos, es extremadamente veloz y liviana, ofrece soporte en varios lenguajes de programación, identificación de fragmentos de códigos, y es compatible con Windows, Linux o Mac, entre otros [15].

Elasticsearch: Es un motor de búsqueda y análisis distribuido, escalable y flexible, que funciona como un depósito central de datos para búsquedas veloces y análisis detallados. Es un componente clave del Elastic Stack, proporcionando soluciones desde vigilancia en tiempo real hasta inteligencia empresarial, facilitando a las organizaciones la obtención de percepciones útiles de grandes cantidades de datos [16].

1.3 Objetivo del Proyecto

1.3.1. Objetivo General

Implementar un sistema de gestión de servicios de TI que utilice técnicas de clasificación automatizada de solicitudes basadas en los principios de ITIL v4, empleando machine learning para mejorar la eficiencia operativa.

1.3.2. Objetivos Específicos

- Depurar los datos de solicitudes de servicios de TI para el entrenamiento de modelos de machine learning, eliminando errores, inconsistencias y datos irrelevantes, asegurando así la precisión del dataset.
- Comparar modelos de machine learning capaces de clasificar solicitudes de servicio de TI, evaluando su desempeño para identificar el modelo más adecuado.
- Integrar el modelo de clasificación automatizada con el sistema de gestión de servicios de TI basado en ITIL v4.
- Evaluar el rendimiento del modelo de clasificación automatizada en términos de precisión y eficiencia en un entorno operativo de servicios de TI, monitoreando su impacto en la reducción del tiempo.

1.4 Justificación del Proyecto

En el mundo actual, el avance de la tecnología o TI está tomando gran impacto en las organizaciones y con ella en todas sus áreas, convirtiéndose en herramientas fundamentales para su desarrollo y con ello ser más eficientes y eficaces en los procesos y servicios que brindan. Con la finalidad de optimizar tiempo y reducir costos, para ello es primordial aplicar el uso de los servicios de TI, que en base a distintos métodos se puede automatizar procesos, optimizar y mejorar, esto lleva a las empresas u organizaciones a ser competitivas, es por ello es necesario implementar la metodología ITIL v4 [7]. La que plantea diversas prácticas para la gestión y esta pueda ser eficiente y efectiva de servicios de TI en las organizaciones.

La aplicación del proceso de gestión de solicitudes de servicio de TI para diversas organizaciones, permiten el registro y orden de solicitudes, de manera que los datos requeridos no son obviados o ingresados con errores, el Sistema de Información se diseñó con una arquitectura de software que satisface a las organizaciones, específicamente, con los procesos de gestión de Solicitudes de Servicio y de planificación de tareas [4].

Diversas organizaciones están empleando soluciones tecnológicas basadas en inteligencia artificial y aprendizaje automático para optimizar la administración de peticiones de servicio. Estas soluciones facilitan la clasificación, categorización y asignación automatizada de las peticiones, contribuyendo a disminuir los errores humanos y a extender los tiempos de respuesta al elevar las peticiones a niveles superiores [6].

Este proyecto se rige mediante la secretaría nacional de planificación del plan de creación de oportunidades 2021 – 2025:

Objetivo del Eje Social:

Objetivo 7: Potenciar las capacidades de la ciudadanía y promover una educación innovadora, inclusiva y de calidad en todos los niveles [17].

Políticas:

7.2 Promover la modernización y eficiencia del modelo educativo por medio de la innovación y el uso de herramientas tecnológicas [17].

1.5 Alcance del Proyecto

El propósito de este proyecto es instaurar un sistema de categorización automática de peticiones para mejorar la administración de servicios en el departamento de Tecnologías de la Información, dado que la mayoría de las organizaciones se encuentran con grandes cantidades de peticiones que se clasifican manualmente, lo que demora la resolución. Las peticiones realizadas por los usuarios necesitan un tratamiento rápido y una categorización exacta.

La implementación de este chatbot en la mensajería de Telegram facilita el uso para el usuario y la interacción en el proceso de envío de solicitudes y esto se rige a la metodología de ITIL v4, también reconocido por sus óptimas prácticas en la administración de servicios relacionados con la tecnología de la información. Este método proporciona una estructura sólida para determinar los procesos, funciones y tareas en la administración de peticiones. Además, el chatbot estará integrado a la herramienta Zammad, que proporciona un ticket para cada solicitud.

El chatbot lleva una estructura con los siguientes módulos: Módulo de verificación de usuarios, se comprueba si el usuario al ingresar su número de cédula se encuentra registrado en el sistema de Zammad y permite registrar nuevos usuarios cuando esta sea necesario. El Módulo de validación de solicitudes, se asegura que las descripciones de las solicitudes escritas por los usuarios sean coherentes y tengan un nivel de detalles, para su identificación y seguimiento adecuado. El Módulo de clasificación de solicitudes, asigna de manera automática el servicio y categoría de las descripciones emitidas por los usuarios. El Módulo de consulta de detalles de tickets, permite consultar al usuario sobre la solicitud registrada y ver detalles de aquello, como: la categoría, el servicio, descripción, fecha de registro y estado del ticket. El Módulo de seguimiento, se enviará una notificación al usuario sobre el estado actual de la solicitud y ticket. El Módulo de cierre, el chatbot finaliza la conversación con el usuario al completar los procesos anteriores.

Para alcanzar estos objetivos, se llevó un proceso de selección de datos de registros de solicitudes, donde se identificó elementos claves para la clasificación automática. Luego, se realizó una limpieza profunda de los datos, eliminando duplicados, celdas vacías, corrigiendo inconsistencias y estandarizando formatos, asegurando así los datos para el entrenamiento de los modelos. Posteriormente se implementó técnicas de Machine Learning con modelos supervisados e iterando en el ajuste de sus hiperparámetros para mejorar el rendimiento del modelo, se realizó pruebas y se evaluó la precisión, sensibilidad, exactitud y una validación cruzada para verificar las predicciones, de cada una de ellas y se seleccionó al mejor modelo con la mejor precisión para la tarea de clasificación. Este modelo se implementará en el chatbot de Telegram. Gracias a este proceso, el sistema está capacitado para manejar solicitudes y clasificarlas de forma automática.

1.6 Metodología del Proyecto

1.6.1 Metodología de Investigación

El proyecto se llevará a cabo la metodología de tipo de estudio Experimental [18], con la finalidad de establecer un sistema de categorización automática de peticiones. Este método se fundamenta en el manejo regulado de variables, la

comparación de resultados y la elección del modelo de aprendizaje automático (Machine Learning) más adecuado para solucionar el problema propuesto.

Para llevar a cabo este proyecto se realizó una investigación profunda de fuentes confiables sobre clasificación de solicitudes, a nivel local, nacional e internacional, recopilando información artículos científicos, trabajos o tesis. La revisión de literatura permitió identificar las mejores prácticas para el proyecto.

Se realizaron evaluaciones de herramientas y tecnologías para identificar su desempeño en la resolución de la propuesta, considerando las ventajas y restricciones de cada una. Se examinaron ejemplos de compañías u organizaciones que las pusieron en marcha, reconociendo tácticas eficaces y retos en la administración de peticiones.

El método experimental facilita la comparación del rendimiento de distintos algoritmos de aprendizaje automático en un ambiente regulado. Incluyen los algoritmos supervisados escogidos: Logistic Regression, Machine Support Vector (SVM), K-Nearest Neighbor (KNN), Random Forest y Naive Bayes, se aplican métricas de precisión, accuracy, recall y F1 Score.

El uso de datos (dataset) de acceso libre relacionado con clasificación de solicitudes o servicios, estos datos sirven para el entrenamiento con el 80% y pruebas un 20% con las técnicas de Machine Learning, como también para la selección del modelo adecuado, para el uso correcto de los datos se emplea un proceso de limpieza, vectorización TF-IDF y SMOTE para el balanceo de datos. Finalmente, se realiza un monitoreo para garantizar la eficiencia del sistema, permitiendo realizar ajustes y mejoras continuas.

1.6.2 Beneficiarios del Proyecto

Los beneficiarios de este proyecto es el departamento de soporte de TI y el cuerpo administrativo que gestionan las solicitudes en una gobernación. El implemento de este sistema de gestión de clasificación de solicitudes, mediante Telegram y la integración con Zammad, permite facilitar el proceso en la recepción y manejo de las solicitudes. Esto reduce el tiempo en que dichas solicitudes sean atendidas,

también ayuda a los miembros del departamento a enfocarse en otras tareas, mejorando así la eficacia general y de respuestas del departamento de TI.

1.6.3 Variables

En este proyecto se evaluará la precisión en la categorización de las solicitudes y el tiempo de respuesta o notificación al usuario, la precisión se medirá mediante la correcta clasificación y la efectividad en la categoría y servicio correspondiente, lo que permitirá la organización efectiva en el flujo de trabajo de TICS, mientras que en el tiempo se evaluará la recepción de la solicitud. Estas métricas ayudarán a determinar la eficiencia del sistema en el tiempo y gestión de solicitudes.

1.6.4 Análisis de recolección de datos

En este proyecto, la recolección de datos se enfoca en la selección y evaluación de algoritmos de Machine Learning con modelos de aprendizaje supervisado para la clasificación de solicitudes. El objetivo es determinar el rendimiento de cada algoritmo en términos de Precisión, Accuracy, Recall y F1 Score, implementando técnicas de procesamiento de lenguaje natural (NLP), para transformar dichos datos en características cuantificables. En este análisis se empleará un conjunto de datos relacionados con registros de solicitudes de servicios de TI, que serán limpiados, procesados, luego ser entrenados y seleccionados para validar el modelo.

La población y muestra se definirán de la siguiente manera:

Población:

Dentro del entorno o campo de Machine Learning, existe gran variedad de algoritmos diseñados para clasificación de textos, estos algoritmos están agrupados en diferentes categorías según su enfoque, naturaleza y el tipo de aprendizaje. En este estudio, nos centramos en algoritmos de aprendizaje supervisados, que aprenden de ejemplos etiquetados. Los principales algoritmos de la población de estudio son:

Supervisados:

- ✓ Neuronal Networks.
- ✓ Naive Bayes.

- ✓ Linear Regression.
- ✓ Logistic Regression.
- ✓ K-Nearest Neighbor.
- ✓ Support Vector Machines (SVM).
- ✓ Random Forest

Muestra:

Para este estudio, se seleccionaron cinco algoritmos de Machine Learning, las cuales serán entrenadas y se evaluará su desempeño en la clasificación de solicitudes de gestión de TI, las cuales tenemos:

Random Forest: Este será evaluado por su capacidad de clasificar y resistencia en el sobreajuste, es ideal para interactuar con datos de características ruidosas.

Support Vector Machine (SVM): Para investigar su eficacia en la categorización de peticiones de diversas clases, en contextos con propiedades de alta dimensión.

Naive Bayes: Este modelo será valorado por su sencillez y eficacia en la categorización de texto, teniendo en cuenta su desempeño en datos, lo que significa que gestiona grandes cantidades de texto y resulta eficaz para problemas de clasificación binaria.

Logistic Regression: Este modelo se valorará en función de su habilidad para categorizar y su desempeño en problemas de diversas clases.

K- Nearest Neighbor (KNN): Se evaluará por su simplicidad y capacidad para manejar problemas de clasificación con pocos datos etiquetados, además de su rendimiento en entornos donde las clases están bien definidas.

1.7 Metodología de desarrollo

El desarrollo del proyecto actual se basará en la metodología de ITIL v4, enfocándose el proceso de Gestión de solicitudes de servicios, con la finalidad aumentar la eficiencia y efectividad en la gestión de servicios de TI a través de la automatización en la clasificación de solicitudes de usuarios, el mismo que se llevará a cabo de la siguiente manera [19].

- **Identificación:** Se identificarán los procesos clave de gestión de solicitudes y se definirán las necesidades del sistema de clasificación automatizada. Se analizarán datos para entender las demandas y desafíos presentes. Además, Se revisan los principios de ITIL v4 y se analiza la factibilidad del uso de Machine Learning para luego adaptarlas al chatbot.
- **Planificación:** A partir de la información ya realizada por una revisión bibliográfica, se analizará el estado actual de la gestión de servicios de TI de diversas empresas u organizaciones, este enfoque ayudará a identificar y crear un plan estratégico para realizar el sistema de clasificación automatizada de solicitudes, mediante el entrenamiento de un modelo de ML y creación del chatbot.
- **Diseño:** Para cumplir el funcionamiento del chatbot, primero se realizará una limpieza profunda de los datos (Dataset), seguida se entrenará los modelos supervisados de Machine Learning. Se establece la arquitectura del sistema y se definen las interfaces de usuario, se planifican pruebas para garantizar el funcionamiento y compatibilidad del sistema.
- **Control y gestión:** Se dará un proceso de monitoreo y seguimiento al chatbot, en la interacción con el usuario. El rendimiento del modelo de Machine Learning será evaluado y monitoreado y se harán ajustes cuando sea necesario para mejorar la precisión en la clasificación de solicitudes.
- **Ejecución:** Finalmente, se llevará a cabo el sistema de administración de peticiones (chatbot), comprobando y logrando los objetivos propuestos. Los resultados alcanzados se registrarán y se examinarán para efectuar mejoras futuras en el sistema.

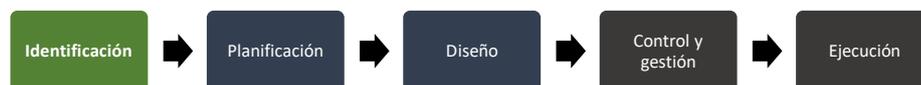


Ilustración 1 Metodología del desarrollo del proyecto

CAPÍTULO 2. PROPUESTA

2.1 Marco Contextual

Una gobernación es una entidad gubernamental local responsable de administrar y coordinar distintas políticas y programas en beneficio de la comunidad. Su tarea primordial es asegurar el progreso social y económico de la zona, fomentando el bienestar de los habitantes. Esto incluye la implementación de iniciativas que fomenten la inclusión social, la atención a poblaciones vulnerables y la mejora de la calidad de vida de los habitantes [20]. Las gobernaciones también juegan un rol fundamental en la gestión de servicios públicos, el fomento de la educación, la salud y la seguridad, además de la administración del medio ambiente y la organización del territorio. Además, tienen la tarea de alinear esfuerzos con otras instituciones gubernamentales y entidades comunitarias para tratar problemas locales, tales como la pobreza, la marginación social y la discriminación. Mediante su labor, aspiran a generar un ambiente más equitativo y justo para todos los ciudadanos, en particular para aquellos que están en circunstancias de vulnerabilidad [20].

Este proyecto satisface la demanda de una transformación digital en la gestión pública, catalogada en la bibliografía como un instrumento esencial para incrementar la eficacia en la administración de servicios [21]. Mediante revisiones bibliográficas, se revela cómo las tecnologías, el Aprendizaje Automático, se están implementando en diversas instituciones, gobiernos y organismos gubernamentales a nivel mundial. Estas investigaciones evidencian que la automatización en la gestión de peticiones optimiza recursos, disminuye los tiempos de respuesta y potencia la exactitud en la distribución de tareas, lo que conduce a una gestión más eficaz y ágil.

Visión

Ser la entidad pública líder en impulsar la inclusión social de los grupos prioritarios y de quienes se encuentran en situación de pobreza, promoviendo de manera efectiva su avance en la escala social [22].

Misión

Formular y ejecutar políticas, normativas, programas y servicios que favorezcan la inclusión social y brinden atención integral a lo largo del ciclo de vida de niños, niñas, adolescentes, jóvenes, personas mayores, personas con discapacidad y aquellos en condiciones de pobreza, con el fin de contribuir a su progreso social y ayudarles a superar la pobreza [22].

2.2 Marco Conceptual

2.2.1 ITIL (Information Technology Infrastructure Library)

Es un esquema de mejores prácticas de Service Management, es una metodología altamente valorada que proporciona un conjunto de prácticas y procesos para mejorar la planificación, diseño, entrega y asistencia de los servicios de Tecnología de la Información.. El principal beneficio es mejorar y potenciar la calidad de los servicios, los procedimientos y actividades en diversas organizaciones y empresas [23].

2.2.2 ITIL v4

Es una guía que necesitan las organizaciones para afrontar nuevos desafíos en entorno a la administración de servicios, aprovechando el uso de la tecnología emergentes y modernas. Garantiza un sistema ágil para gestionar servicios de TI, facilitando a las organizaciones a optimizar sus procesos [24]. Además, ITIL v4 incluye 14 prácticas de administración general, 17 de administración de servicios y 3 de administración técnica. [25].

2.2.3 Gestión de solicitudes de servicio.

La gestión de solicitudes de servicio es implicar la generación, renovación, escalabilidad y solución de peticiones de servicio en un lugar específico de su compañía o entidad. que facilita a los usuarios el envío de peticiones, demandas o consultas. Este procedimiento comprende desde la recepción y categorización de la petición hasta su resolución definitiva, garantizando que las solicitudes sean administradas de forma eficaz y a tiempo [26].

Las solicitudes de servicio son, por otro lado, un elemento regular de la provisión de servicios y no constituyen un fallo o una disminución en la calidad tal como son

los incidentes. Por lo tanto, dado que la solicitud de servicio generalmente ya está definida en cuanto a su origen y alcance, así como acordada por las partes como parte de la operación regular, esto puede generalmente estructurarse a través de un procedimiento estándar bien definido para iniciar, aprobar, ejecutar y gestionar.

Las fases de solicitud de servicio se basan en procesos y procedimientos bien definidos, los cuales se implementan a través de herramientas de seguimiento y automatización de la gestión de solicitudes para incrementar la eficiencia y eficacia. Puede haber distintos tipos de solicitud que demanden flujos de trabajo distintos, pero la eficiencia y el mantenimiento se incrementan limitando los modelos de flujo de trabajo. Al agregar una solicitud a nuestro catálogo de servicios, se puede usar un modelo de flujos de trabajos existentes, o definir uno nuevo.

En la ilustración 2, se muestra la contribución de la gestión de solicitudes de servicio a la cadena de valor del servicio. La gestión de solicitudes de servicio contribuye significativamente a la cadena de valor de un proceso de servicio humano al cumplir con la medida en que todas las actividades de la cadena involucran la práctica [19]. Esto incluye el siguiente:

- Mejora: La gestión de solicitudes de servicio puede desempeñar un papel significativo al actuar como un canal para la implementación de cualquier iniciativa de mejora, así como para tratar con quejas y cumplidos del usuario. También contribuye a la mejora en un nivel más restringido al proporcionar información sobre tendencias, calidad y retroalimentación relacionada con la implementación de dichas solicitudes.
- Comprometerse: La gestión de solicitudes conlleva un diálogo constante para recolectar necesidades de cada usuario, definiendo expectativas y ofreciendo actualizaciones sobre el estado de las solicitudes.
- Diseño y transición: Los componentes de servicios estándar pueden ser implementados al entorno en vivo a través del cumplimiento de solicitudes de servicio.
- Obtener / construir: La adquisición de componentes de servicios preaprobados puede llevarse a cabo mediante solicitudes de servicio.

- Entrega y apoyo: La gestión de solicitudes desempeña un papel importante en la prestación normal de servicios. Esta actividad de cadena de valor se centra en asegurar que los usuarios sigan siendo productivos, a menudo dependiendo del cumplimiento efectivo de sus solicitudes.

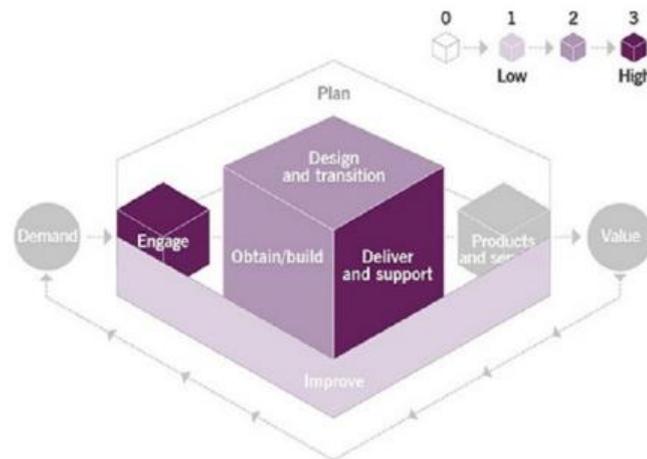


Ilustración 2 Mapa de calor de la contribución de la gestión de solicitudes de servicio al valor [19].

2.2.4 Gestión de catálogos de servicios

En la lustración 3. La gestión de catálogos de servicios incluye un conjunto de tareas constantes vinculadas con la divulgación, renovación y conservación de las descripciones de productos, servicios y sus correspondientes propuestas [19]. Esta práctica ofrece una perspectiva precisa acerca del alcance de los servicios disponibles, además de las condiciones en las que se brindan, está respaldada por roles clave, como el propietario del servicio y otros encargados de mantener la lista de servicios actualizada, ajustándola conforme se añadan, alteren o eliminen [19]. Ejemplos que incluyen:

- Vistas de usuario.
- Opiniones de clientes.
- Puntos de vista de los clientes de TI a TI.

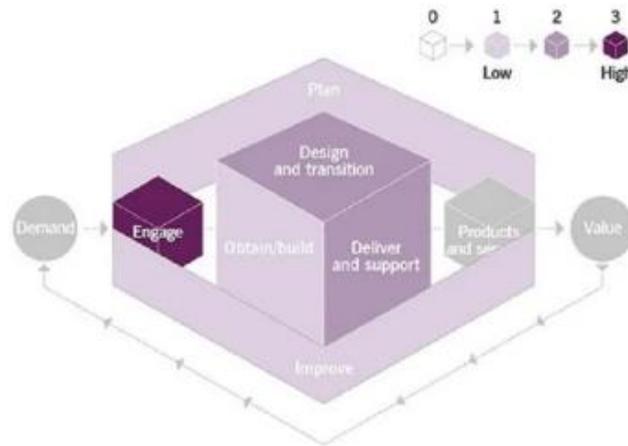


Ilustración 3 Mapa de calor de la contribución de la gestión del catálogo de servicios al valor [19].

2.2.5 Machine Learning

Es una rama de la inteligencia artificial, esta permite que diversas máquinas o tecnologías aprendan ciertas tareas para su funcionamiento, Machine Learning es muy ágil en el reconocimiento de patrones, realizar predicciones, en base al procesamiento de grandes datos, y mejora constantemente a medida que se le proporcionan más datos, lo que le permite optimizar sus resultados y tomar decisiones informadas de manera autónoma [27].

2.2.6 Zammad

Es un software de soporte de código abierto, que sirve para la gestión de tickets, ofrece y facilita la comunicación del usuario a través de diferentes medios o canales como correo electrónico, chat, Telegram, WhatsApp, Facebook, entre otros [8]. Su interfaz y capacidad permiten la integración con otras herramientas facilitando una solución eficaz para la administración y seguimiento de tickets, mejorando la eficiencia y la experiencia del usuario [8].

2.2.7 PostgreSQL

Es un servidor de base de datos relacional de objetos gratuito con funciones gratuitas de orientación a objetos, que abarca herencia, tipos de datos, funciones y restricciones, la licencia BSD describe los factores desencadenantes, las regulaciones y la integridad tradicional, como muchos otros proyectos de open

source, el desarrollo de PostgreSQL no está gestionado para una única empresa, sino para una comunidad de desarrolladores y entidades comerciales, dicha comunidad se le denomina como PGDG (PostgreSQL Global Development Group) [28].

2.2.8 Telegram

Es una aplicación de mensajería gratis, rápida y segura, basada en la nube que funciona en diversas plataformas y se la puede usar en diferentes partes del mundo. Posee chats cifrados y pueden realizar videos llamadas. Además, Telegram facilita una API para integrar un chatbot y agilizar los procesos con otras herramientas, es una herramienta que favorece en la clasificación de gestión de solicitudes [29].

2.2.9 TypeScript

Se trata de un lenguaje de programación de código abierto, o sea, de acceso y uso libre, diseñado para aplicaciones y ejecuciones en proyectos, nos brinda la posibilidad de trabajar de forma organizada. También se le reconoce como un lenguaje de alto nivel. Los beneficios de TypeScript son su mayor agilidad en la identificación de errores y fallos en los códigos, su sintaxis es sumamente intuitiva, proporciona bibliotecas en JavaScript y la documentación de la API, se puede ejecutar en cualquier navegador o dispositivo, facilidad en el trabajo para desarrolladores, permite el trabajo con Angular, jQuery, MongoDB y React. [10].

2.2.10 Postman

Una plataforma gratuita que permite el desarrollo y pruebas de APIs. Es muy útil para programar porque da la posibilidad realizar pruebas, crear y enviar solicitudes en HTTP a cualquier API mediante una interfaz gráfica. También nos permite comprobar el correcto funcionamiento de proyectos, es fácil de usar y da esa posibilidad de crear documentación de todo lo que has desarrollado [11].

2.2.11 Express.js

Es un framework de backend para Node.js esta permite estructurar una aplicación de una manera ágil, nos proporciona funcionalidades como el enrutamiento, opciones para gestionar sesiones, proporciona características y herramientas robustas en backend escalables, también brinda herramientas de interfaz mediante

comandos (CLI) llamada Node Package Manager (NPM), donde los desarrolladores puedan usar paquetes desarrollados [30].

2.2.12 Python

Es un lenguaje de programación y de alto nivel, ampliamente utilizado en las aplicaciones o desarrollo web, software, ciencia de datos y el Machine Learning (ML), cuyo uso para los desarrolladores es eficiente y fácil de interactuar, se puede ejecutar en diferentes plataformas. Este software es gratis y se integra a todo tipo de sistema aumentando así la velocidad del desarrollo [13].

2.2.13 Grafana.

Plataforma interactiva y dinámica open source, permite el almacenamiento, visualización y el análisis de métricas e información en tiempo real de los datos. Se utiliza principalmente para monitorizar grandes cantidades de datos y presentándolos en gráficos [14]. Asimismo, tiene un sistema de alertas para que estemos pendiente de lo que sucede con la infraestructura y reaccionar ante estos posibles problemas [14].

2.2.14 Visual Studio Code.

Esta herramienta desarrollada por Microsoft, uso de instalación ligero que sirve para ver, editar, ejecutar y depurar códigos de aplicaciones o proyectos, es supe, brinda soporte en diferentes lenguajes de programación, detección de fragmentos de códigos, compatible para Windows, Linux o Mac. Visual Studio Code proporciona autocompletado inteligente, es decir, sugiere automáticamente fragmentos de códigos, variables, funciones y otros métodos al momento en que escribas, esto reduce errores y ayuda a que escribas más rápido y de manera precisa [31].

2.2.15 Elasticsearch.

Elasticsearch es un motor de búsqueda y análisis distribuido, escalable y adaptable, que actúa como un almacén central de datos para búsquedas rápidas y análisis profundos, es parte esencial del Elastic Stack, ofreciendo soluciones desde monitoreo en tiempo real hasta inteligencia corporativa, permitiendo a las organizaciones extraer insights valiosos de grandes volúmenes de datos [16].

2.3 Marco Teórico

2.3.1 Implementación de las buenas prácticas de ITIL v4: Mejorar la gestión de solicitudes.

ITIL “Information Technology Infrastructure Library”, fue desarrollada a finales de 1980, y actualmente se ha convertido en un estándar a nivel mundial para las organizaciones en la Gestión de Servicios de TI, ITIL v1 GITM “Infraestructura de Tecnologías de información del Gobierno”, obtuvo 31 libros [32]. En el año 2000 se lanza ITIL v2, que disminuye la gran cantidad de libros agrupándolos de forma lógica, en 2007 se lanza ITIL v3 con modificaciones más significativas y en 2019 se lanza ITIL v4 donde hubo cambios, esta se enfoca en el valor del servicio y ver el ciclo de vida del servicio [32]. Durante su inicio hasta ahora se ha convertido en el marco de referencia de buenas prácticas y admitido por la Gestión de TI en todas partes.

Juan Camilo, menciona que fomentar las buenas prácticas de ITIL trae beneficios a la organización, mientras más robusta sea la aplicación de las prácticas esta reduce los costos en la organización, el objetivo es trazar una línea del estado actual del Service Desk de la Universidad Santo Domingo Sede Bogotá, el departamento de TI aplica un proceso de ITIL v3 y la finalidad de este seminario es que Service Desk cuente con la última versión de ITIL v4 para la gestión de solicitudes, proporcionando a los administradores de sistema de TI el uso de herramientas y documentos que ofrezcan una mejora en la calidad de sus servicios y la optimización del Service Desk [32].

2.3.2 Aplicación de Machine Learning: Modelos para clasificación de solicitudes.

En el siguiente trabajo de titulación, Paullo Montes menciona que las empresas están obligadas a adaptarse a diversos cambios por el continuo desarrollo de las tecnologías de la información. El objetivo es la examinación del área de Service Desk de las empresas y la manera de atención de tickets, se plantea desarrollar un modelo de aprendizaje supervisado que clasifique automáticamente los tickets registrados por los clientes (solicitudes o problemas), el algoritmo reducirá el

trabajo de los humanos, es decir que optimizará errores y ofrecerá un servicio de calidad y aumentar la satisfacción del usuario final [33].

2.3.3 Mejoras en la experiencia de usuario: Análisis del servicio de Help desk Zammad.

En este proyecto, Adrián José, hace un análisis del sistema help desk Zammad para mejorar la eficiencia operativa y la experiencia de usuarios en el gobierno parroquial de Pimocha, permitiendo centralizar y automatizar la gestión de solicitudes, con Zammad se puede administrar el seguimiento de pedidos de los ciudadanos, con esta plataforma, las solicitudes de los usuarios de registran de manera digital, precisando su ingreso hasta su resolución, con esto se elimina la necesidad de manejar varias herramientas y documentos físicos, reduciendo errores y duplicaciones de tareas. Además, Zammad genera notificaciones para que las solicitudes no se pasen por alto, también permite generar informes detallados sobre el rendimiento de los servicios [34].

2.4 Requerimientos

2.4.1 Requerimientos Funcionales

MÓDULO DE INICIO DE SESIÓN

Código	Descripción
RF-1	El chatbot le brindará un saludo inicial al usuario, le da un breve detalle de lo que debe realizar.
RF-2	El chatbot le mostrará el menú al usuario, donde debe elegir tres opciones: 1. Enviar solicitud., 2. Ver detalles de Tickets., 3. Ver respuesta de soporte., 4. Salir o cerrar.
RF-3	El chatbot deberá permitir el registro de la solicitud, solo recepción de

	solicitudes deberá crear un ticket y clasificar las solicitudes de los usuarios.
RF-4	El chatbot deberá confirmar al usuario una vez que la solicitud se haya registrado con éxito y el estado actual de la solicitud
RF-5	El chatbot deberá manejar solicitudes no validas, ofreciendo una retroalimentación al usuario.

Tabla 1 Requerimientos Funcionales – Módulo de inicio de sesión.

MÓDULO DE REGISTRO DE USUARIOS.

Código	Descripción
RF-6	El administrador mediante Zammad puede registrar un nuevo usuario, deberá pedir los datos para su registro.
RF-7	El administrador deberá solicitar que el usuario le otorgue sus datos para su registro antes de proceder a enviar la solicitud: <ul style="list-style-type: none"> ✓ Nombre. ✓ Apellido. ✓ Cédula. ✓ Correo electrónico.
RF-8	El administrador procede a comunicar al usuario que ya puede dar uso del chatbot en Telegram.

Tabla 2 Requerimientos Funcionales – Módulo de registro de usuarios.

MÓDULO DE CLASIFICACIÓN DE SOLICITUDES.

Código	Descripción
RF-9	El chatbot deberá clasificar las solicitudes recibidas de manera automática utilizando el modelo de Machine Learning entrenado.
RF-10	El chatbot deberá categorizar cada una de las solicitudes, facilitando su revisión por el personal de TICS o administrador.
RF-11	Los administradores deberán tener acceso para ver las solicitudes y asignarlas a agentes para su resolución.

Tabla 3 Requerimientos Funcionales – Módulo de clasificación de solicitudes.

MÓDULO DE SEGUIMIENTO DE SOLICITUDES

Código	Descripción
RF-12	El asistente virtual permitirá al usuario consultar el estado de las solicitudes y tickets.
RF-13	El asistente virtual notificará al usuario sobre cambios en el estado de su solicitud.
RF-14	El asistente virtual permitirá al usuario administrador acceder a Zammad para dar seguimiento a las solicitudes.

Tabla 4 Requerimientos Funcionales – Módulo de seguimiento de solicitudes.

MÓDULO DE RESOLUCION DE SOLICITUDES.

Código	Descripción
RF-15	El asistente virtual permitirá el acceso a los administradores para revisar y dar soluciones a las solicitudes clasificadas.
RF-16	El asistente virtual permitirá a los administradores proporcionar una solución textual a cada solicitud.
RF-17	El asistente virtual notificará al usuario una vez que su solicitud haya sido resuelta.

Tabla 5 Requerimientos Funcionales – Módulo de resolución de solicitudes.

MÓDULO DE SEGURIDAD

Código	Descripción
RF-18	El sistema para los usuarios cuenta con un login de autenticación, que el usuario debe ingresar su número de cédula y si está registrado, el chatbot permite el ingreso.
RF-19	Los perfiles de cada usuario estarán restringidos, cada información dependerá de los usuarios, administradores.
RF-20	El asistente virtual no tiene acceso a realizar modificaciones de los datos

	personales, solo los administradores pueden realizar cambios.
--	---

Tabla 6 Requerimientos Funcionales – Módulo de seguridad.

2.4.2 Requerimientos no Funcionales

Código	Descripción
RNF-1	El asistente virtual deberá ser accesible desde cualquier dispositivo o navegador.
RNF-2	Los datos de los usuarios y de las solicitudes estarán protegidas mediante medidas de seguridad que ofrece Zammad.
RNF-3	Los datos de las solicitudes, tickets generados se guardarán en la tabla de tickets y requests de Zammad al momento de ser clasificadas automáticamente.
RNF-4	Los datos de los tickets se actualizarán cada vez que el administrador o TICS realice cambios en Zammad.
RNF-5	Las contraseñas de los usuarios al momento de su registro son encriptadas en la base de datos.

Tabla 7 Requerimientos no funcionales.

2.5 Componentes de la propuesta

2.5.1 Arquitectura del sistema

La ilustración 4, muestra la estructura del sistema del proyecto, donde el usuario inicia enviando una solicitud mediante Telegram, dicha solicitud se dirige al módulo de Machine Learning, donde la solicitud es clasificada automáticamente mediante un modelo previamente ya entrenado, esta clasificación mediante el modelo le permite asignar una categoría específica que facilitará la gestión por parte del departamento de TI. Una vez que la solicitud es clasificada en la base de datos en Zammad. Zammad es una gestión de crear tickets y dar seguimiento a las solicitudes. Zammad se conecta con una base de datos que lleva conexión con PostgreSQL, donde se almacenan todos los tickets y las solicitudes, dando un acceso eficiente a la información por parte de los administradores o agentes de TI. También se les notificará a los usuarios manteniéndolos informados sobre el estado de sus solicitudes. Después de que la solicitud sea procesada en Zammad, se les envía una notificación a los usuarios a través de Telegram indicándoles el estado de las solicitudes, manteniéndolos informados sobre el seguimiento del progreso de su solicitud hasta su cierre. Además, la integración con Elasticsearch y Grafana es para tener una visualización y un análisis de datos. Elasticsearch facilita el análisis de los datos registrados en Zammad y PostgreSQL en tiempo real, mientras que Grafana permite crear gráficos, estadísticas, ambos permiten a los administradores o al departamento de TI a monitorear el rendimiento y el estado de las solicitudes de manera rápida y efectiva.

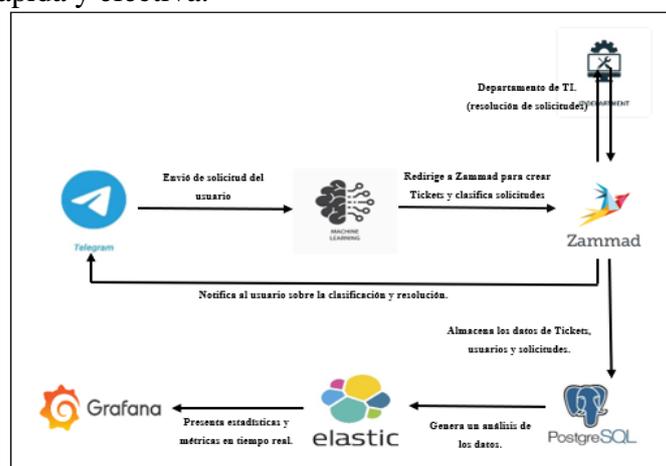


Ilustración 4 Arquitectura del sistema del proyecto.

2.5.2 Proceso de las solicitudes

En la ilustración 5. Se observa el proceso que lleva el chatbot al procesar las solicitudes, este proceso inicia con el ingreso de su credencial para el acceso, el usuario al describir su solicitud, esta pasa a ser clasificado automáticamente mediante la técnica ya entrenada, se le asigna el servicio y la categoría correspondiente. Luego, pasa a crearse un ticket en Zammad junto a dicha solicitud, donde el administrador o el encargado del departamento de TICS darán seguimiento y resolución a la solicitud, al mismo tiempo se guardarán en la base de datos pgAdmin.

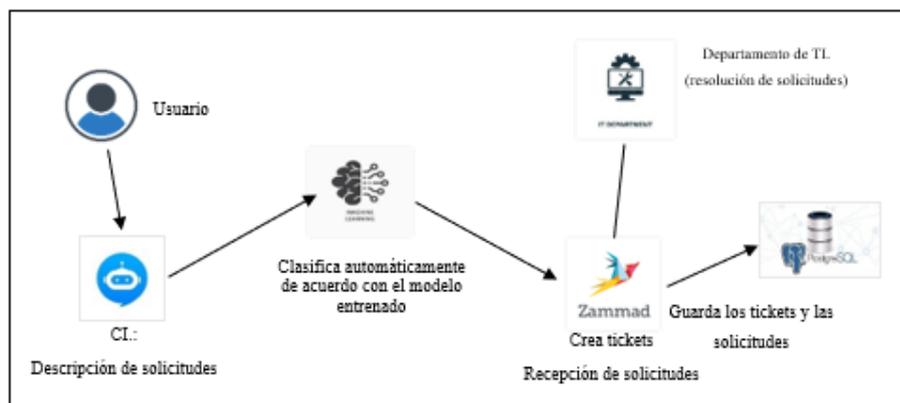


Ilustración 5 Diagrama del proceso de solicitudes

2.5.3 Diseño del catálogo de solicitudes de servicios

El objetivo del diseño del catálogo de peticiones de servicios es estructurar y organizar de forma clara y organizada los servicios que están disponibles para el usuario de la gobernación. Este catálogo clasifica las peticiones de acuerdo a su descripción, ofreciendo un breve detalle de cada servicio para simplificar su reconocimiento, administración y acceso como muestra la ilustración 6. Cada solicitud contiene un código exclusivo, la categoría del servicio, el tipo de petición y una descripción breve que establece el rango y las ventajas del servicio. Esta estructura facilita una administración eficaz y garantiza que los usuarios tengan acceso a los servicios pertinentes de forma rápida y clara.

Categoría	Servicio
<i>Telecomunicaciones</i>	<i>Correo Electrónico</i>
	<i>Portal Web</i>
	<i>Conexión</i>
	<i>Gestión de Usuarios</i>
	<i>Acceso por VPN</i>
	<i>Redes Sociales</i>
<i>Seguridad</i>	<i>Antivirus</i>
	<i>Filtro Web</i>
<i>Gestión de Servidores</i>	<i>Cloud</i>
<i>Gestión de Datos</i>	<i>Base de Datos</i>
<i>Innovación Tecnológica</i>	<i>Implementación de Tecnologías Emergentes</i>
<i>Soporte Técnico</i>	<i>Aplicaciones</i>
	<i>Computador</i>
	<i>Impresora</i>
<i>Formación y Capacitación</i>	<i>Talleres de Capacitación</i>

Ilustración 6 Catálogo de solicitudes.

2.5.4 Proceso de limpieza de datos

Este proceso es fundamental en el desarrollo del entrenamiento de técnicas de Machine Learning, ya que la adecuación de los datos tiene una repercusión directa en el rendimiento del modelo, para este proyecto se ha recopilado varios conjuntos de datos con registros de tickets y solicitudes, estos datos se obtuvieron de diferentes fuentes externas: GitHub, Kaggle Datasets, Datasetsearch y se han generados datos sintéticos.

GitHub		
Requests classification using ML and NLP	https://github.com/BioAITeam/Requests-classification-using-ML-and-NLP	10377 datos
Kaggle Datasets		
Service_Requests_from_2010	https://www.kaggle.com/datasets/sherinaclaudia/nyc311-2010	364559 datos
Datasetsearch		
Solicitudes de Jefatura de Gobierno	https://datosabiertos.infocdmx.org.mx/dataset/solicitudes-de-informacion-2023	138417 datos

Listado de Solicitudes a la Mesa de Servicio de TI	https://www.datos.gov.co/d/ihwm-pyi4	11941datos
---	---	------------

Tabla 8 Datos Externos

2.5.4.1 Limpieza y depuración de los datos:

En el gráfico nos muestra el proceso de limpieza y depuración de datos en Python.

- Eliminación de datos duplicados: Quita las filas duplicadas del dataset.
- Valores faltantes: Elimina filas donde faltan valores.
- Limpieza de texto: Transforma todo el texto en minúsculas, suprime caracteres especiales y números, remueve los espacios vacíos.
- Eliminación de Stopwords: Filtra esas palabras vacías que no tienen importancia o que no aportan un valor significativo al análisis.
- Lematización: Convierte o reduce esas palabras en su forma original, es decir, para normalizar el texto.
- Tokenización: Esta separa el texto de las descripciones en palabras individuales.

```

import re
import spacy
import unicodedata
import pandas as pd
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')
stop_words = set(stopwords.words('spanish'))
nlp = spacy.load("es_core_news_sm")

file_path = './Solicitudes Original.csv'
data = pd.read_csv(file_path, delimiter=';')
data = data.drop_duplicates()
data = data.dropna(subset=['Descripcion', 'Servicio', 'Categoria'])
def preprocess_text(text):
    if isinstance(text, str):
        text = text.lower()
        text = ''.join((c for c in unicodedata.normalize('NFD', text) if unicodedata.category(c) != 'Mn'))
        text = re.sub(r'[^\w\s]', '', text)
        doc = nlp(text)
        tokens = [token.lemma_ for token in doc if token.text not in stop_words and token.is_alpha]
        text = ' '.join(tokens)
    return text
data['Descripcion'] = data['Descripcion'].apply(preprocess_text)
final_data = data[['Descripcion', 'Servicio', 'Categoria']].copy()
cleaned_file_path = './datoslimpios.csv'
final_data.to_csv(cleaned_file_path, index=False, sep=';', encoding='utf-8-sig')
print("Limpieza de datos completada. Guardado en:", cleaned_file_path)

```

Ilustración 7 Limpieza y depuración de datos.

2.5.5 Entrenamiento de modelos de aprendizaje supervisados

Los modelos supervisados tales como: Logistic Regression, Support Vector Machine (SVM), K- Nearest Neighbor (KNN), Random Forest y Naive Bayes. El entrenamiento está conformado por una serie de etapas para que el modelo adquiera

conocimientos de forma más eficaz. Una de las tareas implica la preparación de datos, que se segmenta en dos secciones: conjuntos de entrenamiento con un 80% y un 20% para prueba, y la optimización del modelo mediante métodos de validación cruzada.

Los datos ya limpiados pasan nuevamente por una limpieza y normalización. Este proceso consta en la eliminación de caracteres no deseados y palabras irrelevantes. Luego, los datos de las descripciones se convierten en representaciones numéricas mediante la vectorización con el Vectorizer o TF-IDF, en la que el modelo pueda procesar y analizar.

Después, se separan los datos para el entrenamiento en un 80% y para la prueba en un 20%, asegurando que el modelo pueda extender sus predicciones a datos desconocidos, se lleva a cabo una validación cruzada, en la que los datos de entrenamiento se segmentan en diversas particiones, lo que posibilita la evaluación del modelo en diferentes combinaciones de subconjuntos y de esta manera prevenir el sobreajuste.

Luego de que los datos sean calibrados, el modelo se capacita con el conjunto de entrenamiento y se evalúa con el conjunto de prueba mediante métricas de Precisión, Recall, Accuracy y F1-score, que ayudan a evaluar la balanza entre las predicciones correctas y las erróneas en las distintas clases. Finalmente, se elabora una matriz de confusión para mostrar los logros y fallos, identificando áreas de mejora y ofreciendo datos sobre cómo ajustar el modelo para aplicaciones específicas. Cuando el modelo alcanza un desempeño adecuado, puede ser utilizado en la producción y monitoreado de forma continua para modificar o reentrenar si el rendimiento se sobrepasa debido a la introducción de nuevos datos.

2.5.4.2 Proceso de selección del modelo y entrenamiento de datos:

Modelo Support Vector Machine (SVM).

Para el proceso de entrenamiento de datos como vemos en la ilustración 8, se refiere al primer modelo SVM, en primer lugar debemos importar las librerías correspondientes para procesar texto, transformar datos, construir y evaluar el modelo, cargamos el modelo en lenguaje español con 'spaCy', convierte el texto en

minúsculas, elimina los acentos con ‘unicodedata.normalize’, quita caracteres no alfabéticos, elimina palabras vacías con ‘stop words’, se carga el archivo ‘datoslimpios.csv’, se aplica la función ‘preprocess_text’ para la columna descripción para limpiarla y preparar el texto, y define las variables para los modelos $X = data['Descripción']$, $y_{servicio} = data['Servicio']$ y $y_{categoria} = data['Categoria']$.

```

import re
import spacy
import unicodedata
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import joblib

nlp = spacy.load("es_core_news_sm")
def preprocess_text(text):
    if isinstance(text, str):
        text = text.lower()
        text = ''.join((c for c in unicodedata.normalize('NFD', text) if unicodedata.category(c) != 'Mn'))
        text = re.sub(r'[^\w-zA-Z\s]', '', text)
        doc = nlp(text)
        tokens = [token.text for token in doc if not token.is_stop and token.text.isalpha()]
        text = ' '.join(tokens)
    return text
file_path = './datoslimpios.csv'
data = pd.read_csv(file_path, delimiter=',')
data['Descripcion'] = data['Descripcion'].apply(preprocess_text)
X = data['Descripcion']
y_servicio = data['Servicio']
y_categoria = data['Categoria']

```

Ilustración 8 Proceso 1 del entrenamiento del modelo SVM.

Después seguimos con el proceso de entrenamiento como nos refleja en la ilustración 9, se utiliza ‘LabelEncoder’ para convertir las etiquetas de texto ‘ $y_{servicio}$ y $y_{categoria}$ ’ en valores numéricos, esto nos facilita el entrenamiento del modelo. Luego se dividen los datos en entrenamiento 80% y de prueba 20%, se implementa ‘ $y_{train_servicio}$ y $y_{test_servicio}$ ’ y ‘ $y_{train_categoria}$ y $y_{test_categoria}$ ’, esto es tanto para el modelo de servicio y categoría.

```

encoder_servicio = LabelEncoder()
y_servicio_encoded = encoder_servicio.fit_transform(y_servicio)
encoder_categoria = LabelEncoder()
y_categoria_encoded = encoder_categoria.fit_transform(y_categoria)

X_train, X_test, y_train_servicio, y_test_servicio = train_test_split(X, y_servicio_encoded, test_size=0.2, random_state=42)
_, _, y_train_categoria, y_test_categoria = train_test_split(X, y_categoria_encoded, test_size=0.2, random_state=42)

```

Ilustración 9 Proceso 2 del entrenamiento del modelo SVM.

Se procede a crear y entrenar el modelo SVM para el servicio y categoría como vemos en la ilustración 10, se define un pipeline de ‘TfidfVectorizer’ y

clasificador SVC con kernel lineal y se entrena el modelo usando 'X_train y y_train_servicio' y 'X_train y y_train_categoria' para ambos.

```
# Entrenamiento para servicio
svm_classifier_servicio = Pipeline([('tfidf', TfidfVectorizer()), ('svm', SVC(kernel='linear', probability=True))]
svm_classifier_servicio.fit(X_train, y_train_servicio)
# Entrenamiento para categoría
svm_classifier_categoria = Pipeline([('tfidf', TfidfVectorizer()), ('svm', SVC(kernel='linear', probability=True))]
svm_classifier_categoria.fit(X_train, y_train_categoria)
```

Ilustración 10 Proceso 3 del entrenamiento del modelo SVM.

Se evalúa el modelo para el servicio y para la categoría, se observa en la ilustración 11, 'y_pred_servicio', 'y_pred_categoria' se usa para la predicción en el conjunto de prueba, se calcula la precisión 'accuracy_score' comparando 'y_test_servicio y y_pred_servicio', 'y_test_categoria y y_pred_categoria' y se genera un informe con métricas de evaluación 'precisión, recall, f1-score y support' para cada clase. También se genera la matriz de confusión para la evaluar el rendimiento del modelo para cada clase.

```
# Evaluación y matriz de confusión para servicio
y_pred_servicio = svm_classifier_servicio.predict(X_test)
accuracy_servicio = accuracy_score(y_test_servicio, y_pred_servicio)
print(f'Precisión del modelo de servicio: {accuracy_servicio:.2f}')
print('Informe de clasificación para servicio:')
print(classification_report(y_test_servicio, y_pred_servicio))

conf_matrix_servicio = confusion_matrix(y_test_servicio, y_pred_servicio)
print('Matriz de confusión para servicio:')
print(conf_matrix_servicio)

# Evaluación y matriz de confusión para categoría
y_pred_categoria = svm_classifier_categoria.predict(X_test)
accuracy_categoria = accuracy_score(y_test_categoria, y_pred_categoria)
print(f'Precisión del modelo de categoría: {accuracy_categoria:.2f}')
print('Informe de clasificación para categoría:')
print(classification_report(y_test_categoria, y_pred_categoria))

conf_matrix_categoria = confusion_matrix(y_test_categoria, y_pred_categoria)
print('Matriz de confusión para categoría:')
print(conf_matrix_categoria)
```

Ilustración 11 Proceso 4 del entrenamiento del modelo SVM.

Resultados del modelo Support Vector Machine (SVM), ver ilustración 12:

El modelo SVM exhibe un desempeño sobresaliente en precisión y recall para las clasificaciones de servicio y categoría, con un F1-score balanceado, lo que señala una excelente mezcla entre las dos con una precisión de 97% y 95%.

Evaluación del clasificador de Servicio: Precisión: 0.9691119691119691					Evaluación del clasificador de Categoría: Precisión: 0.9510939510939511				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Acceso por VPN	1.00	0.98	0.99	55	Formación y Capacitación	0.85	0.96	0.90	48
Antivirus	1.00	0.96	0.98	45	Gestión de Datos	0.92	0.96	0.94	49
Aplicaciones	0.95	1.00	0.97	39	Gestión de Servidores	1.00	0.97	0.98	65
Base de Datos	0.98	1.00	0.99	49	Innovación Tecnológica	0.85	0.94	0.90	50
Cloud	1.00	0.97	0.98	65	Seguridad	0.97	0.95	0.96	92
Computador	0.96	0.99	0.97	74	Soporte Técnico	0.95	0.96	0.95	164
Conexión	0.94	0.94	0.94	50	Telecomunicaciones	0.98	0.94	0.96	309
Correo Electrónico	0.98	1.00	0.99	43	accuracy			0.95	777
Filtro Web	1.00	0.94	0.97	47	macro avg	0.93	0.95	0.94	777
Gestión de Usuarios	0.97	0.97	0.97	63	weighted avg	0.95	0.95	0.95	777
Implementación de Tecnologías Emergentes	0.89	1.00	0.94	50					
Impresora	0.98	0.96	0.97	51					
Portal Web	0.98	0.95	0.96	43					
Redes Sociales	0.98	0.93	0.95	55					
Talleres de Capacitación	0.94	0.96	0.95	48					
accuracy			0.97	777					
macro avg	0.97	0.97	0.97	777					
weighted avg	0.97	0.97	0.97	777					

Ilustración 12 Resultados del modelo Support Vector Machine (SVM).

Métrica	Servicio	Categoría
Precisión	0.97	0.95
Recall	0.97	0.95
F1-Score	0.97	0.95
Accuracy	0.97	0.95
Support	777	777

Las matrices de confusión, ilustración 13, para ambos modelos muestran un rendimiento positivo y favorable. La mayoría de las categorías y servicios tienen una alta precisión, con un número predominante de aciertos (verdaderos positivos), lo que indica que el clasificador realiza correctamente la asignación de clases. Aunque existen algunos errores de clasificación, especialmente en las clases con menos ejemplos, los resultados globales demuestran que el modelo es eficaz y capaz de identificar correctamente la mayoría de las solicitudes.

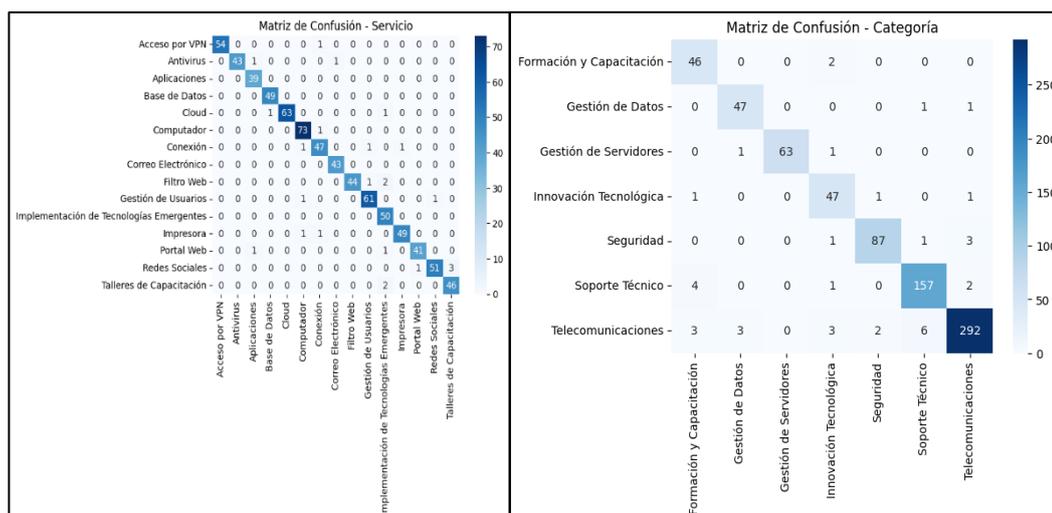


Ilustración 13 Resultados matriz confusión del modelo Support Vector Machine (SVM).

Modelo Naive Bayes.

Para este modelo llevamos la misma estructura en el entrenamiento, primero cargamos el modelo en lenguaje español, aplicamos de nuevo una limpieza de datos, eliminación de caracteres no alfabéticos, palabras vacías, cargamos el archivo, preparar el texto y definimos las variables para los modelos, así como muestra en la gráfica. Luego convertimos las etiquetas de texto de servicio y categoría en valores numéricos, dividimos los datos en entrenamiento 80% y prueba 20% como se visualiza en la ilustración 14.

En el entrenamiento del modelo Naive Bayes, tanto para el servicio y categoría como vemos en la gráfica, se define un pipeline de 'TfidfVectorizer' y clasificador 'nb' con 'MultinomialNB' y se entrena el modelo usando 'X_train y y_train_servicio' y 'X_train y y_train_categoria' para ambos.

```
nb_classifier_servicio = Pipeline([('tfidf', TfidfVectorizer()), ('nb', MultinomialNB())])
nb_classifier_servicio.fit(X_train, y_train_servicio)

nb_classifier_categoria = Pipeline([('tfidf', TfidfVectorizer()), ('nb', MultinomialNB())])
nb_classifier_categoria.fit(X_train, y_train_categoria)
```

Ilustración 14 Proceso 1 del entrenamiento del modelo Naive Bayes.

En la ilustración 15, se evalúa el modelo para el servicio y la categoría, 'y_pred_servicio', 'y_pred_categoria' se usa para la predicción en el conjunto de prueba, se calcula la precisión 'accuracy_score' comparando 'y_test_servicio y

'y_pred_servicio', 'y_test_categoria y y_pred_categoria' y se genera un informe con métricas de evaluación 'precisión, recall, f1-score y support' para cada clase. También se genera la matriz de confusión para la evaluar el rendimiento del modelo para cada clase, cabe mencionar que diversos pasos son iguales para cada modelo, a diferencia que en el entrenamiento si varía de acuerdo al modelo o técnica.

```

y_pred_servicio = nb_classifier_servicio.predict(X_test)
accuracy_servicio = accuracy_score(y_test_servicio, y_pred_servicio)
print(f'Precisión del modelo de servicio: {accuracy_servicio:.2f}')
print('Informe de clasificación para servicio:')
print(classification_report(y_test_servicio, y_pred_servicio))

conf_matrix_servicio = confusion_matrix(y_test_servicio, y_pred_servicio)
print('Matriz de confusión para servicio:')
print(conf_matrix_servicio)

y_pred_categoria = nb_classifier_categoria.predict(X_test)
accuracy_categoria = accuracy_score(y_test_categoria, y_pred_categoria)
print(f'Precisión del modelo de categoría: {accuracy_categoria:.2f}')
print('Informe de clasificación para categoría:')
print(classification_report(y_test_categoria, y_pred_categoria))

conf_matrix_categoria = confusion_matrix(y_test_categoria, y_pred_categoria)
print('Matriz de confusión para categoría:')
print(conf_matrix_categoria)

```

Ilustración 15 Proceso 2 del entrenamiento del modelo Naive Bayes.

Resultados del modelo Naive Bayes, se observa en la ilustración 16:

El modelo Naive Bayes muestra un rendimiento sólido, con buena precisión y recall en la clasificación tanto de servicio como de categoría. El F1-score equilibrado indica una combinación efectiva de ambos. Aunque no es perfecto, clasifica correctamente la mayoría de las instancias.

Evaluación del clasificador de Servicio:					Evaluación del clasificador de Categoría:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Precisión: 0.8867438867438867					Precisión: 0.888030888030888				
Acceso por VPN	0.88	0.89	0.88	55	Formación y Capacitación	0.91	0.83	0.87	48
Antivirus	1.00	0.78	0.88	45	Gestión de Datos	1.00	0.63	0.78	49
Aplicaciones	0.94	0.74	0.83	39	Gestión de Servidores	1.00	0.88	0.93	65
Base de Datos	0.94	0.94	0.94	49	Innovación Tecnológica	0.94	0.68	0.79	50
Cloud	0.87	0.95	0.91	65	Seguridad	0.97	0.79	0.87	92
Computador	0.73	1.00	0.84	74	Soporte Técnico	0.91	0.93	0.92	164
Conexión	0.97	0.76	0.85	50	Telecomunicaciones	0.83	0.98	0.89	309
Correo Electrónico	0.89	0.95	0.92	43					
Filtro Web	0.83	0.96	0.89	47	accuracy			0.89	777
Gestión de Usuarios	1.00	0.83	0.90	63	macro avg	0.94	0.82	0.87	777
Implementación de Tecnologías Emergentes	0.87	0.82	0.85	50	weighted avg	0.90	0.89	0.89	777
Impresora	0.96	0.88	0.92	51					
Portal Web	0.97	0.86	0.91	43					
Redes Sociales	0.91	0.89	0.90	55					
Talleres de Capacitación	0.82	0.96	0.88	48					

Ilustración 16 Resultados del modelo Naive Bayes.

Métrica	Servicio	Categoría
Precisión	0.89	0.89
Recall	0.89	0.89
F1-Score	0.89	0.89
Accuracy	0.89	0.89
Support	777	777

Las matrices de confusión ilustración 17, muestran que el modelo Naive Bayes clasifica correctamente la mayoría de las clases, especialmente las más frecuentes. Sin embargo, algunas clases menos representadas presentan algo de confusión, lo que sugiere oportunidades de mejora.

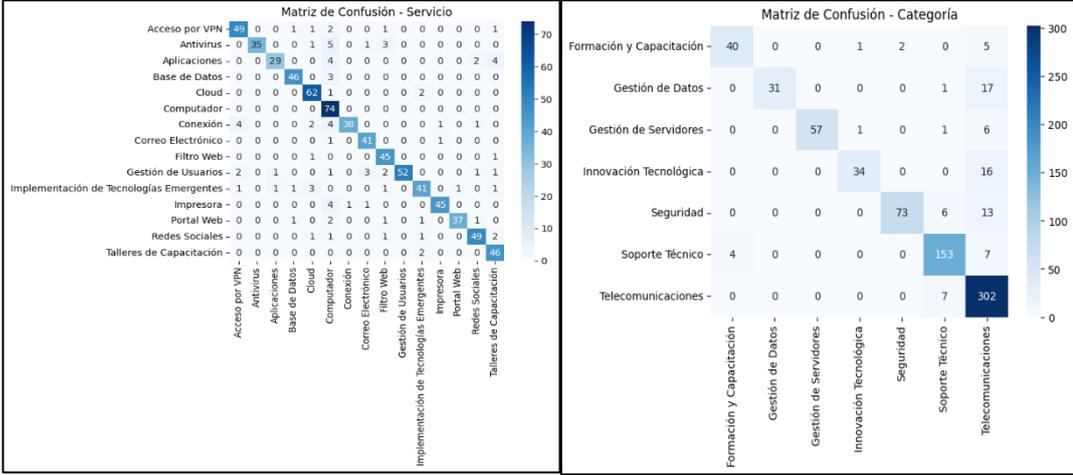


Ilustración 17 Resultados matriz confusión del modelo Naive Bayes.

Modelo K- Nearest Neighbor (KNN).

En el modelo K- Nearest Neighbor (KNN), para el entrenamiento en el servicio y la categoría como vemos en la ilustración 18, se define un pipeline de ‘TfidfVectorizer’ y clasificador ‘knn’ con ‘KNeighborsClassifier’ y se entrena el modelo usando ‘X_train y y_train_servicio’ y ‘X_train y y_train_categoria’ para ambos.

```
knn_classifier_servicio = Pipeline(['tfidf', TfidfVectorizer()), ('knn', KNeighborsClassifier(n_neighbors=5))]
knn_classifier_servicio.fit(X_train, y_train_servicio)
knn_classifier_categoria = Pipeline(['tfidf', TfidfVectorizer()), ('knn', KNeighborsClassifier(n_neighbors=5))]
knn_classifier_categoria.fit(X_train, y_train_categoria)
```

Ilustración 18 Proceso 1 del entrenamiento del modelo K- Nearest Neighbor (KNN).

Evaluamos el modelo para el servicio y la categoría, este código se usa para la predicción en el conjunto de prueba que es con el 20%, se calcula la precisión, comparando 'y_test_servicio y y_pred_servicio', 'y_test_categoria y y_pred_categoria' y se genera un informe con métricas de evaluación 'precisión, recall, f1-score y support' para cada clase, ilustración 19.

```
# Evaluación y matriz de confusión para servicio
y_pred_servicio = knn_classifier_servicio.predict(X_test)
accuracy_servicio = accuracy_score(y_test_servicio, y_pred_servicio)
print(f'Precisión del modelo de servicio: {accuracy_servicio:.2f}')
print('Informe de clasificación para servicio:')
print(classification_report(y_test_servicio, y_pred_servicio))

conf_matrix_servicio = confusion_matrix(y_test_servicio, y_pred_servicio)
print('Matriz de confusión para servicio:')
print(conf_matrix_servicio)

# Evaluación y matriz de confusión para categoría
y_pred_categoria = knn_classifier_categoria.predict(X_test)
accuracy_categoria = accuracy_score(y_test_categoria, y_pred_categoria)
print(f'Precisión del modelo de categoría: {accuracy_categoria:.2f}')
print('Informe de clasificación para categoría:')
print(classification_report(y_test_categoria, y_pred_categoria))

conf_matrix_categoria = confusion_matrix(y_test_categoria, y_pred_categoria)
print('Matriz de confusión para categoría:')
print(conf_matrix_categoria)
```

Ilustración 19 Proceso 2 del entrenamiento del modelo K- Nearest Neighbor (KNN).

Resultados del modelo K- Nearest Neighbor (KNN), ilustración 20:

Como resultado tenemos la precisión, recall, f1-score, estos datos son para la clasificación del servicio y categoría.

Evaluación del clasificador de Servicio (KNN):					Evaluación del clasificador de Categoría (KNN):				
Precisión: 0.8416988416988417					Precisión: 0.8725868725868726				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Acceso por VPN	0.76	0.80	0.78	55	Formación y Capacitación	0.78	0.90	0.83	48
Antivirus	0.75	0.84	0.79	45	Gestión de Datos	0.80	0.82	0.81	49
Aplicaciones	0.79	0.77	0.78	39	Gestión de Servidores	0.91	0.89	0.90	65
Base de Datos	0.82	0.84	0.83	49	Innovación Tecnológica	0.90	0.72	0.80	50
Cloud	0.87	0.89	0.88	65	Seguridad	0.84	0.90	0.87	92
Computador	0.81	0.97	0.88	74	Soporte Técnico	0.86	0.90	0.88	164
Conexión	0.82	0.74	0.78	50	Telecomunicaciones	0.91	0.88	0.89	309
Correo Electrónico	0.80	0.93	0.86	43	accuracy			0.87	777
Filtro Web	0.87	0.96	0.91	47	macro avg	0.86	0.86	0.85	777
Gestión de Usuarios	0.93	0.86	0.89	63	weighted avg	0.87	0.87	0.87	777
Implementación de Tecnologías Emergentes	0.89	0.78	0.83	50					
Impresora	0.94	0.86	0.90	51					
Portal Web	0.91	0.78	0.79	43					
Redes Sociales	0.87	0.84	0.85	55					
Talleres de Capacitación	0.86	0.75	0.80	48					
accuracy			0.84	777					
macro avg	0.84	0.84	0.84	777					
weighted avg	0.85	0.84	0.84	777					

Ilustración 20 Resultados del modelo K- Nearest Neighbor (KNN).

Métrica	Servicio	Categoría
Precisión	0.84	0.87
Recall	0.84	0.86
F1-Score	0.84	0.85
Accuracy	0.84	0.87
Support	777	777

La matriz de confusión para el clasificador KNN muestra una distribución bastante equilibrada entre las clases predichas y las verdaderas, como se observa en la ilustración 21. Aunque algunas categorías tienen un desempeño más bajo en términos de precisión.

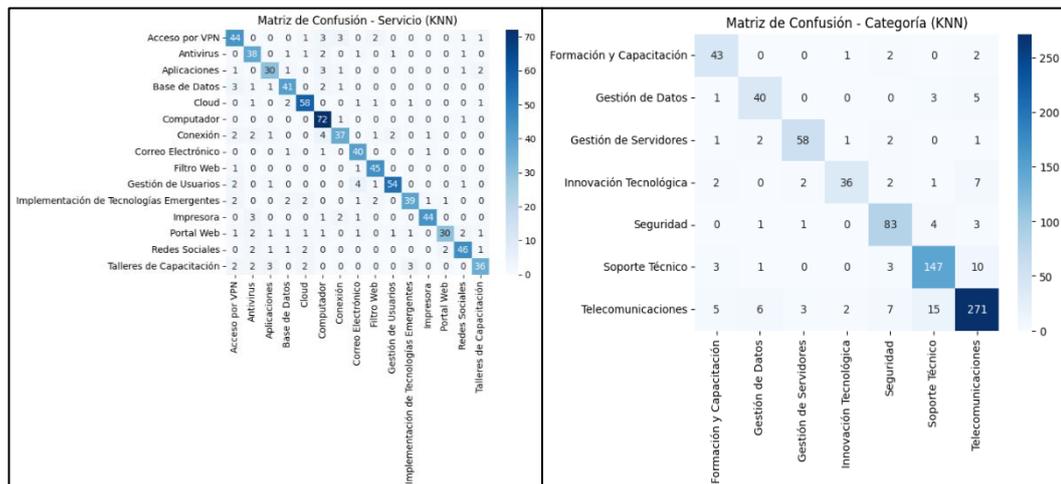


Ilustración 21 Resultados matriz confusión del modelo K- Nearest Neighbor (KNN).

Modelo Logistic Regression.

El entrenamiento para el servicio y la categoría, se define un pipeline de 'TfidfVectorizer' y clasificador 'logistic' con 'LogisticRegression' y se entrena el modelo usando 'X_train y y_train_servicio' y 'X_train y y_train_categoria' para ambos, ilustración 22.

```
logistic_classifier_servicio = Pipeline([('tfidf', TfidfVectorizer()), ('logistic', LogisticRegression(max_iter=1000))])
logistic_classifier_servicio.fit(X_train, y_train_servicio)

logistic_classifier_categoria = Pipeline([('tfidf', TfidfVectorizer()), ('logistic', LogisticRegression(max_iter=1000))])
logistic_classifier_categoria.fit(X_train, y_train_categoria)
```

Ilustración 22 Proceso 1 del entrenamiento del modelo Logistic Regression.

La evaluación del modelo para el servicio y la categoría, se realiza la predicción en el conjunto de prueba que es con el 20% de los datos, se calcula la precisión, comparando 'y_test_servicio y y_pred_servicio', 'y_test_categoria y y_pred_categoria' y se genera un informe con métricas de evaluación 'precisión, recall, f1-score y support' para cada clase, ilustración 23.

```
y_pred_servicio = logistic_classifier_servicio.predict(X_test)
accuracy_servicio = accuracy_score(y_test_servicio, y_pred_servicio)
print(f'Precisión del modelo de servicio: {accuracy_servicio:.2f}')
print('Informe de clasificación para servicio:')
print(classification_report(y_test_servicio, y_pred_servicio))
conf_matrix_servicio = confusion_matrix(y_test_servicio, y_pred_servicio)
print('Matriz de confusión para servicio:')
print(conf_matrix_servicio)

y_pred_categoria = logistic_classifier_categoria.predict(X_test)
accuracy_categoria = accuracy_score(y_test_categoria, y_pred_categoria)
print(f'Precisión del modelo de categoría: {accuracy_categoria:.2f}')
print('Informe de clasificación para categoría:')
print(classification_report(y_test_categoria, y_pred_categoria))
conf_matrix_categoria = confusion_matrix(y_test_categoria, y_pred_categoria)
print('Matriz de confusión para categoría:')
print(conf_matrix_categoria)
```

Ilustración 23 Proceso 2 del entrenamiento del modelo Logistic Regression.

Resultados del modelo Logistic Regression, ilustración 24:

El resultado en términos de Precisión, Recall, F1-score y Accuracy, tanto para la clasificación del servicio y categoría, reflejan la capacidad para identificar las clases para cada servicio y categoría.

Evaluación del clasificador de Servicio (Regresión Logística):					Evaluación del clasificador de Categoría (Regresión Logística):				
Precisión: 0.9575289575289575					Precisión: 0.9562419562419563				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Acceso por VPN	0.98	0.96	0.97	55					
Antivirus	0.98	0.96	0.97	45					
Aplicaciones	0.86	0.95	0.90	39	Formación y Capacitación	0.87	0.96	0.91	48
Base de Datos	0.96	1.00	0.98	49	Gestión de Datos	0.94	0.96	0.95	49
Cloud	1.00	0.97	0.98	65	Gestión de Servidores	1.00	0.95	0.98	65
Computador	0.95	0.99	0.97	74	Innovación Tecnológica	0.92	0.98	0.95	50
Conexión	0.94	0.92	0.93	50	Seguridad	0.98	0.93	0.96	92
Correo Electrónico	0.96	1.00	0.98	43	Soporte Técnico	0.94	0.96	0.95	164
Filtro Web	1.00	0.94	0.97	47	Telecomunicaciones	0.97	0.95	0.96	309
Gestión de Usuarios	0.98	0.90	0.94	63					
Implementación de Tecnologías Emergentes	0.91	0.98	0.94	50	accuracy			0.96	777
Impresora	0.98	0.96	0.97	51	macro avg	0.95	0.96	0.95	777
Portal Web	0.95	0.95	0.95	43	weighted avg	0.96	0.96	0.96	777
Redes Sociales	0.96	0.95	0.95	55					
Talleres de Capacitación	0.94	0.94	0.94	48					
accuracy			0.96	777					
macro avg	0.96	0.96	0.96	777					
weighted avg	0.96	0.96	0.96	777					

Ilustración 24 Resultados del modelo Logistic Regression.

Métrica	Servicio	Categoría
Precisión	0.96	0.96
Recall	0.96	0.96
F1-Score	0.96	0.96
Accuracy	0.96	0.96
Support	777	777

Las matrices de confusión, como se ve en ilustración 25, para los dos modelos exhiben un desempeño positivo. La mayor parte de las categorías y servicios poseen una precisión elevada, con una preponderancia de aciertos, lo que señala que el clasificador lleva a cabo de manera adecuada la distribución de clases. Pese a algunos fallos en la clasificación, particularmente en las clases con menos ejemplos, los resultados a nivel mundial evidencian que el modelo es eficiente y puede reconocer adecuadamente la mayoría de las peticiones.


```

y_pred_servicio = rf_classifier_servicio.predict(X_test)
accuracy_servicio = accuracy_score(y_test_servicio, y_pred_servicio)
print(f'Precisión del modelo de servicio: {accuracy_servicio:.2f}')
print('Informe de clasificación para servicio:')
print(classification_report(y_test_servicio, y_pred_servicio))
conf_matrix_servicio = confusion_matrix(y_test_servicio, y_pred_servicio)
print('Matriz de confusión para servicio:')
print(conf_matrix_servicio)

y_pred_categoria = rf_classifier_categoria.predict(X_test)
accuracy_categoria = accuracy_score(y_test_categoria, y_pred_categoria)
print(f'Precisión del modelo de categoría: {accuracy_categoria:.2f}')
print('Informe de clasificación para categoría:')
print(classification_report(y_test_categoria, y_pred_categoria))
conf_matrix_categoria = confusion_matrix(y_test_categoria, y_pred_categoria)
print('Matriz de confusión para categoría:')
print(conf_matrix_categoria)

```

Ilustración 27 Proceso 2 del entrenamiento del modelo Random Forest.

Resultados del modelo Random Forest, ilustración 28:

Se evaluó en términos de Precisión, Recall, F1-score y Accuracy, tanto para la categorización del servicio como para la categoría. Estos índices de desempeño evidencian la habilidad del modelo para reconocer las clases correspondientes a cada una de ellas.

Evaluación del clasificador de Servicio (Random Forest):					Evaluación del clasificador de Categoría (Random Forest):				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Acceso por VPN	1.00	0.95	0.97	55	Formación y Capacitación	0.94	0.92	0.93	48
Antivirus	0.95	0.89	0.92	45	Gestión de Datos	0.94	0.90	0.92	49
Aplicaciones	1.00	0.90	0.95	39	Gestión de Servidores	1.00	0.97	0.98	65
Base de Datos	0.89	0.96	0.92	49	Innovación Tecnológica	0.89	0.68	0.77	50
Cloud	1.00	0.97	0.98	65	Seguridad	1.00	0.86	0.92	92
Computador	0.84	0.99	0.91	74	Soporte Técnico	0.97	0.97	0.97	164
Conexión	0.85	0.90	0.87	50	Telecomunicaciones	0.89	0.97	0.93	309
Correo Electrónico	0.96	1.00	0.98	43					
Filtro Web	0.98	0.91	0.95	47	accuracy			0.93	777
Gestión de Usuarios	0.88	0.83	0.85	63	macro avg	0.95	0.90	0.92	777
Implementación de Tecnologías Emergentes	0.81	0.88	0.85	50	weighted avg	0.93	0.93	0.93	777
Impresora	0.98	0.96	0.97	51					
Portal Web	0.97	0.88	0.93	43					
Redes Sociales	0.89	0.89	0.89	55					
Talleres de Capacitación	0.96	0.92	0.94	48					
accuracy			0.92	777					
macro avg	0.93	0.92	0.92	777					
weighted avg	0.93	0.92	0.92	777					

Ilustración 28 Resultados del modelo Random Forest.

Métrica	Servicio	Categoría
Precisión	0.92	0.93
Recall	0.92	0.93

F1-Score	0.92	0.93
Accuracy	0.92	0.93
Support	777	777

Las matrices de confusión, ilustración 29, se evidencian que el modelo Random Forest clasifica de manera adecuada la mayoría de las clases, en particular las más comunes. No obstante, ciertas clases menos representadas muestran un cierto desorden, lo que indica posibilidades de optimizar los datos e incorporar nuevos.

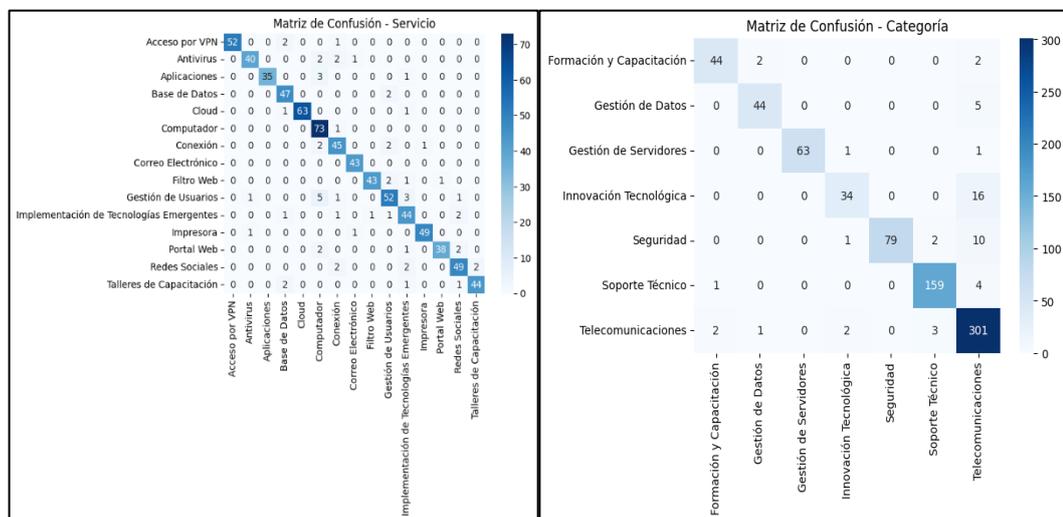


Ilustración 29 Resultados matriz confusión del modelo Random Forest.

Resultados de modelos de aprendizaje supervisados.

Técnica	Métrica	Servicio	Categoría	Support
SVM	Precisión	0.97	0.95	777
	Recall	0.97	0.95	
	F1-Score	0.97	0.95	
	Accuracy	0.97	0.95	
	Precisión	0.96	0.96	777
	Recall	0.96	0.96	

Regression Logistic	F1-Score	0.96	0.96	
	Accuracy	0.96	0.96	
Naive Bayes	Precisión	0.89	0.89	777
	Recall	0.89	0.89	
	F1-Score	0.89	0.89	
	Accuracy	0.89	0.89	
Random Forest	Precisión	0.92	0.93	777
	Recall	0.92	0.93	
	F1-Score	0.92	0.93	
	Accuracy	0.92	0.93	
KNN	Precisión	0.84	0.87	777
	Recall	0.84	0.86	
	F1-Score	0.84	0.85	
	Accuracy	0.84	0.87	

Ilustración 30 Métricas de Evaluación de Modelos de Clasificación.

Observamos que el modelo Support Vector Machine (SVM) destaca en las métricas de rendimiento, por lo que se seleccionará para realizar pruebas de clasificación de solicitudes con datos nuevos en el chatbot.

2.5.5 Diagramas de caso de uso

El esquema de caso de uso es crucial, es una ilustración visual que permite reconocer a los participantes en la interacción con el sistema (chatbot). Este modelo permite reconocer la conexión entre cada actor (ya sea un usuario u otro sistema) con las distintas características del sistema. Mediante figuras simples interconectadas mediante líneas, se describen las posibles interacciones que satisfacen las necesidades del sistema, lo que facilita la comprensión de las relaciones y la organización de los casos de uso de manera nítida y ordenada.

2.5.5.1 Diagrama general.

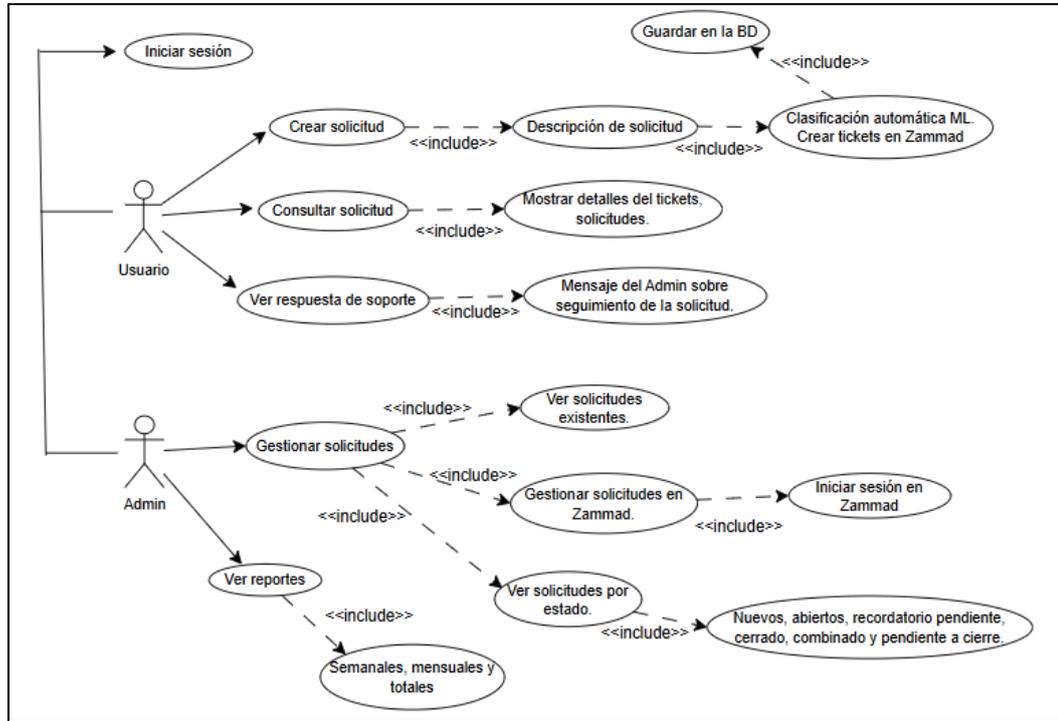


Ilustración 31 Diagrama de caso de uso – rol general.

Caso de uso: Rol general	
Actores	Usuario, Administrador
Tipo	Flujo básico.
Objetivo	Permite el acceso al usuario y al administrador.
Resumen	El chatbot después del inicio de sesión le mostrará un menú de opciones con las funcionalidades tanto para el usuario y para el administrador.
Flujo principal	✓ El usuario o el administrador llamará al chatbot.

	<ul style="list-style-type: none"> ✓ El chatbot da un previo saludo y pedirá sus credenciales. ✓ El chatbot mostrará el menú de opciones para el usuario, con el proceso respectivo en: enviar solicitudes, ver detalles y respuestas. ✓ El chatbot mostrará al administrador opciones de gestionar y ver reportes de las solicitudes emitidas por los usuarios.
Subflujos	El sistema o chatbot tiene acceso a la base de datos pgAdmin y Zammad para realizar registros y consultas de las solicitudes y tickets.
Excepciones	El chatbot muestra en la pantalla de ejecución cuando se conecta a la base de datos y también notifica si hay error al conectarse.

Tabla 9 Caso de uso – Rol del general

2.5.5.2 Diagrama de acceso al sistema.

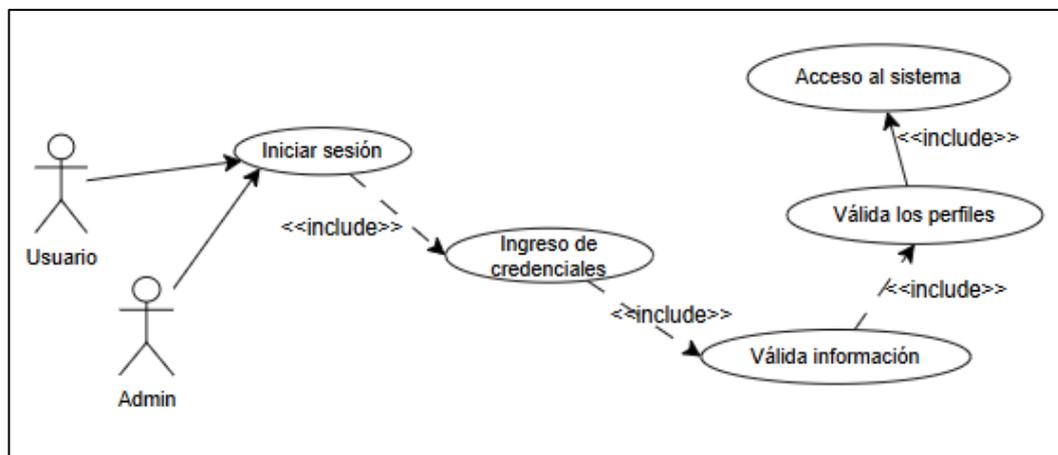


Ilustración 32 Diagrama de caso de uso - acceso al sistema

Acceso al sistema	
Actores	Usuario, Administrador.
Tipo	Flujo básico.
Objetivo	Permite el acceso al usuario
Resumen	Permite el acceso tanto para el usuario o administrador, ingresando sus credenciales, realiza la validación de sus datos al sistema según su perfil y permite el acceso.
Flujo principal	<ul style="list-style-type: none"> ✓ El usuario llamará al chatbot. ✓ El chatbot da un previo saludo y pedirá sus credenciales. ✓ El chatbot permitirá el acceso al sistema. ✓ El chatbot le presentará el menú de acuerdo con el rol que tenga el usuario.
Subflujos	El sistema o chatbot realizará la respectiva validación del usuario, verifica en la base de datos si el usuario se encuentra registrado y el rol que tiene y retorna o muestra el menú al usuario.
Excepciones	El chatbot no permitirá el acceso a usuarios que no se encuentra registrado en la base de datos de Zammad.

Tabla 10 Caso de uso – Acceso al sistema.

2.5.5.3 Diagrama de rol de usuario

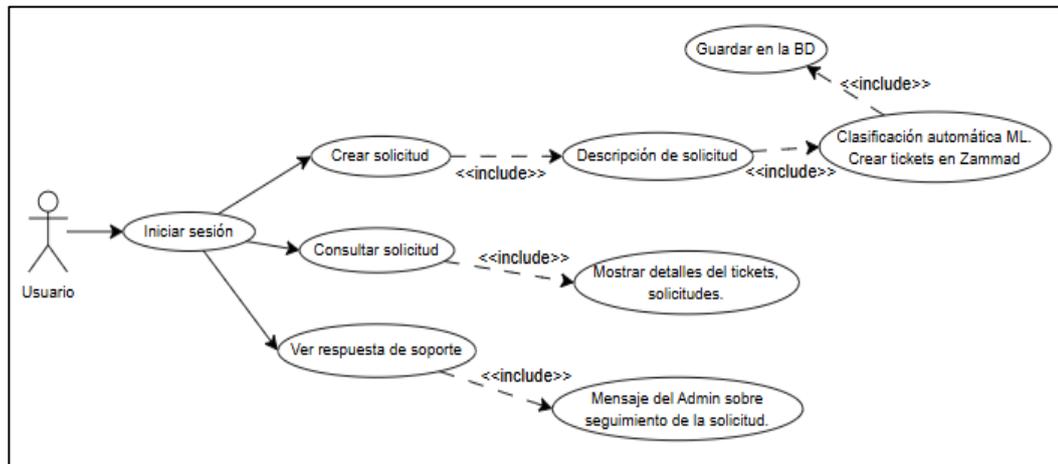


Ilustración 33 Diagrama de caso de uso – rol del usuario

Rol del usuario	
Actores	Usuario
Tipo	Flujo básico.
Objetivo	Permite el acceso al usuario.
Resumen	<p>El chatbot después del inicio de sesión le mostrará un menú con dos botones de opción, crear solicitud y ver detalles de la solicitud (tickets), el primer botón permite al usuario describir la solicitud, en la que el chatbot mediante el modelo entrenado, esta se clasificará automáticamente en el servicio y categoría de acuerdo con la solicitud, posteriormente se creará un ticket en Zammad y estos datos se guardaran en la base de datos (pgAdmin).</p> <p>El segundo botón permite a usuario ver los detalles de la solicitud, tickets,</p>

	<p>fecha, entre otros detalles de cada solicitud realizadas por el mismo.</p> <p>El tercer botón permite al usuario ver mensajes que los administradores envían sobre el seguimiento de sus solicitudes.</p>
Flujo principal	<ul style="list-style-type: none"> ✓ El usuario llamará al chatbot. ✓ El chatbot da un previo saludo y pedirá sus credenciales. ✓ El chatbot mostrará el menú. ✓ El chatbot pedirá que describa la solicitud. ✓ El chatbot clasificará automáticamente dichas solicitudes en Zammad y la base de datos pgAdmin. ✓ El chatbot creará ticket en Zammad y guardará en base de datos pgAdmin. ✓ El chatbot mostrará detalles de las solicitudes enviadas y se verifica el seguimiento de este.
Subflujos	<p>El sistema o chatbot deberá tener conexión precisa con la base de datos pgAdmin para el registro y consulta de solicitudes y tickets.</p>
Excepciones	<p>El chatbot no permitirá el registro de solicitudes que no tengan coherencia o que sean demasiado cortos.</p>

Tabla 11 Caso de uso – Rol del usuario.

2.5.5.4 Diagrama del rol Administrador

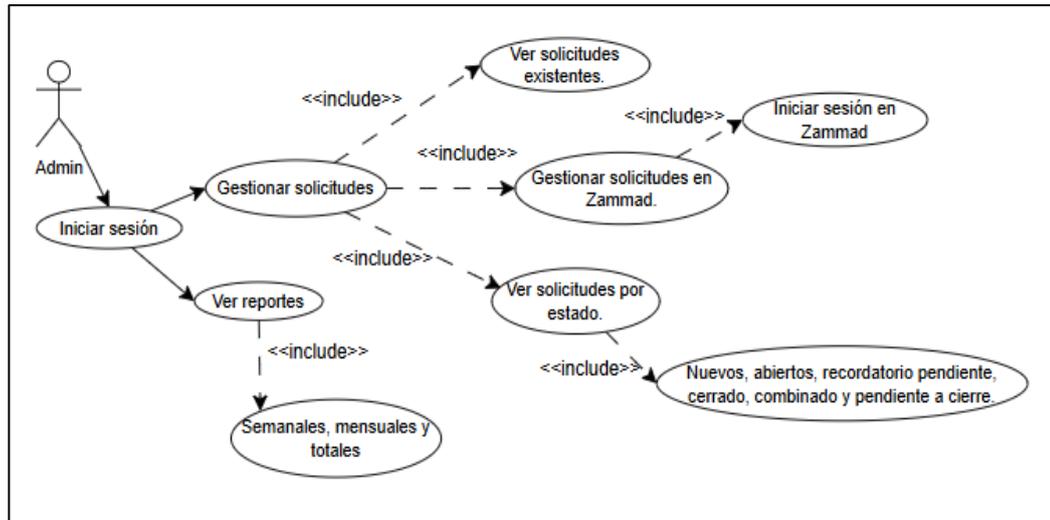


Ilustración 34 Diagrama de caso de uso – rol del administrador

Rol del Administrador	
Actores	Administrador
Tipo	Flujo básico.
Objetivo	Permite el acceso al Administrador
Resumen	El chatbot después del inicio de sesión le mostrará un menú con dos botones de opción, acceder a Gestionar solicitudes, este muestra otra pantalla que permite al administrador ver todas las solicitudes existentes, redirigirse a Zammad para su posterior ingreso y dar seguimiento a las solicitudes emitidas por los usuarios, tickets e incluso a los usuarios registrados, también le permite ver los registros por secciones, es decir, que

	verá las solicitudes por el estado en las que estén.
Flujo principal	<ul style="list-style-type: none"> ✓ El Administrador llamará al chatbot. ✓ El chatbot da un previo saludo y pedirá sus credenciales. ✓ El chatbot mostrará el menú. ✓ El chatbot pedirá cuál de las dos opciones desea el administrador. ✓ El administrador se redirigirá a Zammad, acceder a ella y dar seguimiento a todas las solicitudes de cada uno de los trabajadores, puede editar, enviar, eliminar y dar por cierre cada solicitud. ✓ El administrador solicitará las solicitudes existentes y por estado en archivos .txt
Subflujos	El sistema o chatbot deberá redireccionar al administrador tanto a Zammad como mostrar los datos de las solicitudes registradas y tickets creados.
Excepciones	El chatbot no permitirá el acceso y mostrar el menú secundario si no es administrador.

Tabla 12 Caso de uso – Rol del administrador.

2.5.6 Modelado de Datos

El siguiente diseño de la base de datos es la de Zammad. Zammad como un sistema de gestión de tickets y de código abierto cuenta con su base de datos con 131 tablas

aproximadamente, pero para acceder a ella debemos integrarla con PostgreSQL que conecta mediante pgAdmin. En la siguiente ilustración 31, se muestran los detalles de las tablas principales en la que se le integrará al chatbot tales como: users, roles, roles_users, groups, roles_groups, requests, tickets, ticket_states y tickets_articles.

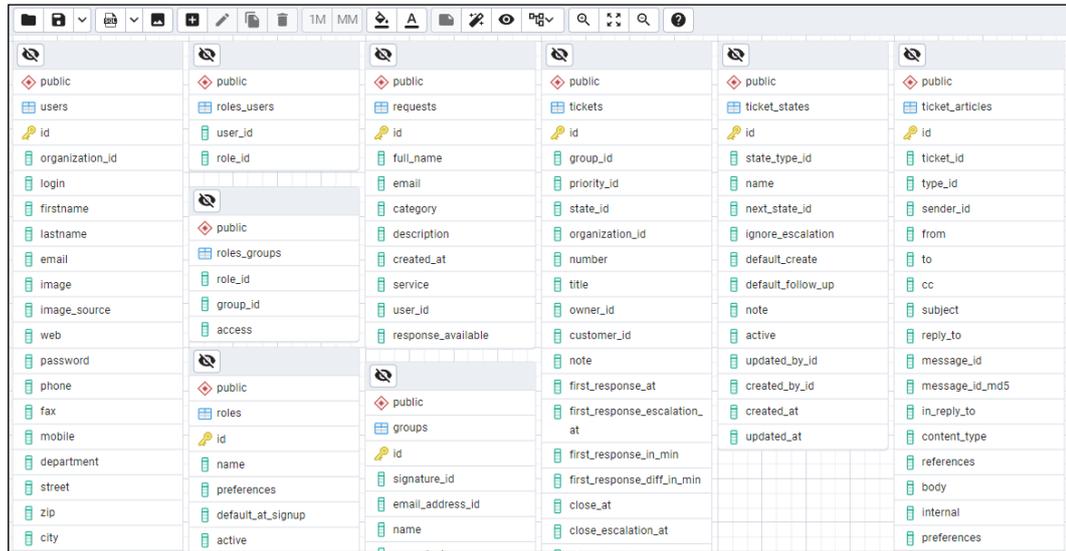


Ilustración 35 Diseño de base de datos PostgreSQL.

2.6 Diseño de Interfaces

La interfaz en el chatbot está diseñada para atender a dos tipos de usuarios: rol trabajador y administrador. Esta interfaz de mensajería nativa en Telegram facilita la interacción entre el usuario y el bot mediante una experiencia de mensajería intuitiva, donde permite enviar mensajes y seleccionar botones con facilidad dependiendo de las necesidades de cada rol.

A continuación, se mencionan los siguientes componentes de la interfaz:

- Nombre del perfil del chatbot.
- Campo de texto para descripción de mensajes y botón para enviar.
- Contenedor de mensajes e interacción entre el usuario y el chatbot.
- Botones que Telegram proporciona para opciones extras.

2.6.1 Interfaz inicio de sesión de rol usuario.

En la ilustración 36, se puede apreciar que el rol trabajador cuenta con tres opciones de selección:

- a) El usuario llama al bot y esta le pide su número de cédula, posteriormente le envía un código al correo para verificar su identidad con respecto al registro que tiene en el sistema.
- b) El chatbot le da acceso y le muestra el menú. Opciones:
- c) Crear solicitud, esta opción permite al trabajador describir la solicitud que desea enviar.
- d) Ver detalles de tickets, brinda al trabajador detalles de los tickets creados y de las solicitudes.
- e) Ver respuesta de soporte, da a conocer al trabajador si el encargado de TICS deja un mensaje sobre el seguimiento de la solicitud.
- f) Salir, esta opción permite que el usuario cierre sesión.

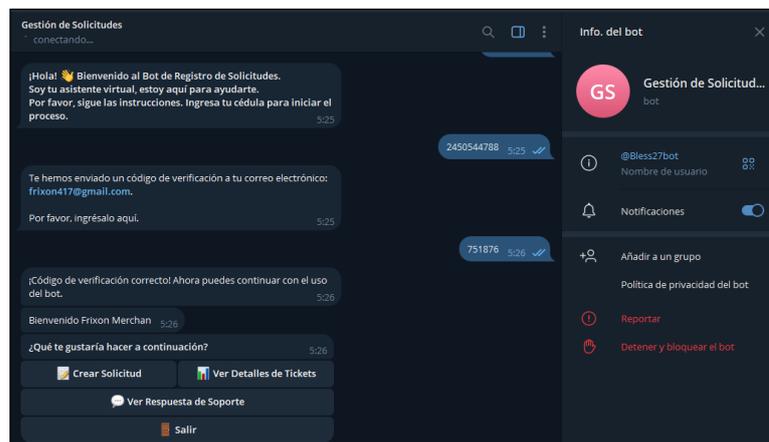


Ilustración 36 Interfaz inicio de sesión de rol usuario.

2.6.2 Interfaz rol usuario: Crear solicitud.

Se puede apreciar que, en la ilustración 37, el chatbot permite al usuario describir la solicitud, registrarla de forma automática y categorizarla dependiendo de la descripción, muestra detalles del registro como: La categoría en que se le asigne, el servicio y el número de ticket.



Ilustración 37 Interfaz rol usuario: Crear solicitud.

2.6.3 Interfaz de creación de Tickets en Zammad.

En la ilustración 38, después de que el usuario haya enviado su solicitud, se creará un ticket en Zammad, esta detalla al usuario que la emitió, el número de ticket, el título, el asunto, la descripción de la solicitud, estado, la etiqueta. Junto a ella se encuentra el botón de actualización.

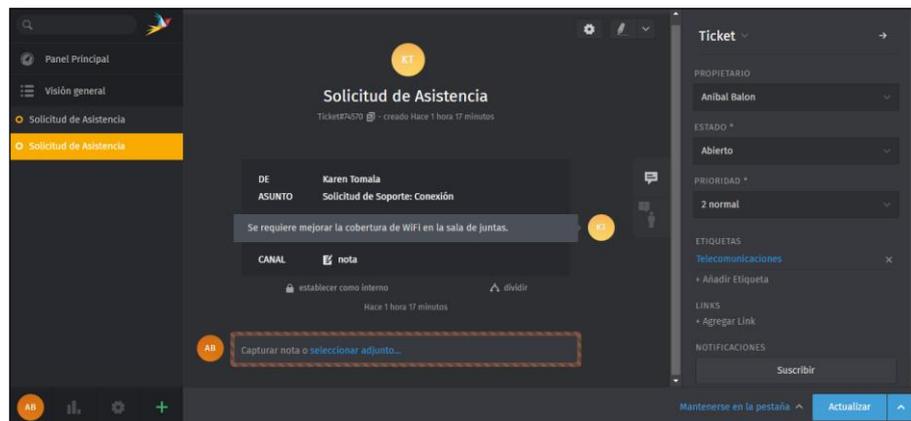


Ilustración 38 Interfaz de creación de Tickets en Zammad.

2.6.4 Interfaz rol usuario: Ver detalles de tickets.

Vemos que, en la ilustración, el chatbot permite al usuario ver los detalles del ticket y de la solicitud registrada, refleja la categoría, el servicio, la descripción, el tickets y fecha en la que fue creada, esto permite al usuario observar el seguimiento que se le dará a la solicitud.

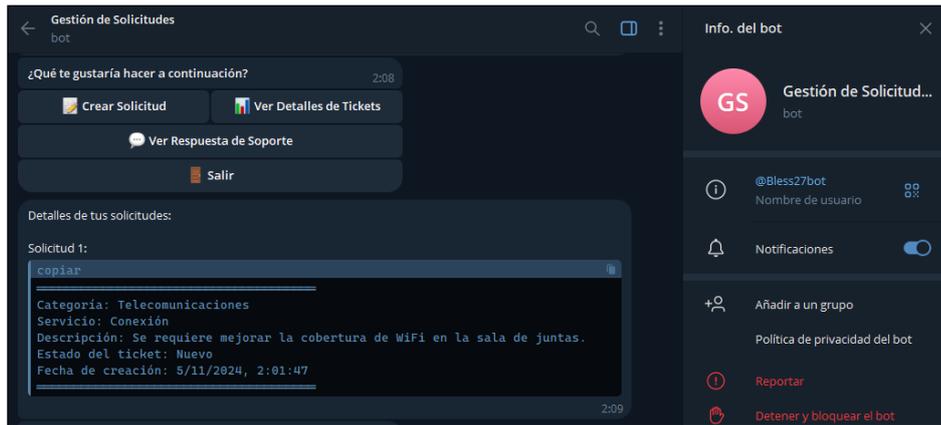


Ilustración 39 Interfaz rol usuario: Ver detalles de tickets.

2.6.5 Interfaz rol usuario: Ver respuesta de soporte.

La ilustración 40, el administrador o encargado de TICS, es aquel que dará seguimiento a las solicitudes en Zammad, el mismo que notificará al usuario o le envía un mensaje sobre el seguimiento. Y en la ilustración 41 el usuario podrá observar el seguimiento de la solicitud mediante el mensaje que le deajo el de soporte.

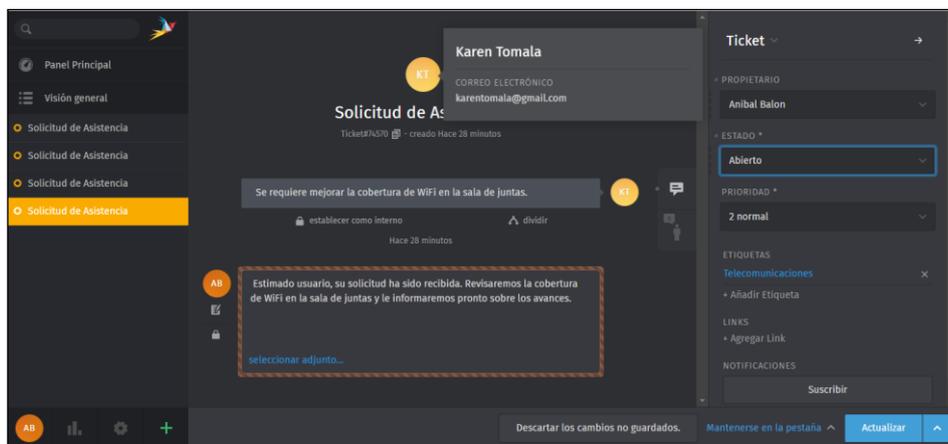


Ilustración 40 Interfaz rol usuario: Ver respuesta de soporte 1.

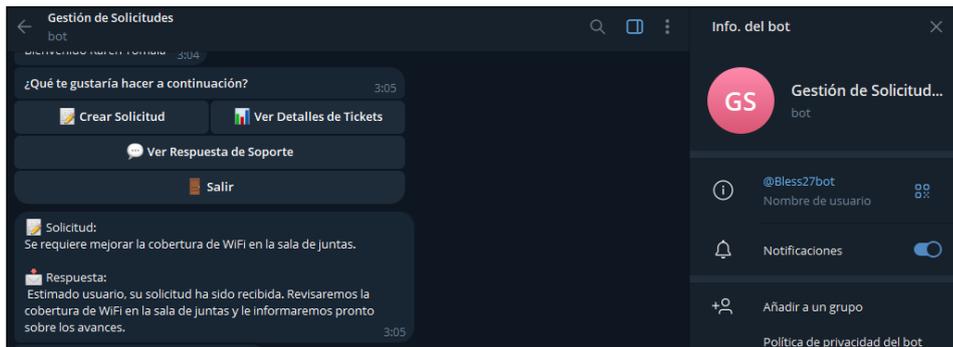


Ilustración 41 Interfaz rol usuario: Ver respuesta de soporte 2.

2.6.6 Interfaz rol Administrador.

En la ilustración 42. El rol administrador tiene dos opciones, una es ‘Gestionar solicitudes’ y la otra es ‘Ver reportes’, La opción 1: Permite que el administrador pueda tener acceso a ‘Ver solicitudes existentes’, ‘Gestionar solicitudes en Zammad’, ‘Ver solicitudes por estado’.

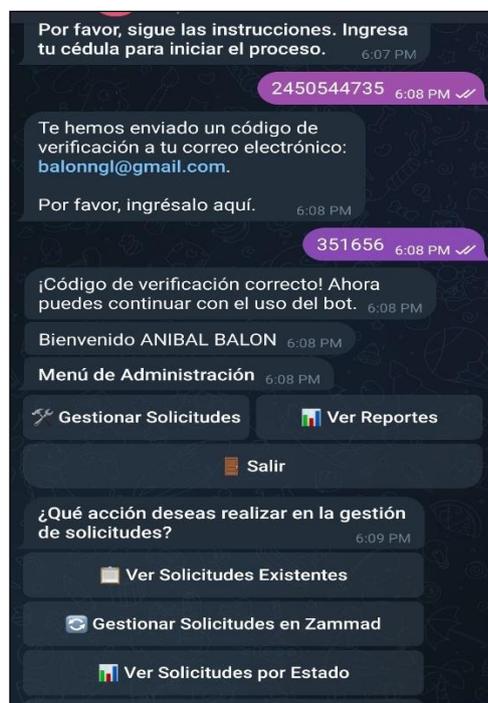


Ilustración 42 Interfaz rol Administrador - Gestionar solicitudes en Zammad.

En ‘Ver solicitudes existentes’, se le proporciona todos los datos registrados de las solicitudes en un archivo.txt como muestra en la ilustración 43.

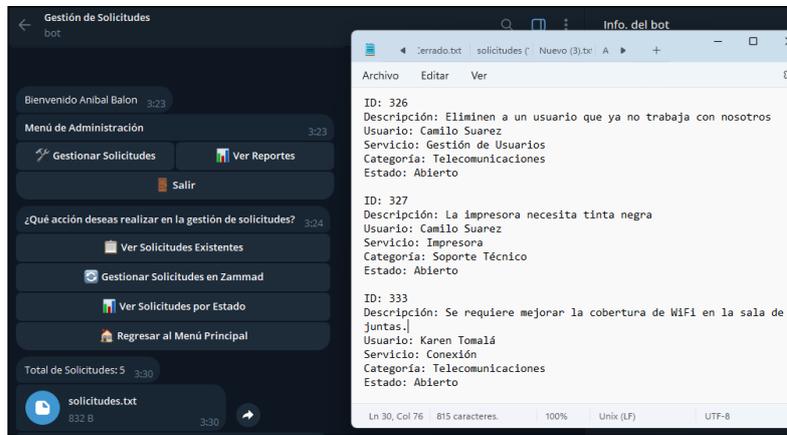


Ilustración 43 Interfaz rol Administrador - Ver solicitudes existentes.

En ‘Gestionar solicitudes en Zammad’, como se muestra en la ilustración 44, el bot le da direccionamiento hacia la herramienta Zammad para dar seguimiento a las solicitudes de cada usuario.

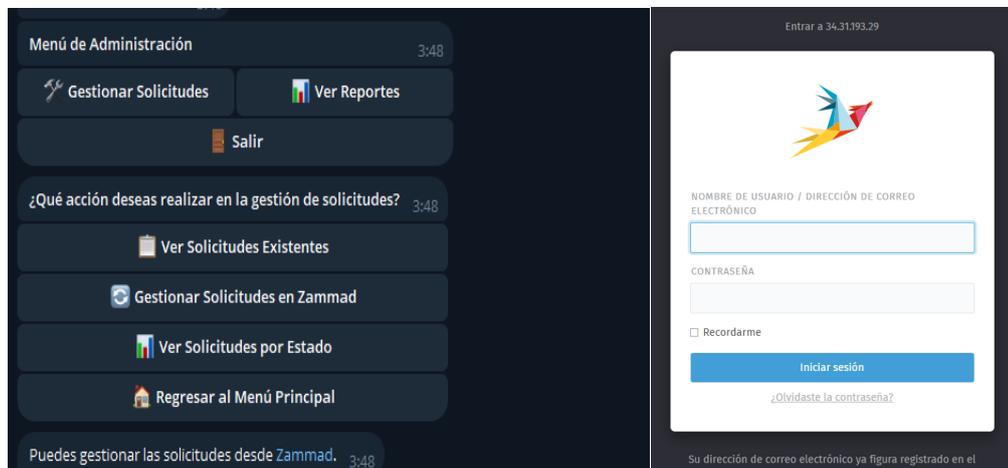


Ilustración 44 Ver solicitudes existentes - Gestionar solicitudes en Zammad.

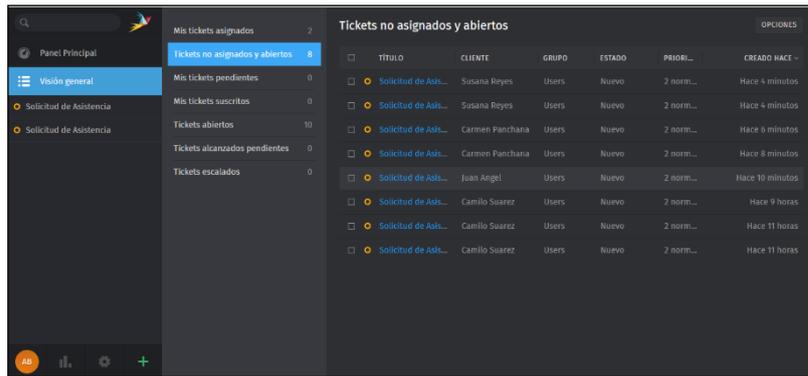


Ilustración 44: Ver solicitudes existentes - Gestionar solicitudes en Zammad.

La ilustración 45, en ‘Ver solicitudes por estado’, se observa seis secciones u opciones en la que el administrador podrá ver los estados en las que se encuentran los tickets de cada solicitud, se le proporcionará en archivos .txt, desde su registro ‘Nuevo’, ‘Abierto’, ‘Recordatorio pendiente’, ‘Cerrado’, ‘Combinado’ y ‘Pendiente a cierre’.

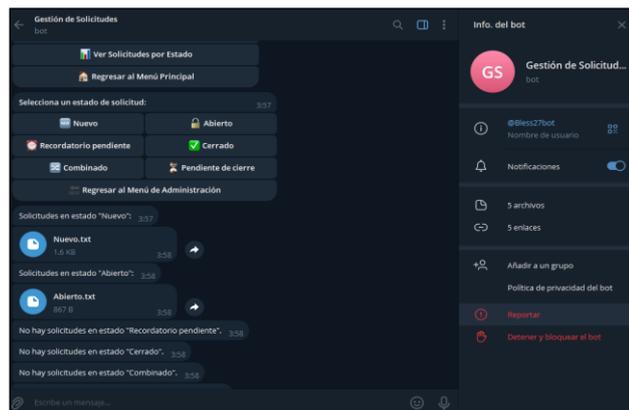
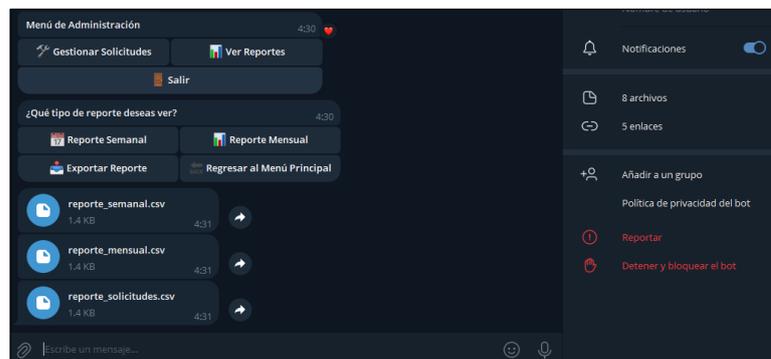


Ilustración 45 Ver solicitudes por estado

En la opción 2 ‘Ver reporte’, como se ve en la ilustración 46, el bot podrá proporcionar al administrador descargar los reportes ya sea por semana, mensual o todos los reportes en un solo archivo .csv



ID	Descripción	Servicio	Categoría	Fecha
339	Solicitud la creación de una cuenta de usuario para el sistema de gestión documental par	Gestión de Usuarios	Telecomunicaciones	5 de noviembre de 2024 a las 03:47:19
338	Instalación de software antivirus en los equipos del área de Recursos Humanos	Antivirus	Seguridad	5 de noviembre de 2024 a las 03:46:58
337	Solicitud inscripción para el próximo taller de capacitación en atención al ciudadano.	Talleres de Capacitación	Formación y Capacitación	5 de noviembre de 2024 a las 03:44:46
336	Instalan una nueva impresora en el área de Atención al Público.	Impresora	Soporte Técnico	5 de noviembre de 2024 a las 03:43:19
335	Por favor, optimicen la base de datos de consultas ciudadanas para mejorar su rendim	Base de Datos	Gestión de Datos	5 de noviembre de 2024 a las 03:40:26
333	Se requiere mejorar la cobertura de WIFI en la sala de juntas.	Conexión	Telecomunicaciones	5 de noviembre de 2024 a las 02:21:47
327	La impresora necesita tinta negra	Impresora	Soporte Técnico	4 de noviembre de 2024 a las 17:57:37
326	Eliminen a un usuario que ya no trabaja con nosotros	Gestión de Usuarios	Telecomunicaciones	4 de noviembre de 2024 a las 16:39:50
323	La pc no inicia sesión, ayudennos en eso	Computador	Soporte Técnico	4 de noviembre de 2024 a las 16:32:03
324	Solicitud cambio de contraseña de mi correo electronico	Correo Electrónico	Telecomunicaciones	4 de noviembre de 2024 a las 16:29:23

Ilustración 46 Ver reportes – Semanal, mensual y anual.

2.6.7 Registro de nuevo usuario.

En la ilustración 47. El administrador o encargado de TICs en aquel que puede registrar un nuevo usuario al sistema mediante Zammad. El administrador solicita al usuario sus datos personales como: Nombres apellidos, número de cédula, correo electrónico, lo que se procede a su registro exitoso y se le admite al usuario el uso del chatbot en Telegram ante alguna petición.

Ilustración 47 Registro de usuario – Mediante el Zammad

2.6.8 Interfaz al monitoreo en Grafana

Grafana con la integración con Elasticsearch de Zammad permite la visualización y el análisis de datos, Grafana permite crear paneles interactivos que son los dashboards, para esto se creó un dashboards para observar la distribución de los tickets por sus estados. El administrador es aquel que puede acceder a Grafana y visualizar los datos en tiempo real de los tickets que se han creado mediante el registro de solicitudes por medio del chatbot, esto mide el rendimiento en la que el sistema está operando y la cantidad de tickets que se están creando como se aprecia en la ilustración 48.



Ilustración 48 Monitoreo de tickests en grafana.

También se integró a Grafana el PostgreSQL de Zammad para evaluar el rendimiento del modelo que ya fue entrenado e implementado en el chatbot, los registros son favorables ante la clasificación de solicitudes de manera automática y que fueron clasificadas en las categorías y servicios correspondiente de acuerdo a las descripciones.

En la ilustración 49, se puede observar el tiempo en que las solicitudes fueron clasificadas, se observa el tiempo transcurrido entre el inicio de la clasificación hasta el momento actual y el tiempo en que la solicitud fue creada y el inicio de su clasificación.

Identificación	Categoría	Servicio	Descripción	creado_en	clasificación_com	tiempo_de_clasifi	clasificación_tiem
100	Soporte Técnico	Impresora	La impresora multifunc	03-12-2024 00:23:23	03-12-2024 00:23:28	00:08:00	0 minutos 5 segundos
101	Formación y Capacitar	Talleres de Capacitaci	Solicito la organizació	03-12-2024 00:26:25	03-12-2024 00:26:30	00:04:58	0 minutos 5 segundos
102	Telecomunicaciones	Acceso por VPN	Requero que se me ot	03-12-2024 00:26:54	03-12-2024 00:26:59	00:04:29	0 minutos 5 segundos
103	Telecomunicaciones	Acceso por VPN	El acceso VPN de algu	03-12-2024 00:27:20	03-12-2024 00:27:25	00:04:03	0 minutos 5 segundos
104	Telecomunicaciones	Correo Electrónico	Solicite la creación de	03-12-2024 00:28:31	03-12-2024 00:28:36	00:02:52	0 minutos 5 segundos
105	Telecomunicaciones	Portal Web	El portal web no está c	03-12-2024 00:28:46	03-12-2024 00:28:51	00:02:37	0 minutos 5 segundos
106	Gestión de servidores	Nube	Aumentar el espacio d	03-12-2024 00:29:00	03-12-2024 00:29:05	00:02:23	0 minutos 5 segundos
108	Formación y Capacitar	Talleres de Capacitaci	Requiere capacitación	03-12-2024 00:30:52	03-12-2024 00:30:57	00:00:31	0 minutos 5 segundos

Ilustración 49 Tiempo en clasificación de solicitudes

En la ilustración 50, verificamos el registro de las solicitudes que fueron clasificadas en las categorías y servicios y la verificación con los valores predichos, es decir, se realiza una comparativa sí coinciden entre ambos.

id	category	service	predicted_category	predicted_service	description	classification_started_s
100	Soporte Técnico	Impresora	Soporte Técnico	Impresora	La impresora multifuncional	03-12-2024 00:23:28
101	Formación y Capacitación	Talleres de Capacitación	Formación y Capacitación	Talleres de Capacitación	Solicito la organización de u	03-12-2024 00:26:30
102	Telecomunicaciones	Acceso por VPN	Telecomunicaciones	Acceso por VPN	Requiero que se me otorgue	03-12-2024 00:26:59
103	Telecomunicaciones	Acceso por VPN	Telecomunicaciones	Acceso por VPN	El acceso VPN de algunos u	03-12-2024 00:27:25
104	Telecomunicaciones	Correo Electrónico	Telecomunicaciones	Correo Electrónico	Solicite la creación de una c	03-12-2024 00:28:36
105	Telecomunicaciones	Portal Web	Telecomunicaciones	Portal Web	El portal web no está carga	03-12-2024 00:28:51
106	Gestión de servidores	Nube	Gestión de servidores	Nube	Aumentar el espacio de alm	03-12-2024 00:29:05
108	Formación y Capacitación	Talleres de Capacitación	Formación y Capacitación	Talleres de Capacitación	Requiere capacitación sobr	03-12-2024 00:30:57

Ilustración 50 Predicción de las solicitudes registradas

En la ilustración 51, presenta la cantidad de solicitudes que fueron clasificadas correctamente en las categorías y servicios. Los resultados están ordenados de mayor a menor según la cantidad de predicciones correctas.

category prevista	servicio previsto	recuento_de_predicciones_correctas
Telecomunicaciones	Correo Electrónico	3
Formación y Capacitación	Talleres de Capacitación	3
Telecomunicaciones	Acceso por VPN	2
Gestión de datos	Base de datos	2
Telecomunicaciones	Portal Web	2
Innovación Tecnológica	Implementación de Tecnologías Emergentes	2
Soporte Técnico	Impresora	2
Seguridad	Filtro Web	1
Soporte Técnico	Impresora	1

Ilustración 51 Total de solicitudes correctas

2.7 Pruebas

Es importante asegurar el correcto desempeño del proyecto, para lo cual es esencial llevar a cabo pruebas de su funcionamiento. El propósito de estas pruebas es asegurar que cada requerimiento propuesto se cumpla adecuadamente en el chatbot en Telegram, asegurando de esta manera que todas las funcionalidades satisfagan las expectativas de los usuarios. Las evaluaciones se realizarán en diversas etapas,

cada una concebida para valorar distintos elementos en la que opera el chatbot. Primero, se llevarán a cabo ensayos de funcionamiento para asegurar que el asistente funcione conforme a lo estipulado en los requerimientos del proyecto. Esto abarca la verificación de órdenes y respuestas del chatbot, además de la habilidad para gestionar interacciones incorrectas o imprevistas de los usuarios. También, se realizarán pruebas de conexión para garantizar la correcta interacción del chatbot con otros sistemas, tales como Zammad y bases de datos PostgreSQL.

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	001
CASO PRÁCTICO	Identificar usuario con su respectivo rol
OBJETIVO	Comprobar que el ingreso al sistema se esté llevando de forma correcta por parte del sistema y el rol de cada usuario
PERFILES	Usuario

PROCESO DE EJECUCIÓN DE PRUEBA

- Llamar a bot ‘Gestión de solicitudes’.
- Solicita el ingreso de su número de cédula.
- Usuario ingresa su credencial (número de cédula).
- El chatbot verifica si el usuario se encuentra registrado en la base de datos.
- Si se encuentra registrado, le envía un código de verificación al correo.
- Permite el acceso con un saludo y su nombre.
- El chatbot le muestra el menú de opciones de acuerdo el rol que tiene el usuario.

EVALUACIÓN DE RESULTADOS

RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido

Efectivamente el bot permite al usuario el acceso, un saludo inicial y mostrándole el menú de opciones acorde al rol que tenga.	✓	
---	---	--

Tabla 13 Prueba de funcionalidad - Identificar usuario con su respectivo rol

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	002
CASO PRÁCTICO	Registro de usuario
OBJETIVO	Comprobar el registro correcto al sistema de un nuevo usuario.
PERFILES	Administrador

PROCESO DE EJECUCIÓN DE PRUEBA

- El administrador es quién puede registrar un nuevo usuario mediante Zammad.
- Solicita los datos: Nombres, apellidos, cédula, correo electrónico.
- Acceso al uso del bot en Telegram.

EVALUACIÓN DE RESULTADOS

RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido
El administrador o encargado de TICs registra exitosamente a un nuevo usuario al sistema, autorización del uso del bot.	✓	

Tabla 14 Prueba de funcionalidad – Registro de usuario en Zammad

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	003
CASO PRÁCTICO	Crear solicitud

OBJETIVO	Confirmar que el usuario pueda registrar solicitudes y ver la categorización en la que se registró.
PERFILES	Usuario

PROCESO DE EJECUCIÓN DE PRUEBA

- El usuario selecciona la opción crear solicitud.
- El bot solicitará al usuario que ingrese detalles de la solicitud.
- El usuario detalla o describe la solicitud.
- El bot registra la solicitud y le envía un mensaje en que categoría, servicio y número de ticket. Todo esto se registra en Zammad y posterior a la base de datos.
- El bot seguido le muestra el menú de opciones, si el usuario desea realizar otra solicitud o el cierre de este.

EVALUACIÓN DE RESULTADOS

RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido
El bot direcciona al usuario a realizar este proceso en que debe describir la solicitud y mostrando su registro exitoso en Zammad y dar seguimiento por parte de soporte.	✓	

Tabla 15 Prueba de funcionalidad – Crear solicitud

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	004
CASO PRÁCTICO	Ver detalles de tickets
OBJETIVO	Mostrar al usuario detalles del ticket, solicitud, sobre el seguimiento que le dará soporte.

PERFILES	Usuario
-----------------	---------

PROCESO DE EJECUCIÓN DE PRUEBA

- El usuario selecciona la opción ver detalles de tickets.
- El bot da detalles del registro de su solicitud, el estado del ticket, categoría en la que fue registrada, así mismo el servicio, la descripción y fecha en la que fue registrada.
- El bot seguido le muestra el menú de opciones, si el usuario desea realizar otra solicitud o el cierre del mismo.

EVALUACIÓN DE RESULTADOS

RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido
El bot proporciona detalles sobre su registro y ver el estado en la que se encuentra su ticket, dado que el seguimiento lo llevan los administradores en Zammad.	✓	

Tabla 16 Prueba de funcionalidad – Ver detalles de ticket.

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	005
CASO PRÁCTICO	Ver respuesta de soporte.
OBJETIVO	Confirmar que el bot muestre los mensajes del administrador o de soporte sobre el seguimiento de la solicitud enviada.
PERFILES	Usuario

PROCESO DE EJECUCIÓN DE PRUEBA

- El trabajador selecciona la opción Ver respuesta de soporte.

- El bot muestra los mensajes que deja el administrador o soporte al usuario sobre el seguimiento de su solicitud, es decir: descripción de la solicitud y seguida respuesta de soporte.
- Finalmente, el bot procede a enviar el menú para nuevos registros o cerrar sesión.

EVALUACIÓN DE RESULTADOS		
RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido
El bot muestra mensajes del administrador que deja a cada usuario durante el seguimiento de cada solicitud.	✓	

Tabla 17 Prueba de funcionalidad – Ver respuesta de soporte.

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	006
CASO PRÁCTICO	Gestionar solicitudes – Ver solicitudes existentes.
OBJETIVO	Comprobar que el administrador pueda acceder a los datos de todas las solicitudes.
PERFILES	Administrador.

PROCESO DE EJECUCIÓN DE PRUEBA

- El administrador selecciona la opción ver solicitudes existentes.
- El bot le da el acceso a observar el total de todas las solicitudes existentes y le envía un archivo .txt., donde le da detalles como: id del usuario, la descripción de la solicitud, el nombre del usuario, el servicio, categoría en la que fue clasificada y el estado en la que se encuentra el ticket.

RESULTADOS DE LA PRUEBA

RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido
El bot envía el total de las solicitudes y un archivo .txt con detalles de todos los registros.	✓	

Tabla 18 Prueba de funcionalidad – Gestionar solicitudes – Ver solicitudes existentes.

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	007
CASO PRÁCTICO	Gestionar solicitudes en Zammad.
OBJETIVO	Verificar el direccionamiento al administrador y seguimiento de las solicitudes mediante Zammad.
PERFILES	Administrador.

PROCESO DE EJECUCIÓN DE PRUEBA

- El administrador selecciona la opción gestionar solicitudes en Zammad
- El bot le muestra un mensaje con acceso a Zammad.
- El administrador da seguimiento a cada solicitud registrada por los usuarios.
- El administrador puede editar, eliminar, crear solicitudes, actualizar los estados y crear nuevos usuarios mediante Zammad.

EVALUACIÓN DE RESULTADOS

RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido
El administrador accede a Zammad mediante un mensaje enviado por el bot en la selección de la opción ‘Gestionar solicitudes en Zammad’, el	✓	

administrador da seguimiento a los registros de cada usuario.		
---	--	--

Tabla 19 Prueba de funcionalidad – Gestionar solicitudes – Gestionar solicitudes en Zammad.

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	008
CASO PRÁCTICO	Ver solicitudes por estado.
OBJETIVO	Comprobar que el bot realice cada acción que haga el administrador, al obtener detalles de cada estado de las solicitudes.
PERFILES	Administrador.

PROCESO DE EJECUCIÓN DE PRUEBA

- El administrador selecciona la opción Ver solicitudes por estado.
- El bot le muestra un submenú con opciones de: ‘Nuevo’, ‘Abierto’, ‘Recordatorio pendiente’, ‘Cerrado’, ‘Combinado’ y ‘Pendiente a cierre’.
- El administrador en cada opción el bot le envía las solicitudes en archivos .txt de acuerdo al estado en que se encuentre.

EVALUACIÓN DE RESULTADOS

RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido
El bot proporciona en cada opción detalles en archivos .txt de cada solicitud acorde a los estados en que se encuentren, los resultados son factibles ya que dan detalles de cada una de las solicitudes que se han dado seguimiento en Zammad.	✓	

Tabla 20 Prueba de funcionalidad – Gestionar solicitudes – Ver solicitudes por estado.

DESCRIPCIÓN DEL CASO DE PRUEBA

PRUEBA N°	009
CASO PRÁCTICO	Ver reportes.
OBJETIVO	Validar que el administrador pueda solicitar los reportes de las solicitudes en un archivo .CSV.
PERFILES	Administrador.

PROCESO DE EJECUCIÓN DE PRUEBA

- El administrador selecciona la opción Ver reportes.
- El bot le muestra un submenú con opciones de: Reporte semanal, reporte mensual y reporte anual’.
- El administrador en cada una de las opciones obtiene un reporte en archivo .csv de las solicitudes registradas por el bot, esto facilita a un seguimiento sobre cómo está funcionando el sistema en la clasificación de las solicitudes.

EVALUACIÓN DE RESULTADOS

RESULTADOS	EVALUACIÓN	
	Exitoso	Fallido
Los resultados muestran que el bot cumple con cada una de las funciones, proporciona de manera correcta los reportes semanales, mensuales y reporte general al administrador en archivos .csv	✓	

Tabla 21 Prueba de funcionalidad – Gestionar solicitudes – Ver reportes.

2.8 Resultados

Como resultado en este presente proyecto de titulación, se implementó un sistema de clasificación automatizada de solicitudes, integrando un modelo de aprendizaje supervisado entrenado con un conjunto de datos depurado. El proceso de

depuración del dataset se llevó a cabo mediante el uso de librerías en Python como pandas, spacy, nltk, y re. Estas herramientas posibilitaron la purificación de los datos al eliminar fallos, inconsistencias y valores nulos en las columnas de descripción, servicio y categoría. La depuración también normalizó el texto, la eliminación de caracteres no alfabéticos y la lematización de palabras con el objetivo de optimizar la calidad de los datos y su idoneidad para el aprendizaje del modelo. Luego de este procedimiento, el dataset pasó a ser un archivo ordenado y limpio para su utilización en el entrenamiento del modelo.

Se llevó a cabo una comparación de varios modelos de aprendizaje automático supervisados para la categorización automatizada de las peticiones. Se evaluaron modelos como SVM, Regresión Logística, Naive Bayes, Random Forest y KNN, como se ve en los resultados del (Gráfico 9). El modelo SVM se evidenció como el más apropiado para el conjunto de datos, logrando una precisión del 97% en la categorización del servicio y del 95% en la categoría. Otras medidas como el recall y la F1-Score también mostraron resultados gratificantes, corroborando el balance entre la exactitud y la habilidad para identificar las instancias pertinentes.

Se integró el modelo de categorización automatizada de peticiones con el sistema de gestión de servicios fundamentado en ITIL v4. La integración se llevó a cabo con Zammad, facilitando la generación automática de tickets cuando se detectó una petición que necesitaba atención extra. Esta integración mejoró el proceso laboral del administrador, asegurando que las peticiones relevantes se registraran de forma eficaz sin necesidad de intervención manual. Además, herramientas como Grafana y Elasticsearch facilitaron la supervisión en tiempo real y la búsqueda veloz de información asociada a las peticiones, esto ayuda considerablemente al administrador, que permite dar un seguimiento de cómo se están clasificando las solicitudes y generando los tickets, esto ayuda a un futuro seguir mejorando el sistema. Los hallazgos logrados corroboran que la solución implementada ha generado un efecto beneficioso en la eficacia operacional del sistema.

CONCLUSIONES

- La depuración del dataset aseguró la calidad de los datos empleados para el entrenamiento del modelo de clasificación. Usando herramientas en Python como pandas, nltk y spaCy, se prescindieron de duplicados, valores nulos y caracteres no pertinentes, consiguiendo así un conjunto de datos organizado y puro.
- Tanto la normalización del texto como la lematización ayudaron de manera significativa a disminuir el ruido en los datos, convirtiéndolos en información apropiada para el entrenamiento. la excelencia del dataset obtenido evidencia la relevancia de un preprocesamiento detallado en proyectos de aprendizaje automático enfocados en la clasificación automatizada.
- La evaluación de varios modelos de aprendizaje automático, tales como SVM, Random Forest, Regresión Logística, Naive Bayes y KNN, facilitó la identificación del que mejor desempeñaba la tarea de categorizar las solicitudes, se tomó en cuenta indicadores como Precisión, Recall, F1-Score y Accuracy, los cuales garantizaron un contraste imparcial entre las técnicas.
- Se consiguió la elección del modelo más apropiado a través de un exhaustivo proceso de pruebas y validación cruzada. El modelo SVM resaltó por su equilibrio entre exactitud y eficacia, exhibiendo resultados uniformes en todas las métricas analizadas.
- La integración del modelo SVM con el sistema de gestión basado en ITIL v4 automatizó la categorización de peticiones, simplificando procesos con herramientas como Zammad y Elasticsearch para un flujo de trabajo eficiente.
- La implementación del chatbot en JavaScript con Node.js, optimizó la interacción con los usuarios al automatizar la categorización de peticiones mediante Telegram. Con Zammad, posibilitó la generación automática de tickets y una clasificación eficaz de solicitudes, mejorando los tiempos de

registro y de respuesta por medio de los administradores o encargados de TI.

- El modelo SVM, con una precisión promedio del 97% en servicio y 95% en categoría, demostró su efectividad en reducir los tiempos de registro de solicitudes en un entorno real.
- La integración de Zammad para la gestión de tickets y Grafana para el monitoreo en tiempo real permitió evaluar la precisión del chatbot y su impacto en la gestión de solicitudes, identificando patrones relevantes.

RECOMENDACIONES

- Se recomienda un estudio más detallado de los datos de entrada para descubrir posibles patrones adicionales o atributos relevantes para mejorar aún más los resultados. Ingresar más información referente a descripciones de solicitudes al conjunto de datos y actualizar continuamente el modelo, esto permitirá que el sistema se adapte a los cambios en la demanda y las nuevas tendencias.
- Se recomienda estudiar y evaluar nuevos modelos avanzados de aprendizaje automático, como BERT o redes neuronales profundas, para superar las limitaciones de los modelos tradicionales y lograr resultados más precisos y eficientes en la clasificación de solicitudes.
- Se recomienda continuar con el uso de la herramienta Zammad para la administración de tickets y registro de solicitudes, esto es favorable porque se puede integrar con otras plataformas. También, es aconsejable tener en cuenta la incorporación de otros canales de mensajería, como lo es WhatsApp, a pesar de tener un costo por suscripción, es considerable por la interacción que tienen los usuarios constantemente con esta plataforma.
- Se recomienda implementar un sistema de retroalimentación para los usuarios y así evaluar la efectividad del chatbot y el sistema de calificación. Esto facilitará los cambios continuos y garantizará que el sistema siga satisfaciendo las necesidades y expectativas de los usuarios.

REFERENCIAS

- [1] J. A. A. CARRILLO, «PROPUESTA DE GESTIÓN DE INCIDENTES Y GESTIÓN DE SOLICITUDES,» 2022.
- [2] J. Gómez, UNIVERSITIC 2020. Análisis de la madurez digital, MADRID - España, 2021.
- [3] F. Llorens, A. Fernández, T. Rodríguez, S. Cadena y C. y Franco, Estudio de la madurez digital en sistemas universitarios iberoamericanos, 2021.
- [4] A. F. Román Nureña, Analizar, diseñar e implementar un sistema de información que soporte el proceso de Gestión de Solicitudes de Servicio del Ministerio de Comunicaciones de una Iglesia Evangélica, Lima, 2019-07-23.
- [5] J. L. Rosas Flores, Implementación de una solución de inteligencia de negocios para optimizar la gestión de solicitudes del área de preventa en una empresa de telecomunicaciones en la ciudad de Lima, Lima-Peru, 2021.
- [6] C. V. M. ROBALLO, «ANÁLISIS PARA LA IMPLEMENTACIÓN DE LA HERRAMIENTA BAYESTREE:,» Octubre, 2020.
- [7] GlobalSuite, ¿Qué es ITIL y para que sirve?
https://www.globalsuitesolutions.com/es/que-es-til-y-para-que-sirve/#ITIL_v4.
- [8] CONTAINERIZE, Zammad "Descripción general"
<https://products.containerize.com/es/helpdesk/zammad/>, 2001-2024.
- [9] B2CHAT, Telegram <https://www.b2chat.io/blog/redes-sociales/telegram-que-es-para-que-sirve-como-funciona/>.

- [10] UNIR, TypeScript <https://www.unir.net/ingenieria/revista/que-es-typescript/>.
- [11] a. i. o. tecnologia, Postman <https://assemblerinstitute.com/blog/que-es-postman/>.
- [12] i. g. then, Express.js <https://ifgeekthen.nttdata.com/s/post/que-es-expressjs-y-primeros-pasos-MCCIDTDOZFQNBXGDJ5WENXXNNY4?language=es>.
- [13] aws.amazon, Python <https://aws.amazon.com/es/what-is/python/>.
- [14] a. cloud, Grafana <https://ausum.cloud/que-es-grafana-y-como-se-usa-en-la-monitorizacion/>.
- [15] M. Velazquez, ¿Qué es Visual Studio Code? <https://recluit.com/que-es-visual-studio-code/>, 2021.
- [16] Elastic, Elasticsearch - <https://www.elastic.co/es/elasticsearch>.
- [17] S. Nacional, PLAN DE CREACIÓN DE OPORTUNIDADES 2021-2025, Ecuador, 2021.
- [18] D. L. Castelló, Metodología experimental - <https://poliformat.upv.es/access/content/user/24389381/Contenido%20abiert%20al%20p%C3%BAblico/Metodolog%C3%ADa%20de%20la%20investigaci%C3%B3n/3.2%20Metodologi%C3%A1%20experimental.pdf>, Universidad Politécnica de Valencia.
- [19] C. R. Istacuy, ITIL Foundation ITIL 4 Edition 'https://www.udocz.com/apuntes/751022/itil-foundation-itil-4-edition-spanish-pdf-carlos', Londres, Inglaterra, 10/7/23.

- [20] P. C. D. L. REPÚBLICA, DECRETO PRESIDENCIAL MEDIANTE EL CUAL SE CAMBIA DE NOMBRE DE MINISTERIO DE BIENESTAR SOCIAL AL DE MINISTERIO DE INCLUSIÓN ECONÓMICA Y SOCIAL.
- [21] A. Cubo, J. L. Hernández Carrión, M. Porrúa y B. Roseth, Transformación digital '<http://dx.doi.org/10.18235/0004402>', Aug 2022.
- [22] M. d. I. E. y. Social, MISIÓN / VISIÓN / VALORES <https://www.inclusion.gob.ec/misionvision/>.
- [23] ITIL, ITIL, ITIL v4, 2024.
- [24] A. Software, ITIL v4, 2023.
- [25] freshworks, ITIL v4 '<https://www.freshworks.com/es/freshservice/itil/itil-v4/>', 2024.
- [26] I. C. Desk, Gestión de solicitudes de servicio '<https://www.ibm.com/docs/es/control-desk/7.6.1.2?topic=overview-service-request-management>', 2021.
- [27] Salesforce, El machine learning '<https://www.salesforce.com/es/resources/definition/machine-learning/>'.
- [28] M. A. M. F. Antonio Aliaga Ibarra, postgresQL '<https://iessanvicente.com/colaboraciones/postgreSQL.pdf>', 2008.
- [29] I. mailchimp, ¿Qué es Telegram? '<https://mailchimp.com/es/resources/what-is-telegram/>'.
- [30] kKinsta, ¿Qué es Express.js? '<https://kinsta.com/es/base-de-conocimiento/que-es-express/#qu-es-expressjs>', 2022.

- [31] OpenWebinars, Visual Studio Code: Editor de código para desarrolladores 'https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/', 2022.
- [32] J. C. B. P. Á. Cuervo López, Formular acciones de mejora utilizando las buenas prácticas de ITIL v4, para mejorar la gestión de solicitudes e incidentes de la universidad Santo Tomás sede principal en Bogotá. 'https://hdl.handle.net/20.500.12494/20192', 2020.
- [33] F. d. M. P. Montes, Comparación de modelos de Machine Learning para determinar qué modelo se aproxima más a una asignación manual en ServiceDesk - https://hdl.handle.net/20.500.12672/21853, Universidad Nacional Mayor de San Marcos, 2023.
- [34] A. J. S. Jiménez, Análisis del Servicio de Help desk Zammad para Mejoras en la Eficiencia y Experiencia del Usuario en el GAD Parroquial de Pimocha. 'http://dspace.utb.edu.ec/handle/49000/17068', 2024.
- [35] Invgate, Proceso de Gestión de Solicitudes de Servicio. https://blog.invgate.com/es/gestion-de-solicitudes-de-servicio, 2023.
- [36] D. R. H. Sampieri, METODOLOGÍA DE LA INVESTIGACIÓN, Quinta edición, Mexico: MCGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V..
- [37] freshworks, ITIL v4, 2024.
- [38] S. Nandy, Machine Learning Classification Algorithms 'https://github.com/sayantann11/all-classification-templates-for-ML'.
- [39] A. Cubo, J. L. Hernández Carrión, M. Porrúa y B. Roseth, Transformación digital - https://publications.iadb.org/es/guia-de-transformacion-digital-del-gobierno#:~:text=La%20transformaci%C3%B3n%20digital%20del%20gob

ierno% 20es% 20el% 20cambio% 20de% 20cultura,forma% 20eficiente% 2C%
20transparente% 20y% 20segura..

ANEXOS

Anexo 1. Catálogo de solicitudes de servicios.

CATÁLOGO DE SOLICITUDES DE SERVICIOS - GOBERNACIÓN			
Código	Categoría	Servicio	Descripción
SG-TEL-001	Telecomunicaciones	Correo Electrónico	Gestión de cuentas de correo electrónico, incluyendo la creación y eliminación de cuentas para el personal.
SG-TEL-002		Portal Web	Mantenimiento del portal web para asegurar la disponibilidad de información pública y la interacción con ciudadanos.
SG-TEL-003		Conexión	Gestión de servicios de conectividad WiFi en todas las instalaciones para facilitar el acceso a Internet.
SG-TEL-004		Gestión de Usuarios	Administración de cuentas de usuario en sistemas de la gobernación, asegurando un acceso adecuado a la información.
SG-TEL-005		Acceso por VPN	Provisión de conexiones VPN para el acceso seguro a recursos internos desde ubicaciones remotas.
SG-TEL-006		Redes Sociales	Gestión de cuentas de redes sociales oficiales para mejorar la comunicación y difusión de información.
SG-SEG-007	Seguridad	Antivirus	Implementación de soluciones para proteger sistemas contra amenazas cibernéticas.
SG-SEG-008		Filtro Web	Establecimiento de filtros de contenido para garantizar un acceso seguro a la web.
SG-GSV-009	Gestión de Servidores	Cloud	Administración de servidores en la nube para asegurar la disponibilidad de servicios.
SG-GDA-010	Gestión de Datos	Base de Datos	Soporte en la administración de bases de datos para garantizar su integridad y disponibilidad.
SG-INT-011	Innovación Tecnológica	Implementación de Tecnologías Emergentes	Investigación e implementación de tecnologías que optimizan servicios públicos.
SG-SOT-012	Soporte Técnico	Aplicaciones	Soporte para la instalación y mantenimiento de aplicaciones utilizadas en la gobernación.
SG-SOT-013		Computador	Asistencia técnica para la reparación y mantenimiento de computadoras.
SG-SOT-014		Impresora	Soporte para la instalación y mantenimiento de impresoras.
SG-FCP-016	Formación y Capacitación	Talleres de Capacitación	Organización de talleres para mejorar habilidades y conocimientos del personal.

Anexo 2. Árbol de problemas.

