CLASIFICACIÓN AUTOMÁTICA DE TICKETS DE SOPORTE TÉCNICO BASADO EN PROCESAMIENTO DE LENGUAJE NATURAL APLICADO A LA GESTIÓN OPERATIVA DE WEBCOOPEC SYSTEM.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR



SANTIAGO GUAYASAMÍN

DOCENTE TUTOR: DR. CARLOS JULIO MAYORGA

MAESTRÍA EN SISTEMAS DE LA INFORMACIÓN CON MENCIÓN EN DATA SCIENCE

16 DE AGOSTO DEL 2026

Contenido

DATOS INICIALES DEL PROYECTO 9
CAPÍTULO I 10
1.1 Introducción 10
1.1.1 Planteamiento del problema *12*
1.2 Objetivos 14
1.2.1 Objetivo general *14*
1.2.2 Objetivos específicos *14*
1.3 Justificación 15
1.3.1 Importancia operativa y empresarial *15*
1.3.2 Relevancia académica y tecnológica *16*
1.3.3 Viabilidad del estudio *16*
1.3.4 Relación con los objetivos de investigación *17*
1.4 Preguntas 18
1.4.1 Pregunta general *18*
1.4.2 Preguntas específicas *18*
CAPÍTULO II 19
MARCO TEÓRICO 19
2.1 Introducción del capítulo 19
2.2 Estrategia de revisión de literatura 20
2.3 Antecedentes de investigación 24
2.4 Bases conceptuales 29
2.4.1 Webcoopec System como proveedor de servicios tecnológicos *30*
2.4.2 Gestión de soporte técnico y sistemas de mesa de ayuda *31*
2.4.3 osTicket como plataforma de gestión de tickets *32*
2.4.4 Procesamiento de lenguaje natural y minería de texto *32*
2.4.5 Machine Learning aplicado a la clasificación automática de tickets *33*
2.4.6 Metodología CRISP-DM y evaluación del modelo *34*
2.4.7 Protección de datos personales *34*
2.5 Fundamentación teórica y modelos de referencia 35
2.5.1 Metodología CRISP-DM *35*

2.5.2 Procesamiento de lenguaje natural *36*
2.5.3 Aprendizaje supervisado y modelos de clasificación *36*
2.5.4 Evaluación del desempeño del modelo *37*
2.5.5 Relación del marco teórico con la investigación *37*
2.6 Variables, dimensiones y categorías de análisis 38
2.6.1 Variables para el análisis descriptivo *38*
2.6.2 Variables de entrada del modelo de Machine Learning *39*
2.6.3 Variable de salida *40*
2.6.4 Variables derivadas del procesamiento *41*
2.6.5 Variables de evaluación *42*
2.7 Brechas de investigación 43
2.8 Síntesis del marco teórico 46
CAPÍTULO III 48
MARCO METODOLÓGICO 48
3.1 Enfoque de la investigación 48
3.2 Tipo de investigación 48
3.3 Diseño de la investigación 49
3.4 Contexto del estudio 50
3.5 Población 51
3.6 Muestra 51
3.7 Unidad de análisis 52
3.8 Fuente de información 53
3.9 Desarrollo de la metodología CRISP-DM 54
3.9.1 Comprensión del Negocio *54*
3.9.2 Comprensión de los datos *60*
3.9.2.1 Conexión con la fuente de datos. 63
3.9.2.2 Identificación de tablas relevantes. 65
3.9.2.3 Descripción inicial de los datos. 68
3.9.2.4 Identificación de la categoría de producto y de la variable objetivo 69
3.9.2.5 Distribución por canal de ingreso. 73
3.9.2.6 Evaluación inicial de la calidad de los datos. 74
3.9.2.7 Separación de variables descriptivas y predictoras. 75
3.9.2.8 Prevención de fuga de información. 77
3.9.2.9 Análisis Exploratorio de Datos (EDA). 79

3.9.3 Preparación de los datos *95*
3.9.3.1 Construcción del dataset. 96
3.9.3.2 Integración de tablas. 97
3.9.3.3 Limpieza de datos. 99
3.9.3.4 Anonimización. 100
3.9.3.5 Tratamiento de nulos. 101
3.9.3.6 Eliminación de duplicados. 102
3.9.3.7 Normalización del texto. 103
3.9.3.8 Tokenización. 103
3.9.3.9 Stopwords. 104
3.9.3.10 Lematización. 104
3.9.3.11 Vectorización TF-IDF. 105
3.9.3.12 Dataset final. 107
3.9.4 Modelado *110*
3.9.4.1 Conjunto utilizado para el entrenamiento. 111
3.9.4.2 Distribución de las categorías de entrenamiento. 112
3.9.4.3 Selección de algoritmos. 113
3.9.4.4 Modelado mediante Naive Bayes. 114
3.9.4.5 Modelado mediante Support Vector Machine. 115
3.9.4.6 Modelado mediante Random Forest. 116
3.9.4.7 Entrenamiento de los modelos. 117
3.9.4.8 Artefactos generados. 118
3.9.4.9 Preparación para la integración con Actybase Web. 119
3.9.4.10 Resultado de la fase de modelado. 120
3.9.5 Evaluación *120*
3.9.5.1 Conjunto de prueba. 121
3.9.5.2 Métricas utilizadas. 126
3.9.5.3 Comparación general de los modelos. 127
3.9.5.4 Resultados de Support Vector Machine. 128
3.9.5.5 Matriz de confusión. 129
3.9.5.6 Análisis de las causas de error. 132
3.9.5.7 Selección del mejor modelo. 133
3.9.5.8 Ejemplos sintéticos de categorización. 133
3.9.5.9 Limitaciones del modelo seleccionado. 134

3.9.5.10 Resultado de la fase de evaluación. 135
3.9.6 Despliegue *136*
3.9.6.1 Arquitectura de la solución. 136
3.9.6.2 Integración con Actybase Web 137
3.9.6.3 Servicio de inferencia. 138
3.9.6.4 Validación funcional. 139
3.9.6.5 Controles y consideraciones operativas. 140
3.9.6.6 Resultado de la fase. 140
CAPÍTULO IV 146
CONCLUSIONES Y RECOMENDACIONES 146
4.1 Conclusiones 146
4.2 Limitaciones 148
4.2.1 Dependencia de la calidad de los datos históricos *148*
4.2.2 Alcance limitado al contexto organizacional de Webcoopec System *148*
4.2.3 Variabilidad del lenguaje utilizado por los usuarios *149*
4.2.4 Evolución permanente de las categorías de soporte *149*
4.2.5 Restricciones temporales del desarrollo *149*
4.2.6 Integración del modelo como prototipo funcional *149*
4.2.7 Protección y anonimización de los datos *150*
4.3 Recomendaciones 150
4.3.1 Mantener un proceso continuo de actualización del modelo *150*
4.3.2 Fortalecer la calidad de la clasificación manual *150*
4.3.3 Integrar el modelo en el entorno de producción *150*
4.3.4 Implementar tableros de inteligencia de negocios *151*
4.3.5 Evaluar modelos de inteligencia artificial más avanzados *151*
4.4 Aportes de la investigación 151
Referencias 153

Índice de Tablas

Tabla 1 Datos iniciales del proyecto 9
Tabla 2 Base de datos y fuentes consultadas 21
Tabla 3 Cadenas de Búsqueda Utilizadas 22
Tabla 4 Criterios de inclusión y exclusión de fuentes 22
Tabla 5 Matriz de Antecedentes Relevantes 28
Tabla 6 Variables de la investigación 42
Tabla 7 Brechas de literatura 45
Tabla 8 Criterios de inclusión y exclusión de tickets 52
Tabla 9 Objetivos del negocio y objetivos técnicos 60
Tabla 10 Tablas identificadas en la base de datos de Webcoopec Ticket 66
Tabla 11 Conteos iniciales de las entidades de Webcoopec Ticket. 68
Tabla 12 Distribución inicial de tickets por categoría 70
Tabla 13 Categorías de incidencia 71
Tabla 14 Distribución de la variable objetivo en los tickets etiquetados 72
Tabla 15 Distribución de tickets por canal de ingreso 73
Tabla 16 Indicadores de calidad de los tickets 74
Tabla 17 Tickets sin producto y con incidencias 74
Tabla 18 Principales valores nulos identificados 82
Tabla 19 Integración de tablas 97
Tabla 20 Tratamiento de valores nulos 101
Tabla 21 Resultado del tratamiento de duplicados 102
Tabla 22 Vectorización 107
Tabla 23 Distribución por categoría después del preprocesamiento 107
Tabla 24 Distribución de las categorías en el conjunto de entrenamiento 112
Tabla 25 Configuración de Naive Bayes 114
Tabla 26 Configuración de Support Vector Machine 115
Tabla 27 Configuración de Random Forest 116
Tabla 28 Artefactos Generados 118
Tabla 29 Comparación de los modelos sobre el conjunto de prueba 127
Tabla 30 Métricas por categoría de Support Vector Machine 128
Tabla 31 Ejemplos sintéticos procesados por el modelo seleccionado 133

Tabla de Figuras

Figura 1 Fuente de datos - Webcoopec Ticket 54
Figura 2 Logotipo de Webcoopec System Cía. Ltda 56
Figura 3 Resolución de calificación de Webcoopec System como empresa de servicios auxiliares 57
Figura 4 Estructura organizacional de Webcoopec System Cía. Ltda. 58
Figura 5 Proceso actual de gestión de tickets 58
Figura 6 Proceso propuesto con incorporación del modelo de Machine Learning 59
Figura 7 Estructura del proyecto de ciencia de datos para la clasificación de tickets 62
Figura 8 Verificación de la conexión entre Python y la copia local de MariaDB 65
Figura 9 Identificación de las tablas disponibles en la base de datos local 67
Figura 10 Relaciones principales entre las tablas utilizadas en la investigación 68
Figura 11 Conteos iniciales de las entidades de Webcoopec Ticket 69
Figura 12 Distribución inicial de tickets por categoría obtenida mediante Python 71
Figura 13 Carga y panorama general del conjunto de datos 81
Figura 14 Principales valores nulos identificados 83
Figura 15 Porcentaje de valores nulos por variable 84
Figura 16 Análisis de registros duplicados 85
Figura 17 Distribución de los tickets según el tipo de incidencia 87
Figura 18 Distribución de los tickets según el producto seleccionado 89
Figura 19 Distribución de los diez clientes con mayor número de tickets 91
Figura 20 Evolución mensual del número de tickets registrados 92
Figura 21 Distribución de la longitud del asunto y de la descripción inicial 94
Figura 22 Construcción inicial del dataset de tickets 97
Figura 23 Variables obtenidas mediante la integración de las tablas de osTicket 99
Figura 24 Ejemplo del proceso de anonimización textual 100
Figura 25 Ejemplo de tokenización, eliminación de stopwords y lematización 105
Figura 26 Vectorización TF-IDF 106
Figura 27 Resumen de la preparación del dataset 108
Figura 28 Distribución de las clases en el dataset final 109
Figura 29 Carga de las matrices utilizadas para entrenar los categorizadores 112
Figura 30 Distribución de los tipos de incidencia en el conjunto de entrenamiento 113
Figura 31 Entrenamiento de los modelos de categorización automática 117
Figura 32 Tiempo requerido para entrenar los modelos candidatos 118

Figura 33 Modelos almacenados después del entrenamiento 119
Figura 34 Carga del conjunto independiente utilizado para evaluar los modelos 125
Figura 35 Comparación del comportamiento de los modelos sobre el conjunto de prueba 127
Figura 36 Medida F1 por categoría obtenida mediante Support Vector Machine 129
Figura 37 Matriz de confusión de Support Vector Machine 130
Figura 38 Matriz de confusión Naive Bayes 130
Figura 39 Matriz de confusión Random Forest 131
Figura 40 Ejemplos de categorización automática mediante el modelo SVM 134
Figura 41 Arquitectura de integración entre Actybase Web y el servicio local de categorización 136
Figura 42 Submenú Ticket del módulo Analítica de Datos 137
Figura 43 Interfaz para la categorización automática de tickets 138
Figura 44 Resultado de la categorización automática de un ticket sintético 139
Figura 45 Análisis descriptivo Webcoopec Ticket 141
Figura 46 Vista complementaria del análisis descriptivo de Webcoopec Ticket 142

DATOS INICIALES DEL PROYECTO

Tabla 1

Datos iniciales del proyecto

Campo	Debe completar el estudiante
Nombre del estudiante	Santiago Xavier Guayasamín Díaz
Tema de tesis	Clasificación automática de tickets de soporte técnico basado en procesamiento de lenguaje natural aplicado a la gestión operativa de webcoopec system.
Pregunta principal de investigación	¿Cómo puede un modelo basado en técnicas de Procesamiento de Lenguaje Natural (PLN) y Machine Learning automatizar la clasificación de los tickets de soporte técnico para optimizar la gestión operativa de la mesa de ayuda de Webcoopec System?
Objetivo general	Desarrollar un modelo basado en técnicas de procesamiento de lenguaje natural y machine learning que permita automatizar la clasificación de tickets de soporte técnico y optimizar la gestión operativa de una mesa de ayuda empresarial mediante análisis descriptivos, predictivos y prescriptivos.
Docente/Tutor	Carlos Julio Mayorga
Fecha de entrega	16 de agosto del 2026

CAPÍTULO I

1.1 Introducción

En la actualidad, la transformación digital ha impulsado a las organizaciones a generar y almacenar grandes volúmenes de información como resultado de sus procesos operativos. Este crecimiento de los datos ha convertido a la Ciencia de Datos, el Big Data y el Machine Learning en herramientas fundamentales para extraer conocimiento, automatizar procesos y respaldar decisiones fundamentadas en evidencia. En este ámbito, el procesamiento inteligente de la información representa una oportunidad para mejorar la eficiencia operativa, acortar tiempos de respuesta y mejorar la calidad de los servicios ofrecidos por las organizaciones.

Uno de los procesos que ha tenido un crecimiento notable en los últimos años se debe a la administración de mesas de ayuda (Help Desk), las cuales constituyen el principal canal de comunicación entre las empresas proveedoras de servicios tecnológicos y sus clientes. A través de estos sistemas se registran incidencias, consultas, solicitudes de soporte, requerimientos de desarrollo, problemas operativos y demás necesidades relacionadas con el uso de los sistemas de información. Debido al incremento constante de solicitudes, la clasificación manual de los tickets representa una actividad crítica que puede generar errores, inconsistencias y retrasos en la atención cuando depende exclusivamente del criterio del personal encargado de su gestión.

Webcoopec System es una empresa ecuatoriana dedicada al desarrollo e implementación de soluciones tecnológicas para cooperativas de ahorro y crédito, cajas de ahorro y demás organizaciones pertenecientes al sector financiero popular y solidario. Mediante la Resolución Nro. SEPS-INSEPS-IGS-DNGRT-2025-0214, emitida por la Superintendencia de Economía Popular y Solidaria (SEPS, 2025), la empresa fue calificada como prestadora de servicios auxiliares, lo que implica el cumplimiento de estándares

regulatorios pertinentes a la continuidad de los servicios tecnológicos, la accesibilidad de los sistemas y la respuesta oportuna por parte de las instituciones financieras a las que brinda soporte.

Como parte de sus procesos de atención al cliente, Webcoopec System desarrolló la plataforma Webcoopec Ticket tomando como base la solución de código abierto osTicket, incorporando funcionalidades adicionales orientadas a cumplir con los requerimientos operativos de la compañía y de los clientes. Este sistema permite registrar, administrar y dar seguimiento a las solicitudes realizadas por las cooperativas clientes, estableciéndose como el medio oficial de comunicación entre la empresa y las instituciones que utilizan sus soluciones informáticas.

Sin embargo, el aumento continuo en la cantidad de clientes y en el volumen de tickets registrados ha evidenciado limitaciones en el proceso de clasificación manual de las incidencias. Actualmente, la categorización de los tickets depende del conocimiento y experiencia del personal de soporte, situación que puede ocasionar asignaciones incorrectas, inconsistencias en la clasificación, incremento en los tiempos de respuesta, dificultades para identificar incidencias recurrentes y limitaciones en la creación de indicadores confiables para la toma de decisiones. Asimismo, una clasificación inadecuada puede afectar procesos administrativos relacionados con la correcta identificación de tickets sujetos a facturación y aquellos contemplados dentro de los contratos de soporte vigentes, impactando directamente en la gestión operativa y comercial de la organización.

Frente a esta problemática, el presente trabajo de investigación propone el desarrollo de un modelo de clasificación automática de tickets de soporte técnico mediante la aplicación de técnicas de Procesamiento de Lenguaje Natural (PLN) y Machine Learning, utilizando información histórica proveniente de la base de datos del sistema Webcoopec Ticket. El estudio incluye el análisis, la preparación y conversión de datos, así como el entrenamiento y

evaluación de diferentes algoritmos de clasificación supervisada, así como la integración del modelo seleccionado dentro del sistema de gestión de tickets diseñado para facilitar el proceso de categorización automática de nuevas incidencias.

En la ejecución del proyecto se aplicará la metodología CRISP-DM (Proceso Estándar Interindustrial para la Minería de Datos), la cual es ampliamente reconocida en el ámbito de la Ciencia de Datos por ofrecer un marco estructurado que facilita la comprensión del contexto empresarial, así como el análisis y preparación de los datos, la modelización, la evaluación y la implementación de soluciones basadas en datos (Chapman et al., 2000). Adicionalmente, los resultados obtenidos serán complementados mediante herramientas de Analítica de Datos que permitirán visualizar indicadores, tendencias y patrones identificados durante el proceso analítico, fortaleciendo la gestión de la mesa de ayuda y apoyando la toma de decisiones basada en datos.

En este aspecto, la presente investigación busca demostrar cómo la aplicación de técnicas modernas de Ciencia de Datos puede contribuir a optimizar la gestión de soporte técnico dentro de una organización del sector tecnológico, automatizando procesos, mejorando la calidad de la información generada y fortaleciendo la eficiencia operativa mediante el uso de modelos predictivos aplicados a datos reales de un entorno empresarial.

1.1.1 Planteamiento del problema

En años recientes, la transformación digital y la implementación de sistemas de información han incrementado considerablemente el volumen de solicitudes de soporte técnico gestionadas por las empresas desarrolladoras de software. Este incremento ha convertido a las mesas de ayuda en un componente estratégico para garantizar la continuidad operativa, la satisfacción del cliente y el cumplimiento de los acuerdos de nivel de servicio (SLA), especialmente en organizaciones que prestan servicios tecnológicos a entidades reguladas.

Webcoopec System, empresa ecuatoriana dedicada al desarrollo e implementación de soluciones tecnológicas para cooperativas de ahorro y crédito y demás entidades del Sector Financiero Popular y Solidario, utiliza la plataforma Webcoopec Ticket, basada en la herramienta de código abierto osTicket, como canal oficial para la recepción, seguimiento y gestión de las incidencias reportadas por sus clientes. A través de esta plataforma se registran diariamente consultas funcionales, errores del sistema, requerimientos de desarrollo, solicitudes comerciales, incidentes operativos y demás actividades relacionadas con el soporte técnico.

Actualmente, el personal de soporte clasifica los tickets de manera manual y asigna la categoría correspondiente de acuerdo con su experiencia y criterio. Este procedimiento presenta diversas limitaciones, ya que la interpretación del contenido de los tickets puede variar entre los diferentes analistas, generando inconsistencias en la categorización y dificultando la estandarización del proceso.

Una clasificación incorrecta puede provocar la asignación de tickets al área equivocada, incrementando los tiempos de respuesta y resolución, afectando la priorización de incidencias y reduciendo la eficiencia operativa de la mesa de ayuda. Del mismo modo, la ausencia de una clasificación homogénea limita la generación de indicadores confiables, dificulta el análisis histórico de las incidencias y restringe la identificación de patrones que permitan implementar acciones preventivas o de mejora continua.

Adicionalmente, la clasificación manual impacta procesos administrativos y comerciales de la organización. Determinados tickets corresponden a servicios contemplados dentro de los contratos de soporte vigentes, mientras que otros representan desarrollos adicionales o requerimientos que deben ser facturados al cliente. Una categorización inadecuada puede ocasionar errores en la identificación de estos servicios, afectando el

proceso de facturación, el control de los trabajos realizados y la correcta administración de los recursos destinados al soporte técnico.

A pesar de que Webcoopec System dispone de una base histórica con miles de tickets almacenados en su sistema de gestión, esta información no ha sido utilizada aún con métodos de Ciencia de Datos que posibiliten automatizar la clasificación de las incidencias y convertir los datos históricos en información valiosa para la organización. La existencia de esta información representa una oportunidad para aplicar técnicas de Procesamiento de Lenguaje Natural (PLN) y Machine Learning capaces de identificar patrones en los textos registrados por los usuarios y predecir automáticamente la categoría más adecuada para cada nuevo ticket.

En este contexto, se plantea la imperante necesidad de desarrollar un modelo automatizado de clasificación mediante técnicas de Machine Learning que optimice la categorización de tickets de soporte, reduzca los tiempos de atención, aumente la calidad de los datos ingresados, mejore el proceso de toma de decisiones basándose en el análisis de datos históricos y potencie la eficiencia operativa del servicio de asistencia técnica de Webcoopec System.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un modelo basado en técnicas de Procesamiento de Lenguaje Natural y Machine Learning que permita automatizar la clasificación de tickets de soporte técnico y optimizar la gestión operativa de la mesa de ayuda de Webcoopec System.

1.2.2 Objetivos específicos

Analizar los datos históricos almacenados en la base de datos del sistema de gestión de tickets para identificar variables relevantes y patrones asociados a las incidencias registradas.

Diseñar el proceso de preparación de datos utilizando técnicas de Procesamiento de Lenguaje Natural para transformar la información textual en variables aptas para el entrenamiento de modelos de Machine Learning.

Entrenar y evaluar diferentes algoritmos de clasificación supervisada para seleccionar el modelo con mejor desempeño mediante métricas como Accuracy, Precision, Recall y F1-Score.

Implementar el modelo seleccionado dentro del sistema de gestión de tickets y generar indicadores analíticos que apoyen la toma de decisiones y la mejora continua del proceso de soporte técnico.

1.3 Justificación

1.3.1 Importancia operativa y empresarial

Esta investigación se fundamenta en la necesidad de mejorar la gestión de la mesa de ayuda de Webcoopec System mediante la automatización de la clasificación de tickets de soporte técnico. Actualmente, la organización administra un volumen creciente de solicitudes provenientes de cooperativas de ahorro y crédito y demás entidades del Sector Financiero Popular y Solidario, las cuales requieren ser clasificadas oportunamente para garantizar una atención eficiente y la adhesión a los estándares de servicio fijados.

La incorporación de técnicas de Procesamiento de Lenguaje Natural (PLN) y Machine Learning permitirá reducir la dependencia de la clasificación manual, disminuir inconsistencias en la categorización de los tickets y apoyar la asignación adecuada de las incidencias hacia las áreas responsables. Asimismo, facilitará la identificación de patrones de comportamiento, incidencias recurrentes y tendencias operativas que podrán ser utilizadas para fortalecer los procesos de mejora continua dentro de la entidad.

Desde el panorama empresarial, la adopción de un sistema de clasificación automática ayudará a optimizar el uso de recursos de soporte técnico, elevar la calidad de la información

registrada y potenciar los procesos administrativos relacionados con la identificación de servicios sujetos a facturación y aquellos contemplados dentro de los contratos de soporte vigentes. De esta manera, la organización dispondrá de información más confiable para la formulación de decisiones y la planificación de acciones orientadas a incrementar la eficiencia de la mesa de ayuda.

1.3.2 Relevancia académica y tecnológica

Desde la perspectiva académica, la investigación aporta a la aplicación de técnicas de Ciencia de Datos, Procesamiento de Lenguaje Natural y Machine Learning en un contexto empresarial real, utilizando información histórica proveniente de un sistema de gestión de tickets de soporte técnico. El estudio permitirá examinar la conducta de los datos textuales, realizar procesos de preparación y transformación de la información, así como comparar diferentes algoritmos de clasificación supervisada para determinar el modelo con mejor desempeño mediante métricas ampliamente aceptadas, tales como Accuracy, Precision, Recall y F1-Score.

En el ámbito tecnológico, la investigación constituye un caso de aplicación práctica de la metodología CRISP-DM para el desarrollo de proyectos de Ciencia de Datos, integrando actividades relacionadas con la comprensión del negocio, exploración y preparación de datos, modelado, evaluación e implementación del modelo dentro de un sistema de información existente. Además, los resultados obtenidos pueden ser una referencia para futuras investigaciones sobre automatización de soporte técnico mediante inteligencia artificial y minería de texto.

1.3.3 Viabilidad del estudio

La presente investigación es técnicamente y operativamente viable debido a que Webcoopec System ha autorizado el desarrollo del proyecto utilizando información histórica proveniente de su sistema de gestión de tickets con fines exclusivamente académicos y de

investigación. Para garantizar la salvaguarda de la información, el manejo de los datos se llevará a cabo respetando los principios de confidencialidad y minimización de datos personales, empleando únicamente la información necesaria para el desarrollo del modelo de clasificación.

Asimismo, se dispone de acceso a la base de datos del sistema Webcoopec Ticket, lo que permitirá realizar directamente el proceso de extracción, análisis y planificación de los datos sin necesidad de crear un conjunto de datos artificial. Esto representa una ventaja significativa, ya que el modelo será entrenado utilizando información real generada durante la operación cotidiana de la organización.

Adicionalmente, el investigador dispone de los recursos tecnológicos y del conocimiento técnico necesario para desarrollar el proyecto, así como del acceso al entorno de desarrollo donde posteriormente se integrará el modelo dentro del sistema de gestión de tickets.

1.3.4 Relación con los objetivos de investigación

Este estudio se encuentra en concordancia con los objetivos establecidos, ya que tiene como finalidad desarrollar un modelo automatizado de clasificación que optimice el proceso de categorización de tickets de soporte técnico mediante el uso de técnicas de Procesamiento de Lenguaje Natural y Machine Learning. Los resultados obtenidos facilitarán la transformación de los datos históricos de la organización en conocimiento valioso para la toma de decisiones, mejorarán la gestión operativa de la mesa de ayuda e identificarán oportunidades de mejora a través del análisis de la información.

La investigación abarca no solo la creación de un modelo predictivo, sino que también integra un enfoque holístico de análisis de datos que permitirá comprender el comportamiento histórico de las incidencias, evaluar el desempeño de diferentes algoritmos de clasificación e implementar una solución aplicable dentro del entorno operativo de Webcoopec System.

1.4 Preguntas

1.4.1 Pregunta general

¿Cómo puede un modelo basado en técnicas de Procesamiento de Lenguaje Natural y Machine Learning mejorar la clasificación automática de tickets de soporte técnico y optimizar la gestión operativa de la mesa de ayuda de Webcoopec System?

1.4.2 Preguntas específicas

¿Qué características presentan los datos históricos almacenados en la base de datos del sistema de tickets y cuáles son relevantes para la clasificación automática?

¿Qué técnicas de Procesamiento de Lenguaje Natural permiten preparar adecuadamente la información textual para el entrenamiento del modelo?

¿Cuál de los algoritmos de Machine Learning evaluados presenta el mejor desempeño para clasificar automáticamente los tickets de soporte técnico?

¿Cómo contribuye la implementación del modelo de clasificación automática a mejorar la eficiencia operativa y la gestión de la mesa de ayuda de Webcoopec System?

CAPÍTULO II

MARCO TEÓRICO

2.1 Introducción del capítulo

Este capítulo busca fundamentar el sustento teórico, normativo y metodológico de la investigación titulada “Clasificación automática de tickets de soporte técnico basado en procesamiento de lenguaje natural aplicado a la gestión operativa de Webcoopec System”. Para ello, se revisan conceptos, antecedentes y enfoques relacionados con la gestión de mesas de ayuda, los sistemas de tickets, el procesamiento de lenguaje natural, la minería de texto, los modelos de Machine Learning y las metodologías utilizadas en proyectos de Ciencia de Datos.

La revisión teórica resulta necesaria porque el problema de investigación no se limita únicamente a la clasificación automática de textos, sino que se ubica dentro de un contexto empresarial y regulatorio específico. Webcoopec System presta servicios tecnológicos relacionados con software financiero y computación para entidades del sector financiero popular y solidario, por lo que la gestión de tickets de soporte técnico debe analizarse considerando aspectos operativos, normativos, técnicos y de protección de datos.

En este sentido, el marco teórico conecta tres dimensiones principales. La primera corresponde al contexto organizacional y normativo, donde se aborda el rol de Webcoopec System como proveedor tecnológico y la importancia del cumplimiento regulatorio en servicios auxiliares del sistema financiero. La segunda dimensión corresponde a la gestión operativa de soporte técnico, en la que se analiza la mesa de ayuda, la clasificación manual de tickets y la obligación de optimizar la eficiencia en la satisfacción de demandas. La tercera dimensión se relaciona con las técnicas de Ciencia de Datos, especialmente el procesamiento de lenguaje natural, la minería de texto y los modelos supervisados de Machine Learning aplicados a clasificación automática.

De esta manera, el capítulo permite justificar las decisiones metodológicas posteriores, como la selección de datos históricos de tickets, el uso de técnicas de limpieza y representación textual, la comparación de modelos de clasificación y la evaluación mediante métricas como accuracy, precision, recall, F1-score y matriz de confusión. Asimismo, la revisión de literatura permitirá identificar estudios previos, brechas de investigación y aportes esperados para el contexto específico de Webcoopec System.

2.2 Estrategia de revisión de literatura

La revisión de la literatura se llevó a cabo con el objetivo de identificar estudios académicos, fuentes técnicas, normativa aplicable y documentación especializada relacionada con la clasificación automática de tickets de soporte técnico mediante Procesamiento de Lenguaje Natural y Machine Learning. Esta estrategia permitió seleccionar fuentes pertinentes para sustentar el problema de investigación, las variables de análisis, la metodología, los modelos propuestos y las métricas de evaluación.

Para la búsqueda de información se consideraron bases académicas y fuentes especializadas como Google Scholar, Scopus, IEEE Xplore, ACM Digital Library, ScienceDirect, SciELO, arXiv y documentación oficial de herramientas tecnológicas relacionadas con sistemas de mesa de ayuda. Adicionalmente, se revisaron fuentes normativas ecuatorianas aplicables al contexto de Webcoopec System, especialmente aquellas relacionadas con servicios auxiliares del sistema financiero popular y solidario, protección de datos personales y cumplimiento regulatorio.

La búsqueda se organizó mediante términos en español e inglés, considerando que gran parte de la literatura técnica sobre clasificación automática de texto y sistemas de soporte se encuentra publicada en inglés. Las combinaciones utilizadas incluyeron términos asociados al fenómeno de estudio, contexto empresarial, técnicas de procesamiento textual, modelos de clasificación y métricas de evaluación.

Tabla 2

Base de datos y fuentes consultadas

Fuente	Justificación de uso	Tipo de fuente
Google Scholar	Permite acceder a artículos, tesis, libros y ponencias académicas relacionadas con Ciencia de Datos, PLN y Machine Learning.	Académica
Scopus / Web of Science	Bases multidisciplinarias útiles para identificar literatura científica revisada por pares.	Académica
IEEE Xplore / ACM Digital Library	Fuentes relevantes para investigaciones en computación, inteligencia artificial, ingeniería de software y sistemas de información.	Académica
SciELO	Repositorio útil para literatura científica latinoamericana y estudios en español.	Académica
arXiv	Fuente de prepublicaciones técnicas recientes sobre aprendizaje automático, modelos de lenguaje y clasificación de tickets.	Académica/técnica
Documentación oficial de osTicket	Permite contextualizar el funcionamiento de la herramienta de mesa de ayuda utilizada para la gestión de tickets.	Técnica
Superintendencia de Economía Popular y Solidaria	Permite sustentar el contexto normativo de Webcoopec System como proveedor de servicios auxiliares de software financiero y computación.	Normativa
Ley Orgánica de Protección de Datos Personales	Permite fundamentar el tratamiento responsable de datos personales contenidos en tickets de soporte.	Legal

Tabla 3

Cadenas de Búsqueda Utilizadas

Bloque	Términos de búsqueda
Fenómeno	"clasificación automática de tickets", "clasificación de tickets de soporte", "automatic ticket classification", "support ticket classification"
Contexto	"mesa de ayuda", "help desk", "service desk", "customer support tickets", "technical support tickets"
Método	"procesamiento de lenguaje natural", "NLP", "text mining", "machine learning", "supervised classification"
Modelos	"Naive Bayes", "Support Vector Machine", "Random Forest", "TF-IDF", "text classification"
Evaluación	"accuracy", "precision", "recall", "F1-score", "confusion matrix"
Cadena final	("automatic ticket classification" OR "support ticket classification") AND ("natural language processing" OR NLP OR "text mining") AND ("machine learning" OR "supervised classification")

Se establecieron criterios de inclusión y exclusión para la selección de las fuentes, con el objetivo de asegurar su relevancia, actualidad y calidad académica. Se priorizaron estudios publicados entre 2018 y 2026, sin excluir fuentes seminales necesarias para explicar conceptos fundamentales como CRISP-DM, TF-IDF o clasificación supervisada. También se incluyeron documentos técnicos y normativos cuando aportaban al contexto específico de la investigación.

Tabla 4

Criterios de inclusión y exclusión de fuentes

Criterio	Incluir	Excluir
Temporalidad	Estudios recientes publicados preferentemente entre 2018 y 2026.	Estudios desactualizados sin aporte conceptual o metodológico claro.

Criterio	Incluir	Excluir
Pertinencia	Trabajos relacionados con NLP, minería de texto, clasificación automática, help desk, service desk o gestión de tickets.	Textos generales que no se conectan con el problema, variables, metodología o modelos del estudio.
Calidad	Artículos científicos, ponencias, libros académicos, tesis, documentación oficial y normativa vigente.	Blogs de opinión, documentos sin autor identificado o fuentes sin respaldo verificable.
Aplicabilidad	Estudios con datos, modelos, métricas o contextos comparables al análisis de tickets de soporte.	Estudios con contextos no transferibles o sin relación con clasificación textual.
Idioma	Publicaciones en español o inglés.	Fuentes en otros idiomas sin traducción verificable.

Como apoyo al proceso de organización bibliográfica se recomienda utilizar Zotero, debido a que permite almacenar referencias, adjuntar archivos PDF, insertar citas en Microsoft Word y generar bibliografía en formato APA 7. De manera complementaria, herramientas como NotebookLM pueden utilizarse para consultar, comparar y sintetizar documentos previamente cargados; sin embargo, la selección e interpretación de fuentes debe mantenerse bajo criterio académico del investigador.

De forma preliminar, la revisión se enfocará en tres grupos de literatura: primero, estudios sobre clasificación automática de tickets y sistemas de soporte; segundo, trabajos sobre metodologías de Ciencia de Datos, especialmente CRISP-DM; y tercero, fuentes legales y normativas relacionadas con el tratamiento de datos, servicios tecnológicos y cumplimiento regulatorio en el contexto ecuatoriano. Esta estrategia permitirá construir un marco teórico trazable, actualizado y directamente alineado con la pregunta de investigación y los objetivos del estudio.

2.3 Antecedentes de investigación

La automatización en la clasificación de tickets de soporte técnico es ahora una investigación clave en Ciencia de Datos, por el aumento de solicitudes que reciben las organizaciones a través de mesas de ayuda, plataformas de soporte y sistemas de atención al cliente. En estos entornos, los tickets suelen contener información textual no estructurada, como asuntos, descripciones, comentarios y respuestas del personal técnico, lo que permite aplicar técnicas de Procesamiento de Lenguaje Natural y Machine Learning para automatizar tareas de clasificación, asignación, priorización o escalamiento.

Uno de los antecedentes más relevantes es el trabajo de Molino et al. (2018), quienes propusieron COTA, un sistema diseñado para optimizar la rapidez y exactitud del servicio al cliente usando métodos de clasificación automática y elección de respuestas. Este estudio es importante porque demuestra que los tickets de soporte pueden ser tratados como datos textuales clasificables, y que la automatización puede generar beneficios operativos medibles. Los autores compararon enfoques basados en ingeniería de características y modelos de aprendizaje profundo, evidenciando que la clasificación de tickets puede apoyar la reducción de tiempos de resolución sin afectar negativamente la satisfacción del usuario. Para la presente investigación, este antecedente resulta pertinente porque conecta directamente el análisis de tickets con mejoras en la gestión operativa de una mesa de ayuda.

De manera similar, Feng et al. (2022) desarrollaron TaDaa, una propuesta de asignación automática de tickets en tiempo real para sistemas de soporte, help desk e issue tracking. A diferencia de enfoques centrados únicamente en clasificar una categoría, este trabajo aborda la asignación del ticket al grupo o resolutor adecuado, lo que representa una evolución práctica de la clasificación automática. El estudio muestra que los modelos de Machine Learning pueden apoyar no solo la categorización inicial, sino también la derivación eficiente de incidentes dentro de una organización. Este enfoque es relevante para Webcoopec

System, debido a que la clasificación correcta de los tickets puede facilitar su asignación al área responsable y reducir errores derivados de la intervención manual.

Otro aporte reciente es el estudio de Truss y Boehm (2024), quienes analizaron la clasificación de tickets de soporte mediante inteligencia artificial y AutoML. Su investigación destaca que la automatización de la clasificación de tickets es clave para mejorar el rendimiento del soporte y reducir los tiempos de atención. Un elemento importante de este antecedente es que demuestra la posibilidad de construir modelos de clasificación incluso en organizaciones que no cuentan con equipos especializados de inteligencia artificial, mediante herramientas de AutoML. Si bien esta tesis no aborda AutoML, la investigación respalda la posibilidad de implementar soluciones de clasificación automática en contextos empresariales donde la eficiencia operativa es un objetivo estratégico.

En una línea más reciente, Liu et al. (2025) propusieron TickIt, un marco basado en modelos de lenguaje de gran escala para el escalamiento automático de tickets en servicios cloud. Este trabajo evidencia que los sistemas modernos de soporte técnico enfrentan desafíos relacionados con el volumen, la complejidad y la actualización dinámica de los tickets. A diferencia de modelos tradicionales que clasifican documentos de manera estática, TickIt considera relaciones entre tickets, temas, estados cambiantes y asignación hacia equipos adecuados. Este antecedente permite observar que la clasificación automática de tickets se está desplazando hacia enfoques más inteligentes y contextuales; sin embargo, también evidencia que muchas soluciones avanzadas requieren infraestructura, datos de alta calidad y capacidad técnica considerable.

Además de los estudios centrados directamente en tickets, la literatura sobre clasificación automática de texto proporciona fundamentos metodológicos importantes para esta investigación. Das et al. (2023) analizaron el uso de TF-IDF como método de ponderación de atributos en datos no estructurados, comparando diferentes algoritmos como

SVM, Naive Bayes, Random Forest y otros clasificadores. Sus resultados respaldan el uso de TF-IDF como técnica de representación textual en problemas de clasificación, debido a que permite transformar textos en vectores numéricos interpretables por los modelos de Machine Learning. Este antecedente es relevante porque en la presente investigación se propone utilizar TF-IDF para representar el contenido textual de los tickets de soporte técnico.

En esta misma línea, Oancea (2025) comparó distintos métodos de representación textual y algoritmos de Machine Learning para clasificación automática, incluyendo TF-IDF, Word2Vec, FastText, Naive Bayes, SVM, Random Forest y Árboles de Decisión. El estudio concluye que la selección del modelo y la técnica de representación textual impactan directamente en el rendimiento de la clasificación. Para esta tesis, dicho antecedente sustenta la necesidad de comparar varios modelos y no asumir de forma anticipada cuál será el mejor algoritmo. Por esta razón, se propone evaluar Naive Bayes, Random Forest y SVM utilizando métricas como accuracy, precision, recall y F1-score.

Desde una perspectiva metodológica, Sánchez Trujillo y Pérez Hernández (2021) aplicaron la metodología CRISP-DM en un proyecto de minería de datos orientado al análisis de enfermedades dermatológicas. Aunque el contexto de aplicación es distinto al de una mesa de ayuda, el estudio demuestra la utilidad de CRISP-DM para estructurar proyectos de Ciencia de Datos desde entender el problema hasta la preparación, modelado, evaluación e interpretación de resultados. Este antecedente es importante porque permite justificar el uso de CRISP-DM como metodología técnica para organizar el desarrollo de la tesis, especialmente considerando que el proyecto requiere extracción de datos históricos, limpieza, análisis, entrenamiento de modelos y evaluación.

En el contexto latinoamericano, también resulta pertinente considerar investigaciones que reflexionan sobre la datificación de fuentes y el uso de Ciencia de Datos para transformar información documental en datos analizables. Manrique-Gómez y Borja Gómez (2025)

plantean que los procesos de datificación permiten convertir fuentes originalmente cualitativas o textuales en estructuras analizables mediante técnicas computacionales. A pesar de que su estudio se sitúa en la historia, su aporte conceptual es útil para esta tesis porque los tickets de soporte técnico también constituyen registros textuales generados por la operación diaria de una organización. En este sentido, la investigación propuesta convierte los tickets históricos en una fuente de datos para identificar patrones, categorías, recurrencias y oportunidades de mejora.

Los antecedentes revisados permiten identificar coincidencias importantes. En primer lugar, varios estudios reconocen que la clasificación manual de tickets puede generar demoras, inconsistencias y sobrecarga operativa. En segundo lugar, existe consenso en que las técnicas de NLP y Machine Learning permiten transformar el texto de los tickets en representaciones numéricas útiles para entrenar modelos de clasificación. En tercer lugar, la literatura muestra que no existe un único modelo universalmente superior, por lo que la comparación entre algoritmos sigue siendo una práctica necesaria para seleccionar la alternativa más adecuada según el contexto, los datos y las categorías disponibles.

Sin embargo, también se observan limitaciones en los estudios previos. Algunas investigaciones se desarrollan en grandes empresas tecnológicas o plataformas cloud con volúmenes de datos y recursos superiores a los disponibles en organizaciones medianas. Otros trabajos se enfocan en asignación de resolutores, selección de respuestas o escalamiento avanzado, pero no necesariamente en la realidad operativa de empresas latinoamericanas proveedoras de software financiero. Además, la mayoría de estudios se concentra en contextos internacionales, por lo que existe una oportunidad de aplicar estos enfoques en una organización ecuatoriana real, regulada por condiciones específicas del sector financiero popular y solidario.

En consecuencia, los antecedentes revisados justifican la pertinencia de la presente investigación. La clasificación automática de tickets mediante PLN y Machine Learning no solo tiene respaldo académico y técnico, sino que además responde a una necesidad operativa concreta de Webcoopec System: reducir la dependencia de la clasificación manual, mejorar la consistencia en la categorización, identificar patrones recurrentes y facilitar el proceso de toma de decisiones en la mesa de ayuda.

Tabla 5

Matriz de Antecedentes Relevantes

Autor/año	Contexto	Método/modelo	Hallazgo principal	Aporte a la tesis
Molino et al. (2018)	Soporte al cliente	NLP, clasificación y deep learning	La automatización de tickets puede reducir tiempos de resolución.	Sustenta la relación entre clasificación automática y eficiencia operativa.
Feng et al. (2022)	Help desk e issue tracking	Transformers y asignación automática	La IA puede sugerir grupos o resolutores adecuados.	Relaciona clasificación con asignación operativa.
Truss y Boehm (2024)	Customer support	AutoML para clasificación de tickets	AutoML puede generar modelos útiles para clasificar tickets.	Refuerza la viabilidad empresarial de automatizar la clasificación.
Liu et al. (2025)	Servicios cloud	LLM para escalamiento de tickets	Los modelos de lenguaje permiten escalamiento dinámico y contextual.	Evidencia tendencias recientes en soporte técnico automatizado.

Autor/año	Contexto	Método/modelo	Hallazgo principal	Aporte a la tesis
Das et al. (2023)	Clasificación textual	TF-IDF y modelos ML	TF-IDF mejora la representación de texto no estructurado.	Justifica el uso de TF-IDF en tickets.
Oancea (2025)	Clasificación automática de texto	TF-IDF, embeddings y modelos ML	El desempeño depende de la representación textual y del modelo.	Justifica comparar Naive Bayes, SVM y Random Forest.
Sánchez Trujillo y Pérez Hernández (2021)	Proyecto Data Mining	CRISP-DM	CRISP-DM estructura proyectos de minería de datos en fases.	Sustenta la metodología técnica del proyecto.
Manrique-Gómez y Borja Gómez (2025)	Datificación de fuentes	Ciencia de datos aplicada a fuentes textuales	Las fuentes documentales pueden convertirse en datos analizables.	Justifica transformar tickets históricos en datos para análisis.

2.4 Bases conceptuales

Los fundamentos conceptuales forman la base teórica de esta investigación. En este apartado se describen los conceptos, tecnologías, metodologías y modelos que sustentan el diseño del sistema de clasificación automática de tickets de soporte técnico mediante técnicas de Procesamiento de Lenguaje Natural (PLN) y Machine Learning.

Debido a que la investigación se desarrolla en un entorno empresarial real, resulta necesario integrar conceptos relacionados con la gestión de servicios de soporte técnico, los sistemas de mesa de ayuda, la Ciencia de Datos, la minería de texto, los algoritmos de aprendizaje supervisado y el tratamiento responsable de los datos. Estos elementos permiten

comprender tanto el contexto organizacional donde surge el problema como las herramientas tecnológicas utilizadas para proponer una solución basada en inteligencia artificial.

Asimismo, las bases conceptuales permiten establecer una relación entre los antecedentes científicos revisados y la metodología propuesta para la investigación, garantizando coherencia entre el problema identificado, los objetivos planteados y las técnicas que serán implementadas durante el desarrollo del proyecto.

2.4.1 Webcoopec System como proveedor de servicios tecnológicos

La presente investigación se desarrolla en Webcoopec System Cía. Ltda., empresa ecuatoriana dedicada al desarrollo de software especializado para cooperativas de ahorro y crédito, cajas de ahorro, asociaciones y demás entidades pertenecientes al sector financiero popular y solidario. Entre sus principales servicios se encuentran el desarrollo de sistemas financieros, soluciones de Analítica de Datos, infraestructura tecnológica, servicios de hospedaje, soporte técnico y mantenimiento de plataformas informáticas.

Como parte de sus actividades, la empresa mantiene una mesa de ayuda destinada a registrar, gestionar y dar seguimiento a las solicitudes de soporte técnico realizadas por sus clientes. Estas solicitudes se almacenan en forma de tickets que contienen información textual relacionada con incidentes, requerimientos, consultas o solicitudes de mejora sobre los diferentes módulos del sistema financiero desarrollado por la organización.

El crecimiento del número de clientes y la diversidad de requerimientos han incrementado el volumen de tickets gestionados diariamente, generando la necesidad de optimizar los procesos de clasificación y asignación inicial de las solicitudes. Actualmente, esta actividad es ejecutada manualmente por el equipo de soporte, lo que puede ocasionar diferencias de criterio, errores de categorización y tiempos adicionales durante la atención de los requerimientos.

Desde el punto de vista regulatorio, Webcoopec System se encuentra calificada por la Superintendencia de Economía Popular y Solidaria (SEPS) como compañía de servicios auxiliares especializada en software financiero y computación, condición que implica el cumplimiento de disposiciones técnicas y operativas relacionadas con la prestación de servicios tecnológicos dirigidos a entidades del sector financiero popular y solidario. Este contexto refuerza la importancia de contar con procesos de soporte técnico eficientes, trazables y alineados con buenas prácticas de gestión de servicios.

En este escenario, la automatización de la clasificación de tickets mediante técnicas de Procesamiento de Lenguaje Natural y Machine Learning representa una alternativa tecnológica que puede ayudar a incrementar la eficiencia de la mesa de ayuda, perfeccionar la asignación de solicitudes y facilitar decisiones fundamentadas en datos.

2.4.2 Gestión de soporte técnico y sistemas de mesa de ayuda

La gestión de soporte técnico constituye un proceso fundamental dentro de las organizaciones que desarrollan y mantienen soluciones tecnológicas, ya que permite registrar, controlar y resolver incidentes, consultas, solicitudes de servicio y requerimientos reportados por los usuarios. De acuerdo con las buenas prácticas de ITIL, el Service Desk funciona como punto central de contacto entre el proveedor de servicios y los usuarios, facilitando la comunicación y la gestión de solicitudes de soporte (PeopleCert, s. f.).

Dentro de este proceso, el **ticket de soporte** representa la unidad básica de gestión. Cada ticket almacena información relacionada con el incidente reportado, incluyendo asunto, descripción, categoría, prioridad, estado, fechas, responsable y demás atributos necesarios para gestionar el ciclo de vida de la petición. La correcta clasificación de estos tickets constituye una actividad crítica, debido a que determina el área responsable de su atención y afecta directamente los tiempos de respuesta y resolución.

En organizaciones donde el volumen de solicitudes es elevado, la clasificación manual puede generar inconsistencias entre analistas, errores de categorización y retrasos en la asignación inicial de los requerimientos. En este contexto, la aplicación de técnicas de Inteligencia Artificial y Procesamiento de Lenguaje Natural permite automatizar este proceso, mejorando el desempeño del soporte y reduciendo los tiempos de resolución (Truss & Boehm, 2024).

2.4.3 osTicket como plataforma de gestión de tickets

La presente investigación utiliza como fuente de información los registros almacenados en **osTicket**, plataforma de código abierto ampliamente utilizada para la administración de mesas de ayuda y soporte técnico. Este sistema permite registrar incidentes, administrar usuarios, controlar estados, definir prioridades, categorizar solicitudes y mantener el historial completo de atención de cada ticket (osTicket, s. f.).

Para la investigación, osTicket constituye la fuente principal de datos históricos que serán utilizados durante el proceso de entrenamiento de los modelos de Machine Learning. El contenido textual almacenado en los campos asunto y descripción representa la principal fuente de información para aplicar técnicas de Procesamiento de Lenguaje Natural, mientras que la categoría asignada históricamente por los analistas servirá como etiqueta supervisada para el entrenamiento de los modelos de clasificación.

2.4.4 Procesamiento de lenguaje natural y minería de texto

El Procesamiento de Lenguaje Natural (PLN) es una rama de la Inteligencia Artificial centrada en el análisis y entendimiento automático del lenguaje humano. Su propósito consiste en transformar información textual no estructurada en representaciones que puedan ser interpretadas por algoritmos computacionales (Jurafsky & Martin, 2026).

Dentro de la presente investigación, el PLN permitirá procesar el contenido textual de los tickets mediante etapas de limpieza, normalización, eliminación de palabras vacías

(*stopwords*), tokenización, lematización y representación vectorial utilizando la técnica **TF-IDF** (*Term Frequency–Inverse Document Frequency*). Esta representación facilita que los algoritmos de Machine Learning identifiquen patrones lingüísticos asociados a las categorías de soporte (Das et al., 2023).

La minería de texto complementa este proceso al permitir descubrir relaciones, tendencias y patrones ocultos dentro de grandes volúmenes de documentos, convirtiéndose en una herramienta ampliamente utilizada para la clasificación automática de información textual.

2.4.5 Machine Learning aplicado a la clasificación automática de tickets

El Machine Learning constituye una disciplina de la Inteligencia Artificial que permite a los sistemas encontrar patrones en los datos, desarrollar pronósticos o clasificaciones sin necesidad de especificar exhaustivamente las condiciones decisionales (Bishop, 2006).

Este estudio emplea un método de **aprendizaje supervisado**, porque los tickets históricos ya disponen de una categoría previamente asignada por los analistas de soporte. A partir de esta información, los algoritmos aprenderán la relación existente entre el contenido textual y la categoría correspondiente.

Se propone comparar el desempeño de tres algoritmos ampliamente utilizados en clasificación de texto: Naive Bayes, reconocido por su eficiencia computacional; Support Vector Machine (SVM), caracterizado por su capacidad para trabajar con espacios de alta dimensionalidad; y Random Forest, estructura de combinación que integra varios árboles de decisión para aumentar la precisión predictiva y minimizar el sobreajuste. La selección del modelo final se realizará con base en métricas objetivas de desempeño.

2.4.6 Metodología CRISP-DM y evaluación del modelo

Para desarrollar el proyecto, se usará la metodología CRISP-DM (Proceso Estandarizado Interindustrial para la Minería de Datos), la cual se caracteriza como una de las más reconocidas en la Ciencia de Datos, ya que permite ofrecer un proceso estructurado y se divide en seis pasos: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación e implementación (Chapman et al., 2000).

La evaluación de los modelos de clasificación se realizará mediante métricas ampliamente aceptadas en problemas de clasificación supervisada, entre ellas Accuracy, Precision, Recall, F1-Score y la Matriz de Confusión, las cuales permitirán comparar objetivamente el desempeño de cada algoritmo y seleccionar la alternativa con mayor capacidad para clasificar correctamente los tickets de soporte (Han et al., 2022).

2.4.7 Protección de datos personales

La base de datos empleada en este estudio incluye información generada durante la operación de la mesa de ayuda de Webcoopec System, por lo que el manejo de los datos deberá hacerse de acuerdo con los principios de la Ley Orgánica de Protección de Datos Personales (2021). En consecuencia, previo al análisis, los datos personales que permitan identificar directa o indirectamente a clientes, usuarios o colaboradores serán anonimizados, garantizando la confidencialidad de la información utilizada con fines exclusivamente académicos.

Asimismo, la investigación respetará los principios de juridicidad, finalidad, pertinencia y minimización de datos personales, confidencialidad y responsabilidad proactiva y demostrada establecidos en la Ley Orgánica de Protección de Datos Personales (2021), garantizando que el manejo de los tickets se restrinja al desarrollo del modelo de clasificación automática y no viole los derechos de los titulares de datos.

2.5 Fundamentación teórica y modelos de referencia

2.5.1 Metodología CRISP-DM

La presente investigación adopta la metodología CRISP-DM (Cross Industry Standard Process for Data Mining) como fundamento orientador para la elaboración del proyecto de Ciencia de Datos. Esta metodología fue desarrollada con el propósito de proporcionar un proceso estructurado, iterativo e independiente de la industria para proyectos basados en minería de datos, permitiendo organizar de manera sistemática las acciones desde entender el problema hasta aplicar y evaluar los modelos (Chapman et al., 2000).

El modelo CRISP-DM está constituido por seis etapas: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación e implementación. Estas etapas mantienen una relación iterativa que permite retroalimentar continuamente el proyecto conforme se identifican nuevos hallazgos durante el análisis.

En la presente investigación, la fase de comprensión del negocio permitirá analizar la problemática existente en la clasificación manual de tickets dentro de la mesa de ayuda de Webcoopec System. Posteriormente, durante la comprensión y preparación de los datos, se realizará la remoción de los tickets históricos, el tratamiento del texto mediante técnicas de Procesamiento de Lenguaje Natural y la generación del conjunto de datos para entrenamiento.

La fase de modelado comprenderá la formulación y evaluación de diversos algoritmos de clasificación supervisada, mientras que la evaluación permitirá determinar cuál presenta el mejor desempeño utilizando métricas objetivas. Finalmente, la metodología concluye con la documentación e interpretación de los resultados obtenidos, permitiendo establecer la viabilidad de implementar un modelo de clasificación automática de tickets dentro de la organización.

Debido a su amplia aceptación en proyectos de Ciencia de Datos y Machine Learning, CRISP-DM desarrolla una metodología apropiada para estructurar el progreso de esta investigación, garantizando la coherencia entre los objetivos definidos, los datos utilizados y los resultados esperados.

2.5.2 Procesamiento de lenguaje natural

El Procesamiento de Lenguaje Natural (PLN) constituye el fundamento tecnológico de la presente investigación, debido a que permite analizar y transformar información textual en formatos que son susceptibles de ser analizados por algoritmos de aprendizaje automático. En el contexto de los sistemas de soporte técnico, los tickets contienen información no estructurada escrita por los usuarios, por lo que resulta necesario aplicar técnicas de limpieza, normalización y representación textual antes del entrenamiento de los modelos (Jurafsky & Martin, 2026).

Para esta investigación se empleará la representación TF-IDF (Term Frequency–Inverse Document Frequency), ampliamente empleada en problemas de clasificación de texto por su capacidad para ponderar la importancia de cada término dentro del conjunto documental. Esta técnica facilita que los algoritmos identifiquen patrones relevantes asociados a cada categoría de soporte.

2.5.3 Aprendizaje supervisado y modelos de clasificación

El problema abordado en esta investigación corresponde a una tarea de clasificación supervisada, ya que los tickets históricos cuentan con categorías previamente asignadas por el personal de soporte. A partir de estos datos etiquetados, los algoritmos aprenden patrones que posteriormente permiten clasificar nuevas solicitudes de forma automática (Bishop, 2006).

Con el propósito de identificar el modelo con mejor desempeño, se evaluarán tres algoritmos ampliamente utilizados en clasificación textual: Naive Bayes, Support Vector Machine (SVM) y Random Forest. La comparación permitirá determinar cuál presenta

mejores resultados para el contexto específico de Webcoopec System, considerando las características del conjunto de datos y las categorías de soporte definidas.

Considerando que el problema se clasifica como una tarea de clasificación multiclase, el modelo deberá distinguir entre cinco categorías funcionales de soporte técnico (Comercial, Funcional, Operativo, Operativo para cobrar y Software), aprendiendo patrones lingüísticos específicos asociados a cada una de ellas. Esta característica implica la utilización de algoritmos capaces de resolver problemas de clasificación multiclase y la evaluación mediante métricas apropiadas para este tipo de escenarios.

2.5.4 Evaluación del desempeño del modelo

La evaluación del modelo constituye una etapa esencial para verificar su capacidad de clasificación sobre datos no utilizados durante el entrenamiento. En esta investigación se emplearán métricas ampliamente aceptadas en problemas de clasificación supervisada: Accuracy, Precision, Recall y F1-Score, complementadas con la matriz de confusión para analizar el comportamiento de cada categoría (Han et al., 2022).

Estas métricas facilitarán la comparación objetiva de los modelos propuestos, permitiendo así la elección de la opción que presente el mejor balance entre capacidad predictiva, exactitud en la clasificación y capacidad de generalización.

2.5.5 Relación del marco teórico con la investigación

Los fundamentos teóricos revisados permiten establecer la relación entre el problema identificado en la clasificación manual de tickets y la solución tecnológica propuesta. La integración de conceptos asociados a la gestión de soporte técnico, los sistemas de mesa de ayuda, el Procesamiento de Lenguaje Natural, la metodología CRISP-DM y los algoritmos de Machine Learning proporciona el soporte conceptual necesario para desarrollar un modelo de clasificación automática encaminado a optimizar la gestión operativa de Webcoopec System.

En consecuencia, el marco teórico no solo sustenta las decisiones metodológicas adoptadas durante la investigación, sino que también evidencia que la aplicación de técnicas de Ciencia de Datos constituye una alternativa viable para mejorar la eficiencia y consistencia en la categorización de tickets de soporte técnico.

2.6 Variables, dimensiones y categorías de análisis

Las variables de esta investigación se dividen en dos categorías. El primero corresponde a las variables utilizadas durante el análisis descriptivo de la información histórica registrada en la mesa de ayuda, cuyo propósito es comprender el comportamiento operativo del servicio de soporte técnico. El segundo grupo corresponde a las variables empleadas para el desarrollo y análisis del modelo de Machine Learning orientado a la clasificación automática de tickets.

Esta separación permite mantener coherencia con la metodología CRISP-DM. Inicialmente, se llevará a cabo una fase de comprensión y análisis exploratorio de los datos, la cual será seguida por la preparación, modelado y evaluación del algoritmo de clasificación.

2.6.1 Variables para el análisis descriptivo

En la fase de análisis de datos se estudiarán diversas variables provenientes de la base de datos del sistema Webcoopec Ticket con el objetivo de identificar patrones de comportamiento, distribución de incidencias, carga operativa y características generales de los tickets registrados.

Estas variables permitirán construir indicadores estadísticos y tableros de control que servirán como apoyo para comprender el problema de negocio antes del desarrollo del modelo predictivo.

Inicialmente se consideran las siguientes variables:

- Cliente
- Usuario que registra el ticket

- Agente asignado
- Departamento
- Categoría
- Prioridad
- Estado
- Fecha de creación
- Fecha de cierre
- Tiempo de resolución
- Tipo de ticket
- Cantidad de respuestas
- Tiempo de primera respuesta
- Tickets facturados
- Tickets no facturados

No obstante, durante la fase de comprensión de los datos de la metodología CRISP-DM se realizará un análisis específico de la estructura de la base de datos para reconocer las variables disponibles y seleccionar aquellas que aporten mayor valor para el análisis descriptivo y la construcción de indicadores.

Los resultados obtenidos permitirán elaborar visualizaciones mediante Tableau, facilitando la identificación de tendencias, comportamiento de usuarios, distribución de categorías, desempeño de agentes, tiempos de atención y demás indicadores relacionados con la gestión de la mesa de ayuda.

2.6.2 Variables de entrada del modelo de Machine Learning

Las variables de entrada corresponden a la información que será utilizada por los algoritmos de aprendizaje supervisado para aprender los patrones existentes entre el contenido de los tickets y su categoría funcional.

Debido a que el problema corresponde a una tarea de clasificación de texto, las principales variables de entrada estarán conformadas por el contenido textual registrado por los usuarios dentro de cada ticket.

Inicialmente se consideran los siguientes atributos:

- Asunto del ticket.
- Descripción del ticket.

De manera complementaria, durante el análisis exploratorio se evaluará la incorporación de otras variables presentes en la base de datos, como prioridad, departamento o tipo de ticket, siempre que su inclusión contribuya a mejorar el desempeño del modelo.

Previo al entrenamiento, el contenido textual será sometido a un proceso de limpieza y transformación mediante técnicas de Procesamiento de Lenguaje Natural, permitiendo transformar los textos en formatos numéricos que puedan ser manejados por los algoritmos de Machine Learning.

2.6.3 Variable de salida

La variable de salida corresponde a la categoría funcional que el modelo deberá predecir automáticamente para cada nuevo ticket registrado en la plataforma Webcoopec Ticket.

Con el propósito de entrenar un modelo representativo de los procesos operativos de la organización, únicamente se utilizarán las categorías actualmente vigentes dentro de la empresa.

Las categorías consideradas son:

Comercial

Solicitudes relacionadas con cotizaciones, proformas, adquisición de nuevos módulos, contratación de servicios y demás requerimientos comerciales.

Funcional

Consultas relacionadas con el funcionamiento del sistema, capacitaciones, parametrización, asistencia funcional y orientación en el uso de los diferentes módulos.

Operativo

Incidencias ocasionadas por errores de operación del usuario, configuraciones incorrectas, ingreso inadecuado de información o problemas derivados del uso normal del sistema.

Operativo para cobrar

Incidencias operativas cuya resolución genera facturación por concepto de soporte técnico bajo la modalidad de hora-hombre, de acuerdo con las condiciones contractuales establecidas con determinados clientes.

Software

Incidencias ocasionadas por errores de programación, fallos funcionales o defectos del software que requieren intervención del área de desarrollo.

Las categorías No definido y Otros no serán utilizadas para el entrenamiento del modelo. La primera corresponde principalmente a registros históricos generados antes de la implementación del proceso formal de categorización de tickets, mientras que la segunda representa una categoría obsoleta que dejó de utilizarse dentro de la organización.

2.6.4 Variables derivadas del procesamiento

Previo al entrenamiento del modelo será necesario transformar la información textual en variables cuantitativas que puedan ser analizadas por los algoritmos de aprendizaje supervisado.

Se utilizarán técnicas de Procesamiento de Lenguaje Natural, que incluyen:

- Transformación del texto a minúsculas.
- Eliminación de caracteres especiales.
- Eliminación de palabras vacías (stopwords).

- Tokenización.
- Lematización.
- Representación vectorial a través de TF-IDF.

Estas variables constituyen las características (*features*) utilizadas durante el entrenamiento de los modelos de clasificación.

2.6.5 Variables de evaluación

El desempeño de los modelos será evaluado mediante indicadores ampliamente utilizados en problemas de clasificación supervisada.

Las métricas consideradas son:

- Accuracy.
- Precision.
- Recall.
- F1-Score.
- Matriz de Confusión.
- Curva ROC y AUC (cuando el algoritmo lo permita).

Estas métricas permitirán comparar objetivamente el desempeño de los algoritmos evaluados y seleccionar el modelo con mayor capacidad de generalización.

Tabla 6

Variables de la investigación

Grupo	Variable	Tipo	Finalidad
Análisis descriptivo	Cliente	Entrada	Estadísticas y dashboards
	Usuario	Entrada	Estadísticas y dashboards
	Agente	Entrada	Estadísticas y dashboards
	Departamento	Entrada	Estadísticas y dashboards
	Estado	Entrada	Estadísticas y dashboards

Grupo	Variable	Tipo	Finalidad
	Prioridad	Entrada	Estadísticas y dashboards
	Fecha creación	Entrada	Estadísticas y dashboards
	Fecha cierre	Entrada	Estadísticas y dashboards
	Tiempo resolución	Entrada	Estadísticas y dashboards
	Categoría	Entrada	Estadísticas y dashboards
Machine Learning	Asunto	Entrada	Entrenamiento del modelo
	Descripción	Entrada	Entrenamiento del modelo
	Categoría	Salida	Etiqueta que el modelo debe predecir
PLN	TF-IDF	Derivada	Vectorización del texto
	Tokenización	Derivada	Preparación del texto
	Lematización	Derivada	Preparación del texto
Evaluación	Accuracy	Métrica	Evaluación del modelo
	Precision	Métrica	Evaluación del modelo
	Recall	Métrica	Evaluación del modelo
	F1-Score	Métrica	Evaluación del modelo
	Matriz de Confusión	Métrica	Evaluación del modelo

2.7 Brechas de investigación

La revisión de la literatura facilitó identificar importantes avances en la aplicación de técnicas de Procesamiento de Lenguaje Natural (PLN) y Machine Learning para la automatización de procesos relacionados con sistemas de soporte técnico, atención al cliente y gestión documental. Diversos estudios han demostrado que los algoritmos de clasificación supervisada pueden optimizar la eficiencia en la categorización de información textual y disminuir los tiempos de procesamiento en entornos organizacionales.

Sin embargo, también se evidencian diversas brechas que justifican el desarrollo de esta investigación. En primera instancia, la mayor parte de los trabajos revisados se desarrollan en empresas tecnológicas internacionales, plataformas de servicios en la nube o

grandes organizaciones, utilizando conjuntos de datos públicos o corporativos que presentan características diferentes a las observadas en empresas latinoamericanas dedicadas al desarrollo de software financiero.

En segundo lugar, gran parte de la literatura se concentra en problemas de asignación automática de tickets, recomendación de respuestas o escalamiento inteligente de incidentes, mientras que existe menor evidencia sobre investigaciones enfocadas específicamente en la clasificación inicial de tickets de soporte técnico utilizando categorías operativas propias de organizaciones de desarrollo de software.

Asimismo, se identificó que no existe consenso respecto al algoritmo de clasificación con mejor desempeño para este tipo de problemas. La eficacia de los modelos se ve influenciada por diversos factores, tales como la calidad del conjunto de datos y las metodologías empleadas en el preprocesamiento, la representación textual utilizada y las características particulares del dominio de aplicación. Esta situación justifica la comparación de diferentes algoritmos supervisados dentro de un mismo contexto organizacional.

Desde el punto de vista metodológico, los estudios revisados coinciden en la aplicación de técnicas de Procesamiento de Lenguaje Natural y modelos de Machine Learning; sin embargo, pocos describen de manera detallada el tratamiento de tickets mal categorizados, el manejo de clases desbalanceadas y la validación de modelos utilizando datos históricos reales provenientes de mesas de ayuda empresariales.

Finalmente, no se identificaron investigaciones publicadas que analicen la clasificación automática de tickets utilizando información histórica proveniente de Webcoopec System, empresa ecuatoriana especializada en el desarrollo de software para entidades del sector financiero popular y solidario. En consecuencia, la presente investigación busca aportar evidencia empírica sobre la aplicación de técnicas de PLN y Machine Learning

en un contexto empresarial real, contribuyendo tanto al ámbito académico como a la optimización de los procesos operativos de la entidad.

Un aspecto adicional identificado durante el análisis preliminar de la información corresponde a la calidad de las etiquetas históricas utilizadas para clasificar los tickets. El conjunto de datos tiene una cantidad significativa de registros asociados a la categoría **"No definido"**, debido a que durante los primeros meses de operación del sistema no existía un proceso formal de categorización. Asimismo, se identificó la existencia de categorías obsoletas o con una cantidad mínima de registros, como **"Otros"**, las cuales no resultan adecuadas para el entrenamiento de modelos supervisados. Esta situación hace necesario desarrollar una estrategia de depuración y selección de datos durante la fase de preparación propuesta por la metodología CRISP-DM.

Tabla 7

Brechas de literatura

Aspecto analizado	**Situación identificada en la literatura**	**Aporte de la presente investigación**
Contexto empresarial	Predominan estudios realizados en organnizaciones internacionales.	Implementación en una empresa ecuatoriana del sector financiero popular y solidario.
Problema abordado	Mayor énfasis en asignación o escalamiento de tickets.	Clasificación automática inicial de tickets de soporte técnico.
Datos utilizados	Principalmente datasets públicos o corporativos internacionales.	Tickets históricos reales de Webcoopec System.
Modelos evaluados	Resultados variables según el contexto de aplicación.	Comparación de Naive Bayes, SVM y Random Forest bajo un mismo conjunto de datos.
Aplicabilidad	Escasa evidencia en organizaciones latinoamericanas.	Generación de evidencia aplicada para empresas de desarrollo de software en Ecuador.

2.8 Síntesis del marco teórico

La elaboración de este marco teórico permitió establecer el sustento conceptual, metodológico y normativo de la investigación, evidenciando que la clasificación automática de tickets de soporte técnico constituye un problema ampliamente estudiado dentro del campo de la Ciencia de Datos y la Inteligencia Artificial. La revisión de literatura confirmó que las técnicas de Procesamiento de Lenguaje Natural y Machine Learning representan alternativas viables para automatizar procesos basados en información textual no estructurada, mejorando la eficiencia operativa y apoyando la toma de decisiones en organizaciones que gestionan grandes volúmenes de información.

Los antecedentes analizados muestran que diversos autores han aplicado modelos de clasificación supervisada para optimizar procesos relacionados con mesas de ayuda, atención al cliente y sistemas de soporte técnico. Asimismo, la literatura especializada respalda el uso de metodologías como CRISP-DM, técnicas de representación textual como TF-IDF y algoritmos como Naive Bayes, Support Vector Machine (SVM) y Random Forest, los cuales constituyen la base metodológica de la presente investigación.

Desde la perspectiva conceptual, el capítulo permitió integrar conocimientos relacionados con la gestión de servicios de soporte técnico, los sistemas de mesa de ayuda, el Procesamiento de Lenguaje Natural, la minería de texto, el aprendizaje supervisado y las métricas de evaluación utilizadas para comparar modelos de clasificación. De igual manera, se incorporó el contexto organizacional y normativo de Webcoopec System, considerando su actividad como proveedor de soluciones tecnológicas para organizaciones del ámbito financiero popular y solidario y la importancia de asegurar el manejo correcto de los datos empleados en la investigación.

Finalmente, la identificación de las brechas existentes en la literatura evidencia la oportunidad de desarrollar una propuesta aplicada que contribuya tanto al conocimiento

académico como a la solución de una problemática organizacional concreta. En consecuencia, el marco teórico proporciona el fundamento necesario para el desarrollo del capítulo metodológico, donde se describirá el proceso de extracción, preparación, modelado y evaluación de los datos mediante la metodología CRISP-DM, con el propósito de construir un modelo de clasificación automática de tickets que contribuya a optimizar la gestión operativa de la mesa de ayuda de Webcoopec System.

CAPÍTULO III
MARCO METODOLÓGICO

3.1 Enfoque de la investigación

Este estudio utiliza un enfoque cuantitativo, ya que se basa en la recolección, procesamiento y análisis de datos históricos de la mesa de ayuda de Webcoopec System. La información será obtenida desde la base de datos del sistema de tickets y posteriormente transformada en variables analizables mediante técnicas de Procesamiento de Lenguaje Natural, minería de texto y Machine Learning.

El enfoque cuantitativo facilita medir objetivamente el comportamiento de los datos y detectar patrones frecuentes, analizar la distribución de categorías y evaluar el desempeño de los modelos de clasificación automática mediante métricas numéricas como accuracy, precision, recall, F1-score y matriz de confusión.

Asimismo, este enfoque es adecuado porque la investigación no se basa en percepciones subjetivas, entrevistas o encuestas, sino en registros históricos generados por la operación real de la mesa de ayuda. Por tanto, los resultados podrán ser comparados, medidos y validados estadísticamente.

3.2 Tipo de investigación

La investigación es de tipo aplicada, descriptiva, correlacional y exploratoria.

Es aplicada porque busca resolver una problemática concreta dentro de Webcoopec System: la clasificación manual de tickets de soporte técnico. El propósito no es únicamente generar conocimiento teórico, sino desarrollar una solución en base a la Ciencia de Datos que pueda aportar a la optimización operativa de la mesa de ayuda.

Es descriptiva porque permitirá caracterizar el comportamiento histórico de los tickets registrados, identificando categorías más frecuentes, volumen de solicitudes, palabras recurrentes, distribución temporal, estados, prioridades y patrones operativos dentro del sistema de soporte.

Es correlacional porque analizará la relación existente entre las características textuales de los tickets, como asunto, descripción, palabras clave y frecuencia de términos, y la categoría asignada históricamente por el personal de soporte.

Finalmente, posee un componente exploratorio, debido a que se analizará el comportamiento de los datos históricos antes del desarrollo del modelo, con el fin de identificar la calidad del dataset, posibles valores faltantes, clases desbalanceadas, tickets repetitivos, inconsistencias de categorización y potenciales áreas de mejora en el proceso de clasificación.

3.3 Diseño de la investigación

El diseño de la investigación es no experimental, transversal y analítico.

No es experimental ya que no se controlarán variables independientes ni se intervendrá directamente en el proceso operativo de atención de tickets. La investigación utilizará datos históricos ya existentes en la base de datos de la mesa de ayuda de Webcoopec System, por lo que el análisis se realizará sobre información previamente registrada.

Es transversal debido a que los datos se obtendrán en un período definido de tiempo, correspondiente a los tickets registrados dentro del rango definido para el estudio. Esto permitirá analizar el comportamiento de la mesa de ayuda durante un período determinado, sin realizar seguimiento longitudinal de los mismos casos en diferentes momentos.

Es analítico porque, además de describir los datos, se aplicarán técnicas de procesamiento, transformación, modelado y evaluación para identificar patrones y construir modelos de clasificación automática. En este sentido, el uso de Machine Learning no constituye por sí mismo el diseño de investigación, sino una técnica analítica utilizada dentro del diseño metodológico.

Adicionalmente, para el desarrollo técnico de la solución propuesta se empleará la metodología CRISP-DM (Cross Industry Standard Process for Data Mining). Se utiliza

ampliamente en proyectos de Ciencia de Datos al estructurar el proceso de desarrollo en seis etapas: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue (Chapman et al., 2000).

3.4 Contexto del estudio

La investigación se llevará a cabo en la empresa Webcoopec System Cía. Ltda., organización ecuatoriana dedicada al desarrollo de software especializado para cooperativas de ahorro y crédito, cajas de ahorro y otras entidades del sector financiero popular y solidario. Entre los servicios que presta la empresa se encuentran el desarrollo de software, soporte técnico, mantenimiento de aplicaciones, infraestructura tecnológica, Analítica de Datos y soluciones informáticas para instituciones reguladas por la Superintendencia de Economía Popular y Solidaria (SEPS).

Como parte de la gestión del soporte técnico, la empresa utiliza la plataforma osTicket, un sistema de mesa de ayuda (Help Desk) de código abierto que permite registrar, administrar y dar seguimiento a los requerimientos reportados por los clientes (osTicket, s. f.). Cada ticket almacena información relacionada con el asunto del requerimiento, descripción del problema, área responsable, prioridad, estado, fecha de creación, técnico asignado y demás atributos necesarios para la gestión del soporte.

Actualmente, el personal de soporte categoriza los tickets manualmente, lo que genera diferencias en los criterios de clasificación entre técnicos, inconsistencias en las categorías asignadas y mayores tiempos durante el proceso de atención. Esta situación dificulta el análisis histórico de los requerimientos, afecta la generación de indicadores y limita la automatización de procesos relacionados con la mesa de ayuda.

En este marco, la presente investigación sugiere la creación de un modelo de clasificación automatizada fundamentado en metodologías de Procesamiento de Lenguaje Natural (PLN) y Machine Learning, utilizando información histórica proveniente de la base

de datos de osTicket, con el propósito de apoyar el proceso de categorización y ayudar a optimizar la eficacia operativa del manejo de soporte técnico.

3.5 Población

La población objeto de análisis estará constituida por aproximadamente 17,000 tickets históricos registrados en la plataforma Webcoopec Ticket desde la implementación del sistema en diciembre de 2023 hasta la fecha definida para la extracción de los datos.

La población incluye tickets correspondientes a diferentes módulos del sistema, incidencias técnicas, consultas funcionales, requerimientos de parametrización, errores operativos y solicitudes de asistencia registradas en la plataforma osTicket. Estos registros representan la fuente principal de información para el desarrollo del modelo de clasificación automática.

3.6 Muestra

Este estudio empleará una muestra censal, porque se usará la totalidad de los registros históricos de la base de datos del sistema Webcoopec Ticket que cumplan los criterios establecidos de calidad para el modelo de clasificación automática. Esta decisión permite aprovechar al máximo la información generada por la organización y obtener un modelo entrenado sobre datos representativos del comportamiento real de la mesa de ayuda.

La base de datos histórica contiene aproximadamente 17.000 tickets registrados desde la implementación de la plataforma en diciembre de 2023 hasta la fecha de extracción de la información. No obstante, previo al entrenamiento del modelo se llevará a cabo un proceso de depuración y organización de los datos, con el fin de asegurar la calidad de la información empleada en el aprendizaje supervisado.

Durante la fase de preparación de los datos de la metodología CRISP-DM se aplicarán criterios de inclusión y exclusión, eliminando registros incompletos, duplicados o que no proporcionen datos significativos para el entrenamiento del modelo. Asimismo, se excluirán

aquellas categorías que no representan procesos operativos vigentes dentro de la organización, con el fin de construir un modelo consistente y alineado con la realidad actual de Webcoopec System.

Los criterios considerados son los siguientes:

Tabla 8

Criterios de inclusión y exclusión de tickets

Criterios de inclusión	Criterios de exclusión
Tickets con categoría funcional asignada.	Tickets pertenecientes a la categoría No definido.
Tickets con asunto y descripción registrados.	Tickets pertenecientes a la categoría Otros.
Tickets correspondientes a clientes reales de la organización.	Tickets utilizados para pruebas, capacitación o validaciones internas.
Tickets cerrados o resueltos que reflejen la clasificación definitiva realizada por el personal de soporte.	Tickets duplicados o con información inconsistente.
Tickets generados dentro del período definido para el estudio.	Tickets con información insuficiente para el procesamiento del lenguaje natural.

Tras la implementación del proceso de limpieza de datos, el conjunto de datos se segmentará en subconjuntos destinados al entrenamiento y a la prueba, permitiendo entrenar los diferentes algoritmos de Machine Learning y evaluar objetivamente su capacidad de clasificación sobre registros no utilizados durante el aprendizaje.

3.7 Unidad de análisis

La unidad de análisis equivale a cada ticket registrado en la plataforma osTicket de Webcoopec System. Cada ticket representa una observación individual dentro del estudio y contiene información estructurada y no estructurada relacionada con un requerimiento de soporte técnico realizado por un cliente.

Cada registro incorpora atributos como el asunto del ticket, descripción del problema, fecha de creación, estado, prioridad, departamento asignado, técnico responsable y categoría de clasificación, entre otros datos utilizados durante el proceso de atención. Estas características constituyen la base para el análisis estadístico descriptivo y el entrenamiento del modelo de clasificación automática.

Considerar cada ticket como unidad de análisis permite evaluar de manera individual las características textuales y operativas de los requerimientos, permitiendo reconocer patrones de comportamiento y el desarrollo de modelos predictivos mediante técnicas de Procesamiento de Lenguaje Natural y Machine Learning.

3.8 Fuente de información

La fuente primaria de información será la base de datos transaccional del sistema de mesa de ayuda osTicket utilizado por Webcoopec System para la gestión de requerimientos de soporte técnico. Esta base de datos almacena el historial de tickets generados por los clientes durante la prestación del servicio, constituyendo una fuente primaria de información para la investigación.

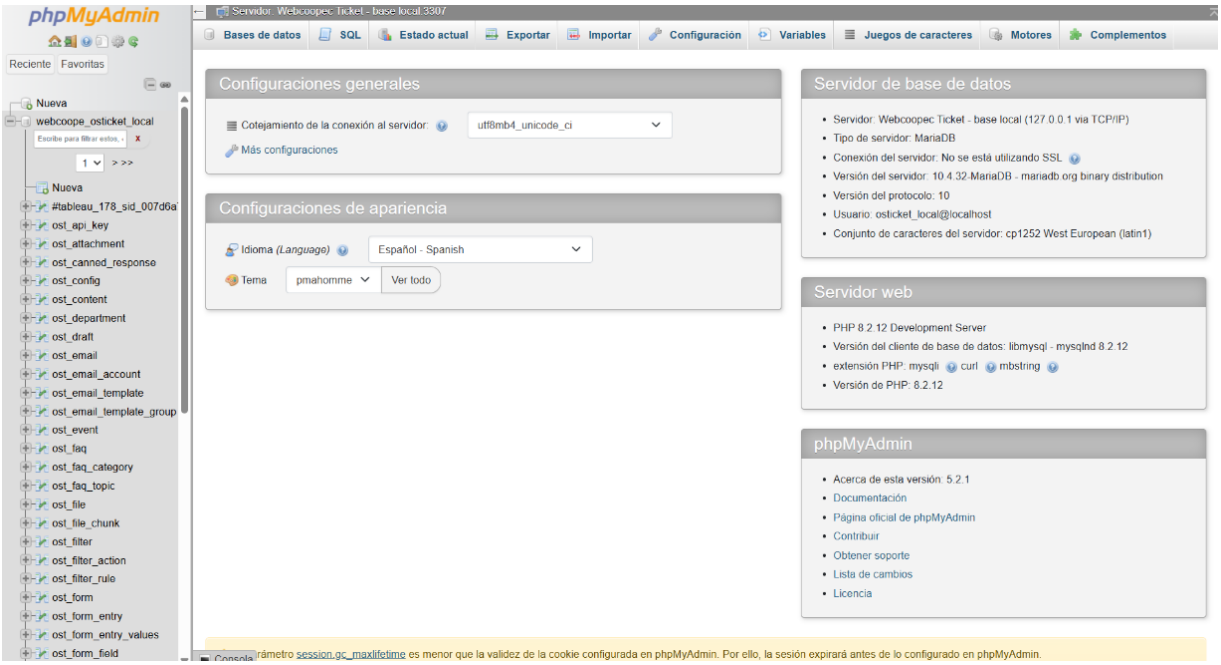
El acceso a la información será realizado únicamente después de obtener la autorización institucional correspondiente, mediante la aprobación del Comité Técnico de la empresa y la validación del Administrador de Base de Datos (DBA), garantizando que la extracción de los registros cumpla los procedimientos internos de seguridad de la información.

Los datos se utilizarán únicamente con fines académicos y de investigación. Previamente al proceso de análisis se aplicarán mecanismos de anonimización para eliminar información que permita identificar directamente a personas naturales o jurídicas, preservando únicamente los atributos necesarios para el desarrollo del modelo de clasificación automática.

La Figura 1 muestra la fuente de datos utilizada para la extracción y análisis de los registros históricos de Webcoopec Ticket.

Figura 1

Fuente de datos - Webcoopec Ticket



Nota. Captura de pantalla de la base de datos local de Webcoopec Ticket visualizada mediante phpMyAdmin.

3.9 Desarrollo de la metodología CRISP-DM

3.9.1 Comprensión del Negocio

La metodología CRISP-DM (Cross Industry Standard Process for Data Mining) establece como primera etapa la comprensión del negocio, que tiene como objetivo examinar el entorno organizacional, identificar la problemática existente, comprender las necesidades de la organización y traducir los objetivos del negocio en objetivos técnicos que puedan resolverse mediante técnicas de Ciencia de Datos y Machine Learning.

En la presente investigación, esta fase se desarrolla en la empresa Webcoopec System Cía. Ltda., organización ecuatoriana dedicada al desarrollo de soluciones informáticas para cooperativas de ahorro y crédito, cajas de ahorro y demás entidades pertenecientes al Sector Financiero Popular y Solidario. La empresa mantiene una mesa de ayuda denominada

Webcoopec Ticket, implementada sobre la plataforma de código abierto osTicket, la cual constituye el canal oficial para la recepción, seguimiento y gestión de los requerimientos realizados por sus clientes.

Como proveedor tecnológico, Webcoopec System fue calificada por la Superintendencia de Economía Popular y Solidaria (SEPS, 2025) como empresa prestadora de servicios auxiliares mediante la Resolución Nro. SEPS-INSEPS-IGS-DNGRT-2025-0214, requisito que implica el cumplimiento de condiciones técnicas y operativos relacionados con la disponibilidad de los servicios, continuidad tecnológica y atención oportuna de las instituciones financieras que utilizan sus soluciones.

Durante la operación diaria de la mesa de ayuda se registran solicitudes relacionadas con consultas funcionales, incidencias operativas, errores del software, requerimientos comerciales y demás servicios de soporte técnico. El incremento progresivo de la cantidad de clientes y del volumen de tickets ha generado una creciente dependencia del proceso manual de clasificación, el cual actualmente es realizado por los analistas de soporte utilizando su experiencia y criterio para determinar la categoría correspondiente a cada requerimiento.

La clasificación manual presenta limitaciones relacionadas con la subjetividad del proceso, diferencias de criterio entre analistas, incremento de los tiempos de atención, dificultad para identificar patrones históricos y problemas en la generación de indicadores de gestión. Adicionalmente, una categorización incorrecta puede afectar procesos administrativos internos, especialmente aquellos relacionados con la identificación de tickets que deben ser facturados bajo la modalidad de soporte hora-hombre y aquellos que forman parte de los contratos de mantenimiento vigentes con los clientes.

Frente a esta problemática, la presente investigación propone desarrollar un modelo de clasificación automática basado en técnicas de Procesamiento de Lenguaje Natural (PLN) y Machine Learning Supervisado, empleando los datos históricos guardados en la base de

datos del sistema Webcoopec Ticket. El modelo permitirá sugerir automáticamente la categoría más probable para cada nuevo ticket, contribuyendo a estandarizar el proceso de clasificación, optimizar la administración operativa del servicio de asistencia y convertir los datos históricos en conocimiento aplicable para la toma de decisiones.

El éxito del proyecto no estará determinado únicamente por la capacidad del modelo para clasificar adecuadamente los tickets, sino también por su aporte al mejoramiento de los procesos operativos de la organización. En este sentido, se espera que la solución contribuya a disminuir los tiempos de clasificación, reducir errores de categorización, mejorar la consistencia de los datos históricos y facilitar la generación de indicadores estratégicos mediante herramientas de Analítica de Datos como Tableau.

La Figura 2 presenta el logotipo institucional de Webcoopec System Cía. Ltda.

Figura 2

Logotipo de Webcoopec System Cía. Ltda

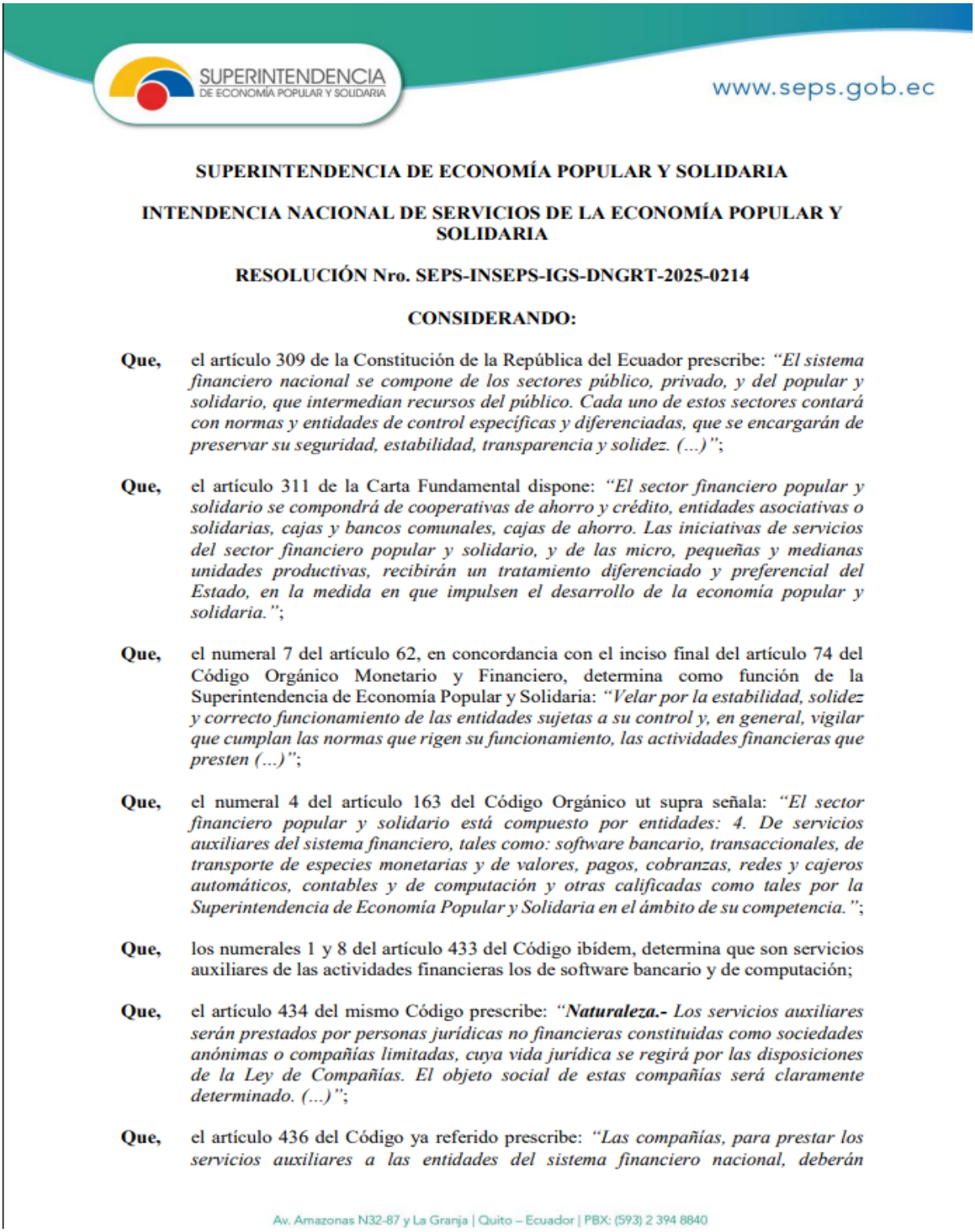


Nota. Logotipo corporativo de Webcoopec System Cía. Ltda.

La Figura 3 presenta la resolución mediante la cual la Superintendencia de Economía Popular y Solidaria calificó a Webcoopec System como empresa prestadora de servicios auxiliares.

Figura 3

Resolución de calificación de Webcoopec System como empresa de servicios auxiliares



SUPERINTENDENCIA DE ECONOMÍA POPULAR Y SOLIDARIA

www.seps.gob.ec

SUPERINTENDENCIA DE ECONOMÍA POPULAR Y SOLIDARIA

INTENDENCIA NACIONAL DE SERVICIOS DE LA ECONOMÍA POPULAR Y SOLIDARIA

RESOLUCIÓN Nro. SEPS-INSEPS-IGS-DNGRT-2025-0214

CONSIDERANDO:

Que, el artículo 309 de la Constitución de la República del Ecuador prescribe: *"El sistema financiero nacional se compone de los sectores público, privado, y del popular y solidario, que intermedian recursos del público. Cada uno de estos sectores contará con normas y entidades de control específicas y diferenciadas, que se encargarán de preservar su seguridad, estabilidad, transparencia y solidez. (...)"*;

Que, el artículo 311 de la Carta Fundamental dispone: *"El sector financiero popular y solidario se compondrá de cooperativas de ahorro y crédito, entidades asociativas o solidarias, cajas y bancos comunales, cajas de ahorro. Las iniciativas de servicios del sector financiero popular y solidario, y de las micro, pequeñas y medianas unidades productivas, recibirán un tratamiento diferenciado y preferencial del Estado, en la medida en que impulsen el desarrollo de la economía popular y solidaria."*;

Que, el numeral 7 del artículo 62, en concordancia con el inciso final del artículo 74 del Código Orgánico Monetario y Financiero, determina como función de la Superintendencia de Economía Popular y Solidaria: *"Velar por la estabilidad, solidez y correcto funcionamiento de las entidades sujetas a su control y, en general, vigilar que cumplan las normas que rigen su funcionamiento, las actividades financieras que presten (...)"*;

Que, el numeral 4 del artículo 163 del Código Orgánico ut supra señala: *"El sector financiero popular y solidario está compuesto por entidades: 4. De servicios auxiliares del sistema financiero, tales como: software bancario, transaccionales, de transporte de especies monetarias y de valores, pagos, cobranzas, redes y cajeros automáticos, contables y de computación y otras calificadas como tales por la Superintendencia de Economía Popular y Solidaria en el ámbito de su competencia."*;

Que, los numerales 1 y 8 del artículo 433 del Código ibídem, determina que son servicios auxiliares de las actividades financieras los de software bancario y de computación;

Que, el artículo 434 del mismo Código prescribe: *"**Naturaleza.-** Los servicios auxiliares serán prestados por personas jurídicas no financieras constituidas como sociedades anónimas o compañías limitadas, cuya vida jurídica se regirá por las disposiciones de la Ley de Compañías. El objeto social de estas compañías será claramente determinado. (...)"*;

Que, el artículo 436 del Código ya referido prescribe: *"Las compañías, para prestar los servicios auxiliares a las entidades del sistema financiero nacional, deberán*

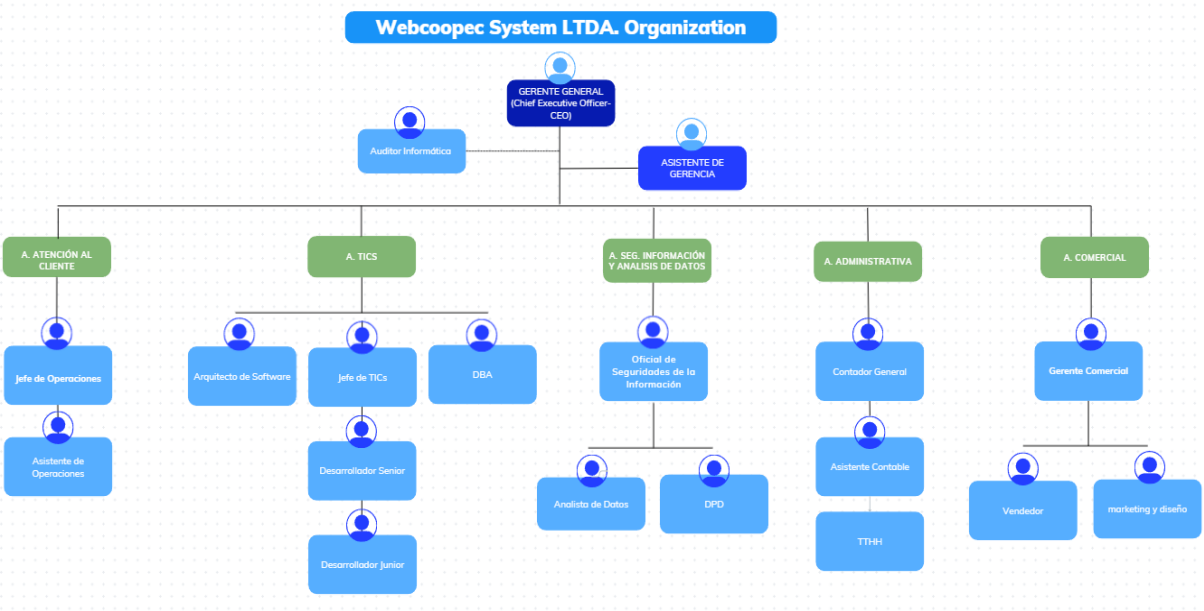
Av. Amazonas N32-87 y La Granja | Quito – Ecuador | PBX: (593) 2 394 8840

Nota. Resolución Nro. SEPS-INSEPS-IGS-DNGRT-2025-0214, emitida por la Superintendencia de Economía Popular y Solidaria (2025).

La Figura 4 presenta la estructura organizacional de Webcoopec System Cía. Ltda.

Figura 4

Estructura organizacional de Webcoopec System Cía. Ltda.



Nota. Organigrama institucional de Webcoopec System Cía. Ltda.

La Figura 5 presenta el proceso actual de gestión de tickets de soporte técnico en Webcoopec System.

Figura 5

Proceso actual de gestión de tickets



Actualmente, una parte significativa del ciclo de vida de un ticket se destina al proceso de clasificación manual. Este proceso, en muchos casos, requiere la intervención de comités técnicos, la realización de consultas, investigaciones y revisiones por parte del personal especializado en la temática del incidente. Como consecuencia, se incrementa el tiempo necesario para la asignación y atención del ticket, lo que repercute de manera adversa en los tiempos de respuesta y en la eficacia del servicio de soporte.

La Figura 6 presenta el proceso propuesto para la gestión de tickets con la incorporación del modelo de Machine Learning.

Figura 6

Proceso propuesto con incorporación del modelo de Machine Learning



Con la implementación del modelo de *Machine Learning* propuesto, se espera disminuir de manera considerable los plazos de respuesta en la gestión de solicitudes de atención. La clasificación automática permitirá identificar de manera rápida y precisa el tipo de soporte requerido, así como sugerir el técnico más adecuado para su atención, eliminando

en gran medida la necesidad de realizar análisis manuales o convocar comités técnicos para cada caso.

Además, el modelo aprovechará la base histórica de conocimiento generada a partir de los tickets previamente categorizados, lo que permitirá mejorar continuamente la precisión de las predicciones y optimizar el proceso de asignación. Como resultado, se incrementará la eficiencia operativa de la mesa de ayuda,se acortarán los tiempos de respuesta y se optimizará la calidad del servicio a los usuarios.

Tabla 9

Objetivos del negocio y objetivos técnicos

Objetivo del negocio	Objetivo técnico del proyecto
Reducir los tiempos de clasificación de tickets.	Desarrollar un modelo de Machine Learning que clasifique automáticamente nuevos tickets.
Disminuir errores de categorización.	Comparar diferentes algoritmos supervisados y seleccionar el de mayor desempeño.
Estandarizar la clasificación realizada por los analistas.	Utilizar categorías históricas como variable objetivo del entrenamiento.
Detectar incidencias recurrentes.	Aplicar técnicas de Procesamiento de Lenguaje Natural para identificar patrones en los textos.
Mejorar la gestión operativa del soporte.	Generar indicadores descriptivos sobre categorías, clientes, agentes y comportamiento histórico.
Identificar correctamente los tickets cobrables.	Diferenciar la categoría Operativo para cobrar respecto del resto de categorías operativas.
Facilitar la toma de decisiones.	Construir dashboards analíticos en Tableau utilizando la información clasificada por el modelo.

3.9.2 Comprensión de los datos

La segunda etapa de la metodología CRISP-DM se refiere a la comprensión de los datos. Su objetivo es discernir las fuentes de información accesibles, conocer su estructura, analizar las relaciones existentes entre las entidades y efectuar una evaluación inicial de su

calidad antes de aplicar procesos de limpieza, transformación o modelado (Chapman et al., 2000).

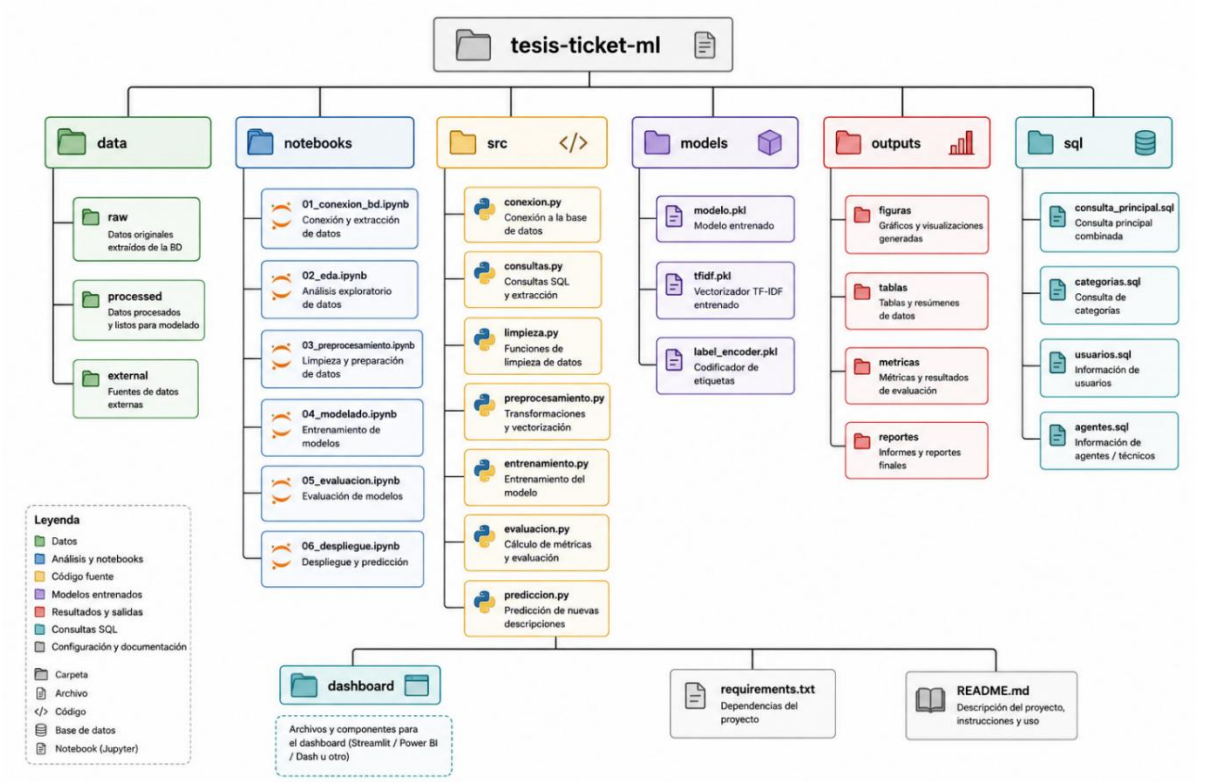
Para el desarrollo de esta fase se utilizó un respaldo de la base de datos transaccional del sistema Webcoopec Ticket. Dicho respaldo fue restaurado en un entorno local aislado, implementado mediante XAMPP y MariaDB. De esta manera, las consultas y los procedimientos de exploración se ejecutaron exclusivamente sobre una copia local, sin establecer conexiones ni realizar modificaciones sobre la base de datos de producción.

La vinculación y el análisis de los datos se llevaron a cabo utilizando la programación en Python, utilizando un entorno virtual independiente y una estructura organizada de proyecto. Esta estructura contiene carpetas específicas para los datos, notebooks, código fuente, consultas SQL, modelos, figuras, tablas, métricas, reportes y componentes de despliegue. La separación de estos elementos facilita la reproducibilidad del análisis y permite mantener un orden coherente con las fases posteriores de CRISP-DM.

La Figura 7 presenta la estructura utilizada para organizar los componentes del proyecto de ciencia de datos desarrollado para la clasificación automática de tickets.

Figura 7

Estructura del proyecto de ciencia de datos para la clasificación de tickets



La estructura del proyecto fue denominada tesis-ticket-ml. En la carpeta notebooks se organizaron los análisis de conexión, exploración, preprocesamiento, modelado, evaluación y despliegue. La carpeta src contiene los módulos reutilizables de conexión, consultas, limpieza, preprocesamiento, entrenamiento, evaluación y predicción. Por otra parte, las consultas empleadas para examinar la base de datos se almacenaron separadamente en la carpeta sql.

3.9.2.1 Conexión con la fuente de datos. La fuente principal de información corresponde a la base de datos local webcoope_osticket_local, restaurada a partir del respaldo de Webcoopec Ticket. La conexión se realizó mediante Python y la biblioteca PyMySQL, utilizando parámetros almacenados en un archivo de configuración separado del código fuente.

Como medida de seguridad, las credenciales no fueron incorporadas directamente en los notebooks ni en los archivos de código. Además, la sesión analítica fue configurada en modo de solo lectura, con el objetivo de impedir modificaciones accidentales durante la inspección de la información.

El siguiente fragmento presenta el procedimiento utilizado para establecer la conexión con la copia local de MariaDB:

```
import os
from contextlib import contextmanager
from pathlib import Path

import pymysql

ROOT = Path(__file__).resolve().parents[1]


def cargar_env(ruta=ROOT / ".env"):
    if not ruta.exists():
        raise FileNotFoundError(
            "Copie .env.example como .env y complete la configuración local."
        )

    for linea in ruta.read_text(encoding="utf-8").splitlines():
        linea = linea.strip()

        if linea and not linea.startswith("#") and "=" in linea:
```

```
            clave, valor = linea.split("=", 1)
            os.environ.setdefault(clave.strip(), valor.strip())


@contextmanager
def conexion_solo_lectura():
    cargar_env()

    conexion = pymysql.connect(
        host=os.getenv("DB_HOST", "127.0.0.1"),
        port=int(os.getenv("DB_PORT", "3307")),
        user=os.environ["DB_USER"],
        password=os.getenv("DB_PASSWORD", ""),
        database=os.environ["DB_NAME"],
        charset="utf8mb4",
        cursorclass=pymysql.cursors.DictCursor,
        autocommit=False
    )

    try:
        with conexion.cursor() as cursor:
            cursor.execute("SET SESSION TRANSACTION READ ONLY")
            cursor.execute("START TRANSACTION READ ONLY")

        yield conexion
        conexion.rollback()

    finally:
        conexion.close()
```

La ejecución del código permitió verificar correctamente la comunicación con MariaDB versión 10.4.32. Posteriormente, se realizaron consultas de metadatos y conteos

agregados que no incluyeron nombres, direcciones de correo electrónico, números telefónicos, direcciones IP ni contenidos completos de los tickets.

La Figura 8 muestra la verificación de la conexión entre Python y la copia local de la base de datos MariaDB utilizada en la investigación.

Figura 8

Verificación de la conexión entre Python y la copia local de MariaDB



3.9.2.2 Identificación de tablas relevantes. La inspección automatizada permitió identificar 73 tablas dentro de la base de datos. El sistema utiliza el prefijo ost_, correspondiente a la estructura de osTicket. No todas las tablas encontradas serán utilizadas para construir el conjunto de datos del modelo, debido a que varias contienen configuraciones, sesiones, archivos adjuntos, plantillas, registros del sistema o información generada después de la creación de un ticket.

Las principales tablas identificadas para la investigación se presentan a continuación:

Tabla 10

Tablas identificadas en la base de datos de Webcoopec Ticket

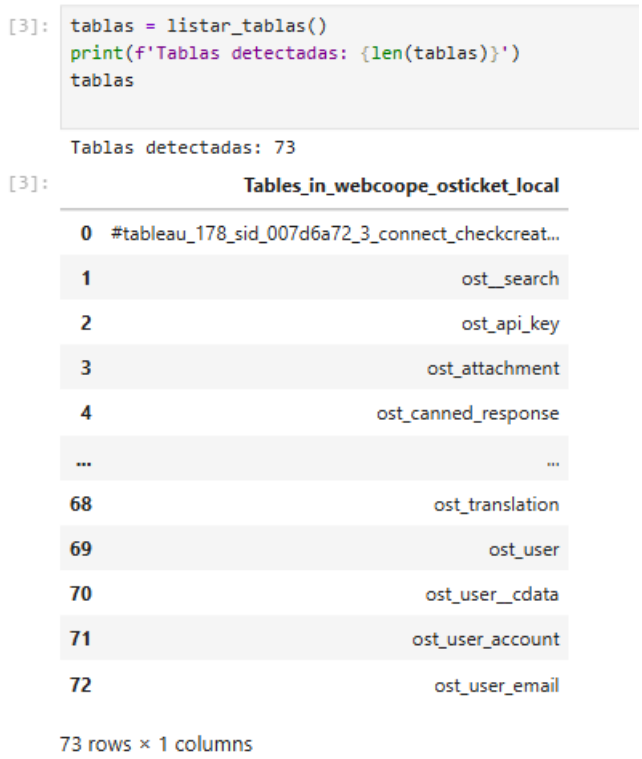
Tabla	Descripción	Utilidad en la investigación
ost_ticket	Contiene la información general de los tickets.	Constituye la entidad central del análisis.
ost_ticket__cdata	Almacena campos dinámicos asociados al ticket, incluido el asunto.	Permite recuperar el título o asunto del requerimiento.
ost_help_topic	Contiene los temas de ayuda o categorías.	Proporciona la descripción de la variable objetivo.
ost_user	Registra a los usuarios solicitantes.	Permite conocer la cantidad de usuarios y relacionarlos con los tickets.
ost_organization	Contiene las organizaciones vinculadas con los usuarios.	Representa a los clientes institucionales.
ost_staff	Registra a los agentes o técnicos.	Se utiliza para la caracterización descriptiva del sistema.
ost_thread	Relaciona cada ticket con su hilo de comunicación.	Permite localizar la conversación asociada al ticket.
ost_thread_entry	Contiene mensajes, respuestas y notas del hilo.	Permite identificar el mensaje inicial del requerimiento.

Nota. Información obtenida de la estructura de la base de datos de Webcoopec Ticket.

La Figura 9 muestra la identificación de las tablas disponibles en la copia local de la base de datos utilizada para el análisis.

Figura 9

Identificación de las tablas disponibles en la base de datos local



```
[3]: tablas = listar_tablas()
     print(f'Tablas detectadas: {len(tablas)}')
     tablas
```

```
Tablas detectadas: 73
```

[3]:

	Tables_in_webcoope_osticket_local
0	#tableau_178_sid_007d6a72_3_connect_checkcreat...
1	ost__search
2	ost_api_key
3	ost_attachment
4	ost_canned_response
...	...
68	ost_translation
69	ost_user
70	ost_user__cdata
71	ost_user_account
72	ost_user_email

73 rows × 1 columns

Nota. Resultado generado durante la exploración de la copia local de la base de datos de Webcoopec Ticket.

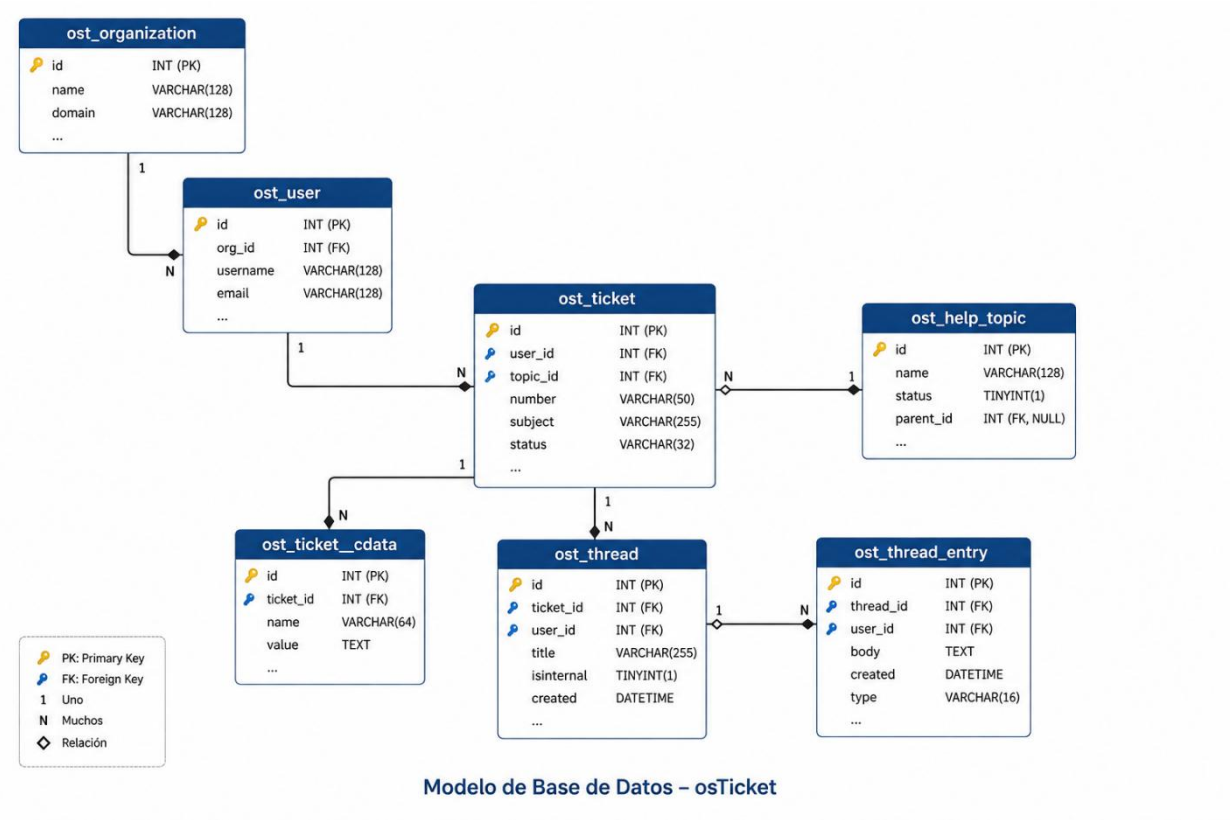
La tabla central es ost_ticket. Su campo user_id permite relacionar cada ticket con el usuario registrado en ost_user, mientras que topic_id lo relaciona con su categoría histórica en ost_help_topic. El asunto se encuentra en ost_ticket__cdata, utilizando ticket_id como campo de correspondencia.

Por otra parte, el contenido textual se organiza mediante ost_thread y ost_thread_entry. La primera relaciona el identificador del ticket con un hilo de conversación, mientras que la segunda almacena las entradas correspondientes a mensajes, respuestas y notas.

La Figura 10 presenta la estructura simplificada de las relaciones entre las principales tablas utilizadas en la investigación:

Figura 10

Relaciones principales entre las tablas utilizadas en la investigación



La evaluación de integridad referencial no encontró registros sin coincidencia en las cuatro relaciones principales analizadas: tickets con usuarios, tickets con categorías válidas, tickets con hilos y tickets con sus campos dinámicos. Los 15 registros cuyo topic_id es igual a cero fueron reconocidos como tickets sin categoría y no como referencias inválidas.

3.9.2.3 Descripción inicial de los datos. El inventario inicial permitió obtener los conteos de las entidades más importantes del sistema.

Tabla 11

Conteos iniciales de las entidades de Webcoopec Ticket.

Entidad	Cantidad
Tickets	16.993
Usuarios solicitantes	99
Agentes	30

Entidad	Cantidad
Clientes u organizaciones	66
Categorías o temas de ayuda	8

Nota. Elaborado a partir del respaldo local de Webcoopec Ticket (2026).

La Figura 11 presenta los conteos iniciales de las principales entidades identificadas en la base de datos de Webcoopec Ticket.

Figura 11

Conteos iniciales de las entidades de Webcoopec Ticket

```
[4]: conteos = conteos_entidades()
     conteos
```

[4]:

	entidad	total
0	Tickets	16993
1	Usuarios solicitantes	99
2	Agentes	30
3	Clientes	66
4	Categorias	8

Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Los conteos anteriores representan una descripción general del respaldo y no constituyen todavía el conjunto definitivo de entrenamiento. La cantidad final de observaciones se basará en los criterios de inclusión y exclusión definidos en la fase de preparación de datos.

El periodo cubierto por el respaldo comprende desde el 30 de diciembre de 2023 hasta el 5 de agosto de 2026. Por tanto, la información contiene aproximadamente dos años y siete meses de registros históricos del sistema.

3.9.2.4 Identificación de la categoría de producto y de la variable objetivo.

Durante la comprensión de los datos se identificaron dos tipos de clasificación que cumplen

funciones diferentes dentro del sistema Webcoopec Ticket. La primera corresponde al producto relacionado con el requerimiento, mientras que la segunda representa el tipo de incidencia que se desea predecir mediante el modelo de aprendizaje automático.

La categoría almacenada en ost_ticket.topic_id identifica el producto al que pertenece cada ticket. Su descripción se obtiene mediante la relación con el campo ost_help_topic.topic. Entre los productos registrados se encuentran SIRYUSWEB, WEBCOOP, FYNNBOX, SIFINCAJ, RISKWEB e Analítica de Datos.

Esta información no constituye la variable objetivo del modelo, debido a que el usuario selecciona el producto al momento de registrar el ticket. Por consiguiente, la categoría de producto representa una variable descriptiva y una posible variable predictora, ya que se encuentra disponible antes de que el sistema genere la sugerencia del tipo de incidencia.

La distribución inicial de los tickets por producto se presenta en la Tabla 12.

Tabla 12

Distribución inicial de tickets por categoría

Categoría	Tickets	Porcentaje
Reportar Incidencia SIRYUSWEB	7.925	46,64 %
Reportar Incidencia WEBCOOP	7.505	44,17 %
Reportar Incidencia FYNNBOX	727	4,28 %
Reportar Incidencia SIFINCAJ	718	4,23 %
Reportar Incidencia RISKWEB	72	0,42 %
Solicitud Analítica de Datos	17	0,10 %
Sin categoría	15	0,09 %
Consulta general	14	0,08 %
Total	**16.993**	**100,00 %**

***Nota*.** Datos obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

La Figura 12 muestra el resultado de la consulta utilizada para obtener la distribución inicial de los tickets por categoría.

Figura 12

Distribución inicial de tickets por categoría obtenida mediante Python

```
[5]: categorias = distribucion_categorias()
     categorias
```

[5]:

	topic_id	categoria	tickets	porcentaje
0	12	Reportar Incidencia SIRYUSWEB	7925	46.64
1	11	Reportar Incidencia WEBCOOP	7505	44.17
2	15	Reportar Incidencia FYNNBOX	727	4.28
3	13	Reportar Incidencia SIFINCAJ	718	4.23
4	14	Reportar Incidencia RISKWEB	72	0.42
5	16	Solicitud Inteligencia de Negocios	17	0.10
6	0	SIN_CATEGORIA	15	0.09
7	1	Consulta general	14	0.08

Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

La variable que se pretende predecir se encuentra en el campo Incidencia de la tabla ost_ticket__cdata. Este campo está configurado en osTicket como una lista denominada "Tipo Incidencia" y almacena internamente códigos numéricos. Para obtener las denominaciones correspondientes, dichos códigos se relacionan con los elementos de la lista registrados en ost_list_items.

Las categorías de incidencia encontradas en la base de datos son:

Tabla 13

Categorías de incidencia

Código	Tipo de incidencia
1	Soporte Funcional
2	Soporte Operativo
3	Soporte Software
4	Otras Incidencias
5	Soporte Comercial (Cotizaciones)
6	Soporte Operativo para cobrar

Nota. Información obtenida de la configuración de Webcoopec Ticket (2026).

Por tanto, el campo Incidencia constituye la variable objetivo del modelo de clasificación. El sistema deberá aprender a sugerir una de estas categorías utilizando la información disponible cuando se crea el ticket, principalmente el asunto, la descripción inicial y el producto seleccionado por el usuario.

De los 16.993 tickets analizados, 5.866 poseen una incidencia registrada, mientras que 11.127 presentan el campo Incidencia nulo o vacío. En la interfaz del sistema, estos últimos pueden interpretarse como casos “no definidos”; sin embargo, en la estructura de la base de datos no existe una opción independiente con ese nombre dentro de la lista de tipos de incidencia.

La distribución de los tickets que poseen una incidencia válida se presenta a continuación:

Tabla 14

Distribución de la variable objetivo en los tickets etiquetados

Tipo de incidencia	Tickets	Porcentaje de tickets etiquetados
Soporte Funcional	2.561	43,66 %
Soporte Software	1.589	27,09 %
Soporte Operativo	1.488	25,37 %
Soporte Operativo para cobrar	140	2,39 %
Soporte Comercial (Cotizaciones)	85	1,45 %
Otras Incidencias	3	0,05 %
Total de tickets etiquetados	**5.866**	**100,00 %**

Nota. Datos obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Los hallazgos muestran dos problemas relevantes para el desarrollo del modelo. En primer lugar, el 65,48 % de los tickets no posee una incidencia registrada, por lo que estos casos no pueden utilizarse directamente como ejemplos etiquetados para un modelo de clasificación supervisada. En segundo lugar, las 5.866 observaciones etiquetadas presentan un desbalance considerable entre las clases.

La categoría Soporte Funcional representa el 43,66 % de los tickets etiquetados, mientras que Otras Incidencias contiene únicamente tres registros. Las categorías Soporte Operativo para cobrar y Soporte Comercial también poseen una representación reducida.

En la fase de preparación de los datos se deberá determinar el tratamiento de los tickets sin incidencia y establecer si las categorías con una cantidad insuficiente de ejemplos pueden conservarse, agruparse o requieren un proceso adicional de etiquetado manual. Esta decisión deberá ser validada con los responsables de Webcoopec antes del entrenamiento definitivo.

Debido al desbalance observado, la exactitud global no será utilizada como única medida de evaluación. Se analizarán la precisión, la exhaustividad y la medida F1 por categoría, además de los promedios macro y la matriz de confusión.

3.9.2.5 Distribución por canal de ingreso. El campo source de ost_ticket identifica el medio utilizado para registrar cada requerimiento. Su distribución inicial se muestra en la Tabla 15.

Tabla 15

Distribución de tickets por canal de ingreso

Canal	Tickets	Porcentaje
Web	14.332	84,34 %
Teléfono	2.455	14,45 %
Correo electrónico	194	1,14 %
Otros	12	0,07 %
Total	16.993	100,00 %

Nota. Datos obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

La mayor parte de los tickets fue registrada mediante la interfaz web. El canal de ingreso puede considerarse inicialmente como una variable descriptiva y, de forma

condicionada, como una predictora candidata, siempre que se confirme que su valor se encuentra disponible antes de generar la sugerencia de categoría.

3.9.2.6 Evaluación inicial de la calidad de los datos. La evaluación inicial incluyó la búsqueda de valores nulos, valores codificados con cero, asuntos vacíos, posibles repeticiones e inconsistencias en las relaciones entre tablas.

Tabla 16

Indicadores de calidad de los tickets

Indicador	Resultado
Total de tickets	16.993
Números de ticket nulos o vacíos	0
Tickets sin usuario	0
Tickets sin agente asignado	2
Tickets sin fecha de cierre	122
Asuntos vacíos	0
Asuntos únicos	14.116
Asuntos repetidos	2.877

Nota. Datos obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Tabla 17

Tickets sin producto y con incidencias

Indicador	Resultado
Tickets sin producto o tema de ayuda	15
Tickets sin tipo de incidencia	11.127
Tickets con tipo de incidencia válido	5.866

Nota. Datos obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Los 15 tickets sin categoría deberán revisarse durante la preparación de los datos, debido a que no poseen una etiqueta válida para el entrenamiento supervisado. Los dos tickets sin agente asignado no se consideran necesariamente errores, debido a que la asignación puede depender del estado operativo del requerimiento.

De igual manera, los 122 valores nulos en la fecha de cierre no deben interpretarse automáticamente como información defectuosa, ya que pueden corresponder a tickets abiertos al momento de generar el respaldo. La fecha de cierre tampoco será utilizada como variable predictora, porque no se encuentra disponible al momento de registrar y clasificar un ticket.

Se identificaron 2.877 registros cuyos asuntos coinciden con asuntos existentes después de convertirlos a minúsculas y eliminar espacios laterales. Sin embargo, esta coincidencia no permite concluir que se trate de tickets duplicados. Diferentes clientes pueden registrar problemas similares utilizando el mismo asunto.

La identificación de duplicados requerirá aplicar criterios adicionales, como la comparación simultánea del asunto, el cuerpo del mensaje inicial, el usuario u organización solicitante y la proximidad temporal. Este procedimiento se desarrollará en la fase de preparación de los datos.

3.9.2.7 Separación de variables descriptivas y predictoras. Para evitar errores metodológicos durante la construcción del modelo, las variables identificadas fueron clasificadas de acuerdo con la función que desempeñan dentro de la investigación. Esta separación permite distinguir la variable que se desea predecir, las características disponibles para generar la predicción y los campos empleados únicamente con fines descriptivos o de trazabilidad.

La variable objetivo corresponde al campo Incidencia de la tabla ost_ticket__cdata. Este campo almacena el tipo de incidencia asignado históricamente a cada ticket y se relaciona con la lista denominada “Tipo Incidencia”. Las clases identificadas son Soporte Funcional, Soporte Operativo, Soporte Software, Otras Incidencias, Soporte Comercial y Soporte Operativo para cobrar.

Debido a que Incidencia representa el resultado que se desea predecir, este campo no puede utilizarse como variable de entrada del modelo. Su función será proporcionar la etiqueta de los ejemplos históricos durante el entrenamiento y permitir la comparación entre las predicciones generadas y las categorías reales durante la evaluación.

Como variables predictoras candidatas se consideraron aquellas que se encuentran disponibles en el momento en que el usuario registra el ticket:

- Asunto del ticket.
- Texto del mensaje inicial.
- Producto seleccionado por el usuario, representado mediante topic_id.
- Canal de ingreso del requerimiento.
- Fecha y hora de creación, siempre que su utilización se encuentre metodológicamente justificada.

El asunto y el mensaje inicial contienen la descripción textual del problema y son las fuentes clave para determinar el tipo de incidencia. El producto seleccionado permite conocer el sistema al que pertenece el requerimiento, mientras que el canal de ingreso indica el medio utilizado para registrarlo.

A diferencia de la variable objetivo, el campo topic_id no representa el tipo de incidencia. Este campo identifica el producto o tema de ayuda seleccionado por el usuario al momento de abrir el ticket. Por tanto, puede considerarse una variable predictora, debido a que su valor ya se encuentra disponible antes de que el modelo genere la sugerencia de clasificación.

Los siguientes campos se consideran variables descriptivas o de trazabilidad:

- Identificador interno del ticket.
- Número del ticket.
- Identificador del usuario.

- Identificador de la organización.
- Identificador del departamento.
- Identificador del agente.
- Identificador del equipo.
- Estado del ticket.
- Fechas de actualización y cierre.

Los identificadores internos permiten relacionar las tablas y verificar la integridad de los datos, pero no describen el contenido del requerimiento. Por esta razón, no se incorporarán directamente como variables predictoras.

Los identificadores de usuario y organización también se conservarán para análisis descriptivos, trazabilidad y validación de las particiones de datos. Su utilización directa podría provocar que el modelo memorice comportamientos particulares de determinados clientes, en lugar de aprender patrones generalizables presentes en el contenido textual.

La inclusión definitiva de cada variable se establecerá durante la fase de preparación de los datos. Como criterio general, una característica solo podrá utilizarse como predictora si se demuestra que está disponible antes de generar la clasificación y que no revela directa o indirectamente el resultado esperado.

3.9.2.8 Prevención de fuga de información. La fuga de información se produce cuando un modelo utiliza, durante su entrenamiento, datos que no estarían disponibles en el momento real de efectuar una predicción. Este problema puede generar resultados aparentemente elevados durante las pruebas, pero un funcionamiento deficiente cuando el modelo recibe tickets nuevos.

En la presente investigación, la predicción deberá realizarse inmediatamente después de que el usuario registre el ticket. Por consiguiente, únicamente podrán utilizarse características disponibles en ese instante.

El campo ost_ticket__cdata.Incidencia constituye la variable objetivo y deberá mantenerse separado de las variables predictoras durante todas las etapas del procesamiento. Su inclusión como entrada del modelo produciría una fuga directa de información, debido a que contiene la respuesta que se pretende estimar.

También se excluirán como predictoras los campos y contenidos generados después de la creación o clasificación del ticket:

- Tipo de incidencia histórico almacenado en ost_ticket__cdata.Incidencia.
- Fecha de cierre.
- Fecha de reapertura.
- Fecha de última actualización.
- Tiempo de resolución.
- Estado final del ticket.
- Indicador de ticket respondido.
- Indicador de ticket vencido.
- Agente asignado después de la clasificación.
- Equipo asignado después de la clasificación.
- Departamento asignado como consecuencia del tratamiento del ticket.
- Respuestas elaboradas por los técnicos.
- Notas internas.
- Eventos posteriores a la creación.
- Información generada durante la resolución del requerimiento.

En el caso de la tabla ost_thread_entry, únicamente podrá utilizarse el mensaje inicial registrado por el solicitante. Las respuestas de los agentes, las notas internas y las demás entradas creadas posteriormente serán excluidas, debido a que pueden contener referencias explícitas a la clasificación, al diagnóstico o a la solución aplicada.

El producto almacenado en ost_ticket.topic_id no se considera fuga de información, debido a que el usuario lo selecciona al momento de abrir el ticket. En consecuencia, este campo puede evaluarse como predictor siempre que se confirme que el modelo realizará la clasificación después de que el producto haya sido seleccionado.

Para evitar que la información del conjunto de evaluación influya en el entrenamiento, la vectorización textual, la selección de características, el tratamiento del desbalance y cualquier transformación basada en estadísticas serán ajustados exclusivamente con los datos de entrenamiento. Posteriormente, las transformaciones aprendidas se aplicarán sin recalcular sus parámetros sobre los conjuntos de validación y prueba.

Además, cuando existan tickets repetidos o relacionados con un mismo caso, se evaluará la necesidad de mantenerlos dentro de una misma partición. Esto evitará que textos prácticamente idénticos aparezcan simultáneamente en los conjuntos de entrenamiento y evaluación, situación que podría producir una estimación artificialmente optimista del rendimiento.

La regla metodológica establecida determina que una variable solamente podrá incorporarse al modelo si estaba disponible exactamente en el momento en que el sistema debe sugerir el tipo de incidencia.

3.9.2.9 Análisis Exploratorio de Datos (EDA). Una vez identificadas las tablas y relaciones relevantes, se realizó un Análisis Exploratorio de Datos, conocido como EDA por las siglas de *Exploratory Data Analysis*. El propósito fue analizar las características generales del conjunto de datos, examinar la distribución de las variables clave e identificar problemas de calidad, previo a la fase de preparación de los datos.

El análisis se desarrolló en el notebook 02_eda.ipynb mediante Python y las bibliotecas Pandas y Matplotlib. Para garantizar la reproducibilidad, el notebook carga la

información desde el archivo tickets_eda.csv, generado previamente a partir de la copia local de MariaDB.

Durante esta fase no se eliminaron, reemplazaron ni modificaron registros. Los valores nulos, asuntos repetidos y observaciones extremas fueron únicamente cuantificados y documentados. Su tratamiento definitivo se realizará en la fase de preparación de los datos.

Código:

```
from pathlib import Path
import re
import pandas as pd
import matplotlib.pyplot as plt

ROOT = Path.cwd().parent if Path.cwd().name == 'notebooks' else Path.cwd()
DATASET = ROOT / 'data' / 'raw' / 'tickets_eda.csv'
FIGURAS = ROOT / 'outputs' / 'figuras'
TABLAS = ROOT / 'outputs' / 'tablas'
FIGURAS.mkdir(parents=True, exist_ok=True)
TABLAS.mkdir(parents=True, exist_ok=True)
plt.style.use('seaborn-v0_8-whitegrid')
COLOR = '#245b8a'
print('Dataset:', DATASET)
print('Figuras:', FIGURAS)
```

El conjunto analizado contiene 16.993 registros y 19 variables. Cada fila representa un ticket de soporte e incluye información relacionada con su fecha de creación, canal de ingreso, usuario, cliente, agente, departamento, estado, prioridad, asunto, descripción inicial, producto y tipo de incidencia.

La cobertura temporal comprende desde el 30 de diciembre de 2023 hasta el 5 de agosto de 2026. Por tanto, el respaldo permite analizar aproximadamente dos años y siete meses de actividad registrada en Webcoopec Ticket.

Código:

```
df = pd.read_csv(DATASET, low_memory=False)

df['created'] = pd.to_datetime(df['created'], errors='coerce')

print(f'Filas: {df.shape[0]:,}')

print(f'Columnas: {df.shape[1]}')

display(df.head(3).drop(columns=['subject', 'descripcion'], errors='ignore'))

display(df.dtypes.rename('tipo').to_frame())
```

La Figura 13 presenta la carga inicial y el panorama general del conjunto de datos utilizado en la investigación.

Figura 13

Carga y panorama general del conjunto de datos

```
Filas: 16,993
Columnas: 19
```

	ticket_id	numero_ticket	created	source	user_id	cliente_id	agente_id	departamento_id	departamento	status_id	estado	prioridad_codigo	prioridad	producto_id	producto	incidencia_codigo	categoria
0	1	276910	2023-12-30 20:57:03	Web	1	0	1	1	Soporte	3	Cerrado	NaN	No definida	1	Consulta general	4.0	Otras Incidencias
1	5	410133	2024-02-07 09:04:41	Web	5	3	1	3	Mantenimiento	3	Cerrado	2.0	Media	0	NaN	2.0	Soporte Operativo
2	6	201500	2024-02-07 10:30:45	Web	5	3	2	1	Soporte	3	Cerrado	3.0	Critica	11	Reportar Incidencia WEBCOOP	NaN	NaN

	tipo
ticket_id	int64
numero_ticket	int64
created	datetime64[ns]
source	object
user_id	int64
cliente_id	int64
agente_id	int64
departamento_id	int64
departamento	object
status_id	int64
estado	object
prioridad_codigo	float64
prioridad	object
subject	object
descripcion	object
producto_id	int64
producto	object
incidencia_codigo	float64
categoria	object

Nota. Resultado obtenido a partir del respaldo local de Webcoopec Ticket (2026).

Evaluación de valores nulos

La exploración inicial permitió identificar valores ausentes en determinadas variables. El principal hallazgo corresponde al tipo de incidencia, debido a que 11.127 tickets no poseen una categoría registrada. Esto representa el 65,48 % del conjunto total.

También se encontraron 15 tickets sin producto especificado y 2 registros sin descripción inicial. Estas ausencias no fueron corregidas durante el EDA, porque su tratamiento debe responder a criterios metodológicos y operativos que serán definidos durante la preparación de los datos.

Tabla 18

Principales valores nulos identificados

Variable	Registros nulos	Interpretación
Tipo de incidencia	11.127	Tickets sin etiqueta para el aprendizaje supervisado
Producto	15	Producto no especificado
Descripción inicial	2	Tickets sin contenido descriptivo recuperado

Nota. Datos obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Los resultados muestran que 5.866 tickets cuentan con un tipo de incidencia válido. Estos registros constituyen el conjunto inicialmente etiquetado, aunque su utilización definitiva dependerá de las decisiones adoptadas durante la preparación.

Código:

```
nulos = pd.DataFrame({
   'nulos': df.isna().sum(),
   'porcentaje': (df.isna().mean() * 100).round(2)
}).sort_values('nulos', ascending=False)
nulos.to_csv(TABLAS / 'eda_valores_nulos.csv', encoding='utf-8-sig')
display(nulos.head(12))
```

```
grafico_nulos = nulos[nulos['nulos'] > 0].head(10).sort_values('porcentaje')
fig, ax = plt.subplots(figsize=(9, 5))
ax.barh(grafico_nulos.index, grafico_nulos['porcentaje'], color=COLOR)
ax.set_title('Porcentaje de valores nulos por variable')
ax.set_xlabel('Porcentaje (%)')
fig.tight_layout()
fig.savefig(FIGURAS / '01_valores_nulos.png', dpi=300, bbox_inches='tight')
plt.show()
```

La Figura 14 muestra los valores nulos identificados en las principales variables del conjunto de datos analizado.

Figura 14

Principales valores nulos identificados

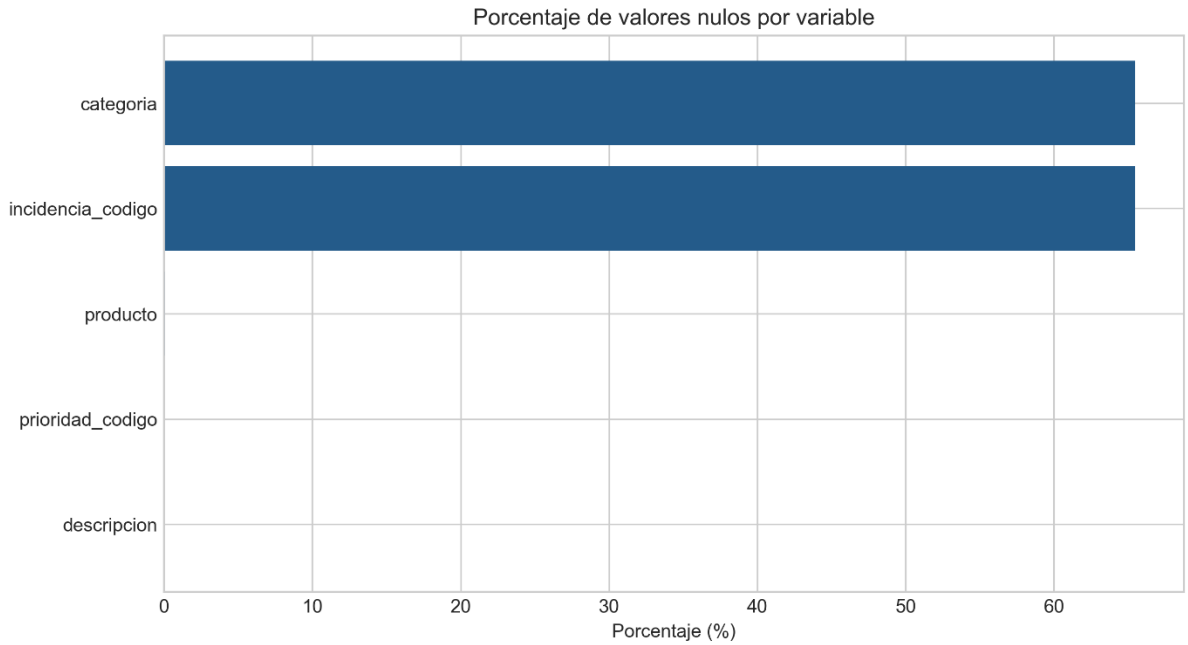
	nulos	porcentaje
categoria	11127	65.48
incidencia_codigo	11127	65.48
producto	15	0.09
descripcion	2	0.01
prioridad_codigo	1	0.01
ticket_id	0	0.00
numero_ticket	0	0.00
created	0	0.00
source	0	0.00
cliente_id	0	0.00
user_id	0	0.00
estado	0	0.00

Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

La Figura 15 presenta el porcentaje de valores nulos identificado en las principales variables del conjunto de datos analizado.

Figura 15

Porcentaje de valores nulos por variable



Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Análisis de registros repetidos

No se encontraron identificadores de ticket duplicados ni filas completamente duplicadas. Esto indica que cada registro mantiene un identificador único dentro del conjunto analizado.

No obstante, se identificaron 3.799 registros involucrados en grupos con asuntos repetidos, después de convertir los textos a minúsculas y eliminar espacios laterales. Este resultado no significa que existan 3.799 tickets duplicados. Un mismo asunto puede ser utilizado por diferentes clientes para informar problemas semejantes.

La identificación definitiva de duplicados requerirá comparar simultáneamente el asunto, la descripción inicial, el usuario, el cliente y la proximidad temporal. Este procedimiento se realizará durante la preparación de los datos.

Código:

```
duplicados_id = int(df['ticket_id'].duplicated().sum())
asunto_normalizado = df['subject'].fillna('').str.lower().str.strip()
asuntos_repetidos = int(asunto_normalizado.duplicated(keep=False).sum())
filas_exactas = int(df.duplicated().sum())
resumen_duplicados = pd.DataFrame({
   'indicador': ['Identificadores duplicados', 'Filas completamente duplicadas', 'Registros con asunto repetido'],
   'cantidad': [duplicados_id, filas_exactas, asuntos_repetidos]
})
display(resumen_duplicados)
```

La Figura 16 presenta los resultados del análisis de duplicados realizado sobre el conjunto de datos.

Figura 16

Análisis de registros duplicados

	indicador	cantidad
0	Identificadores duplicados	0
1	Filas completamente duplicadas	0
2	Registros con asunto repetido	3799

Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Distribución de la variable objetivo

La variable objetivo corresponde al tipo de incidencia almacenado en el campo Incidencia de ost_ticket__cdata. Para representar la totalidad del conjunto durante el EDA, los valores nulos o vacíos se visualizaron como “No definido”. Esta denominación se utilizó únicamente con fines descriptivos y no implica su conversión automática en una clase de entrenamiento.

La variable objetivo es categoría, obtenida del campo Incidencia. Los valores ausentes se muestran cómo No definido únicamente para describir su magnitud; no se convierten todavía en una clase de entrenamiento.

Código:

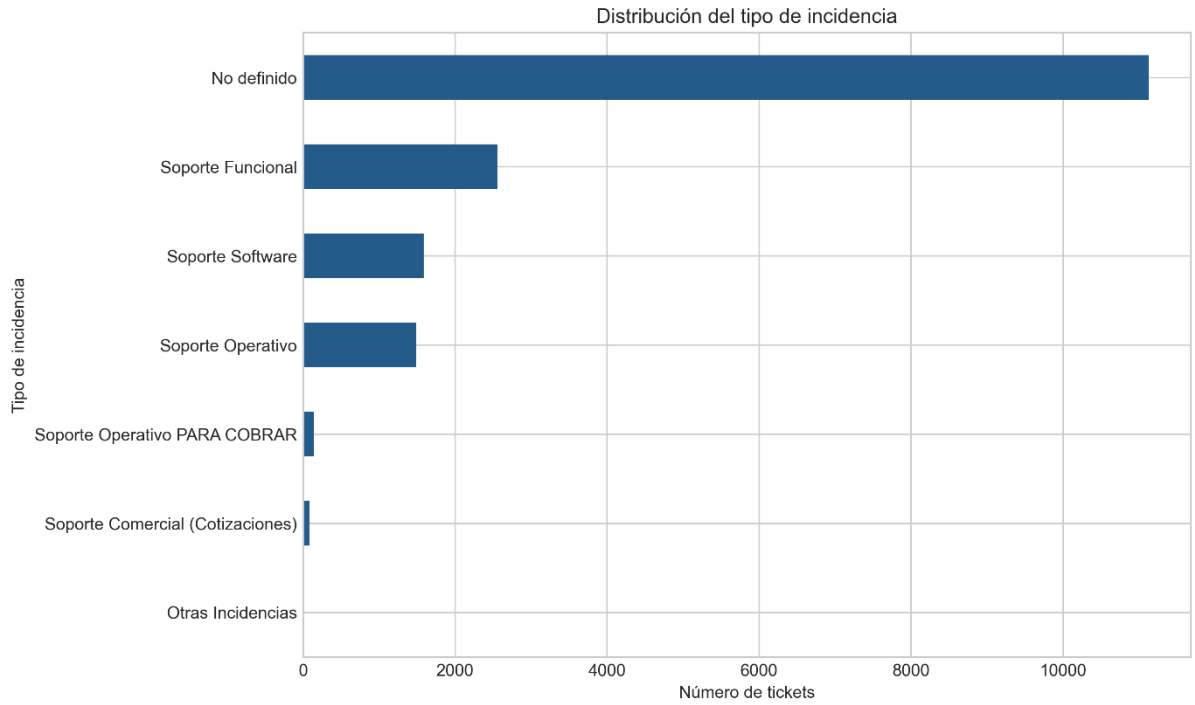
```
 categoria_eda = df['categoria'].fillna('No definido')
conteo_categoria = categoria_eda.value_counts()
tabla_categoria = pd.DataFrame({
    'tickets': conteo_categoria,
    'porcentaje': (conteo_categoria / len(df) * 100).round(2)
})
tabla_categoria.to_csv(TABLAS / 'eda_distribucion_incidencia.csv', encoding='utf-8-sig')
display(tabla_categoria)


fig, ax = plt.subplots(figsize=(10, 6))
conteo_categoria.sort_values().plot(kind='barh', ax=ax, color=COLOR)
ax.set_title('Distribución del tipo de incidencia')
ax.set_xlabel('Número de tickets')
ax.set_ylabel('Tipo de incidencia')
fig.tight_layout()
fig.savefig(FIGURAS / '02_distribucion_incidencia.png', dpi=300, bbox_inches='tight')
plt.show()
```

La Figura 17 presenta la distribución de los tickets según el tipo de incidencia registrado en el conjunto de datos analizado.

Figura 17

Distribución de los tickets según el tipo de incidencia



Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

La figura evidencia que la categoría “No definido” concentra la mayor parte de los registros. Entre los tickets etiquetados, Soporte Funcional constituye la clase mayoritaria, seguida por Soporte Software y Soporte Operativo.

Las categorías Soporte Operativo para cobrar, Soporte Comercial y Otras Incidencias presentan una cantidad considerablemente menor de observaciones. Esta distribución confirma la existencia de un desbalance de clases que deberá considerarse durante el modelado y la evaluación.

Durante el EDA no se aplicaron técnicas de balanceo, agrupación o eliminación de categorías. Estas decisiones se analizarán posteriormente y deberán considerar tanto la cantidad de ejemplos disponibles como el significado operativo de cada incidencia.

Distribución por producto

La categoría de producto almacenada mediante topic_id fue analizada separadamente de la variable objetivo. Esta distinción es necesaria porque el usuario selecciona el producto

al momento de registrar el ticket, mientras que el tipo de incidencia es la categoría que el modelo deberá predecir.

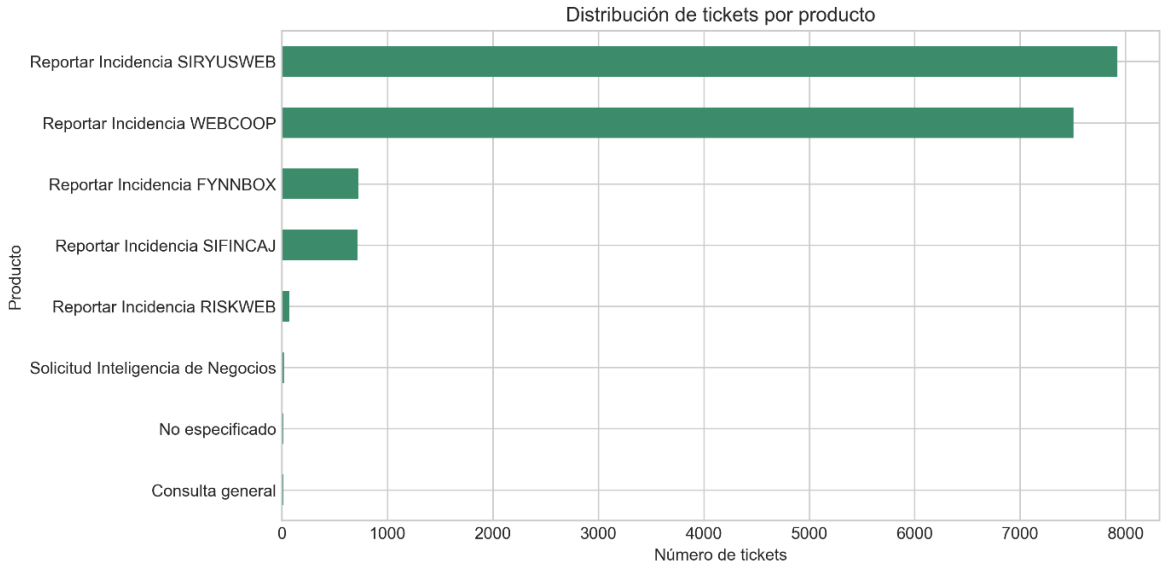
Código:

```
productos = df['producto'].fillna('No especificado').value_counts()
display(productos.rename('tickets').to_frame())
fig, ax = plt.subplots(figsize=(10, 5))
productos.sort_values().plot(kind='barh', ax=ax, color='#3c8c6c')
ax.set_title('Distribución de tickets por producto')
ax.set_xlabel('Número de tickets')
ax.set_ylabel('Producto')
fig.tight_layout()
fig.savefig(FIGURAS / '03_distribucion_producto.png', dpi=300, bbox_inches='tight')
plt.show()
```

La Figura 18 presenta la distribución de los tickets según el producto registrado en el conjunto de datos analizado.

Figura 18

Distribución de los tickets según el producto seleccionado



Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Los productos SIRYUSWEB y WEBCOOP concentran conjuntamente la mayor parte de los tickets. Los productos FYNNBOX y SIFINCAJ poseen una representación intermedia, mientras que RISKWEB e Analítica de Datos presentan cantidades inferiores.

Debido a que el producto está disponible antes de generar la clasificación, podrá evaluarse como una variable predictora durante la preparación y el modelado.

Distribución por clientes

La exploración identificó 27 organizaciones con tickets registrados dentro del respaldo. Para proteger la información institucional, el análisis utilizó identificadores anonimizados y no presentó nombres de clientes.

El gráfico de los diez clientes con mayor actividad permitió comprobar que el volumen de tickets no se distribuye uniformemente entre las organizaciones. Este hallazgo deberá considerarse al dividir los datos, debido a que una partición completamente aleatoria podría colocar tickets muy similares de un mismo cliente tanto en entrenamiento como en evaluación.

La información de cliente será utilizada con fines descriptivos, de trazabilidad y control de las particiones. No será incorporada automáticamente como variable predictora, porque podría provocar que el modelo memorice comportamientos particulares de determinadas organizaciones.

Código:

```
clientes = df['cliente_id'].fillna('Sin organización').astype(str).value_counts().head(10)

clientes.index = ['Cliente ' + valor if valor != 'Sin organización' else valor for valor in clientes.index]

fig, ax = plt.subplots(figsize=(9, 5))

clientes.sort_values().plot(kind='barh', ax=ax, color='#a66a3f')

ax.set_title('Diez clientes con mayor número de tickets')

ax.set_xlabel('Número de tickets')

ax.set_ylabel('Cliente anonimizado')

fig.tight_layout()

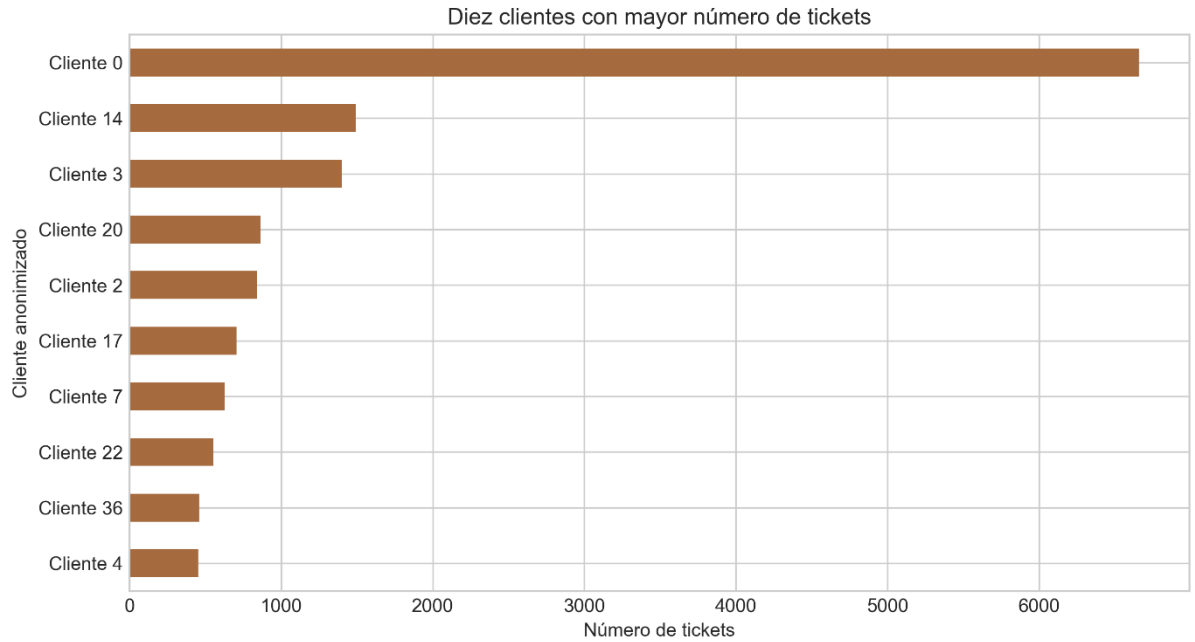
fig.savefig(FIGURAS / '04_principales_clientes.png', dpi=300, bbox_inches='tight')

plt.show()
```

La Figura 19 presenta la distribución de los diez clientes con mayor número de tickets registrados en el conjunto de datos analizado.

Figura 19

Distribución de los diez clientes con mayor número de tickets



Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

Evolución temporal de los tickets

Se realizó una agregación mensual utilizando la fecha de creación de cada ticket. Este análisis permitió observar las variaciones generales del volumen de requerimientos durante el periodo cubierto por el respaldo.

Código:

```
tickets_mes = df.dropna(subset=['created']).set_index('created').resample('MS').size()

fig, ax = plt.subplots(figsize=(11, 5))

ax.plot(tickets_mes.index, tickets_mes.values, marker='o', linewidth=1.8, color=COLOR)

ax.set_title('Evolución mensual del número de tickets')

ax.set_xlabel('Mes')

ax.set_ylabel('Número de tickets')

fig.tight_layout()

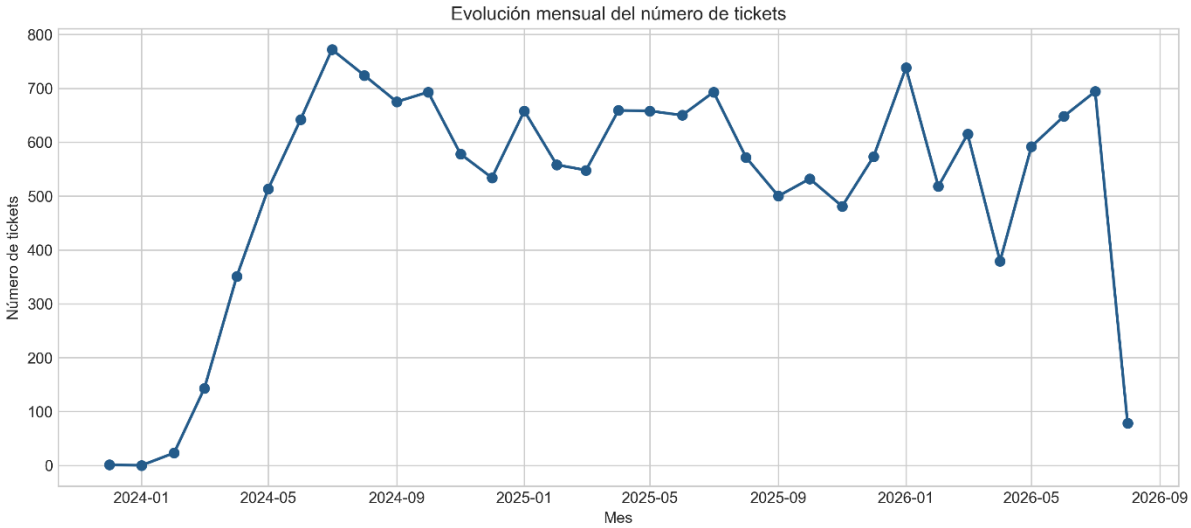
fig.savefig(FIGURAS / '05_evolucion_mensual.png', dpi=300, bbox_inches='tight')
```

plt.show()

La Figura 20 presenta la evolución mensual del número de tickets registrados durante el período analizado.

Figura 20

Evolución mensual del número de tickets registrados



Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

La serie temporal muestra que el volumen de tickets varía entre los meses analizados. Estas diferencias pueden estar relacionadas con la actividad operativa de los clientes, actualizaciones de los sistemas, cierres contables, implementación de nuevas funcionalidades o periodos de mayor demanda.

En esta fase no se interpretaron estas variaciones como valores atípicos ni se eliminaron periodos con mayor o menor actividad. Su posible utilidad como característica temporal será evaluada posteriormente.

Longitud del asunto y la descripción

Como parte de la caracterización textual, se calculó la longitud del asunto y de la descripción inicial en número de caracteres.

El asunto presenta una longitud media de 29,86 caracteres y una mediana de 30. El valor mínimo registrado fue de 2 caracteres y el máximo de 50.

La descripción inicial presenta una longitud media de 534,97 caracteres y una mediana de 385. El valor máximo alcanza 49.606 caracteres, lo que evidencia la existencia de descripciones considerablemente más extensas que la mayoría.

Código:

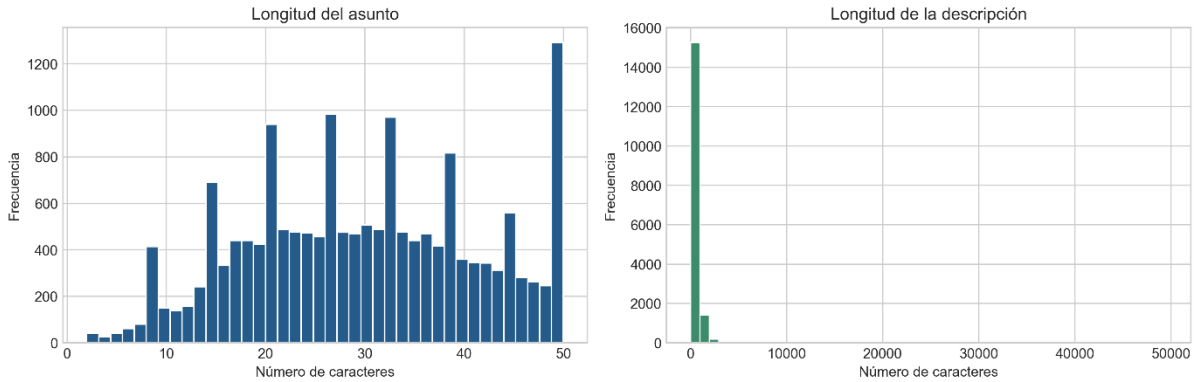
```
df['longitud_asunto'] = df['subject'].fillna('').astype(str).str.len()
df['longitud_descripcion'] = df['descripcion'].fillna('').astype(str).str.len()
display(df[['longitud_asunto', 'longitud_descripcion']].describe().round(2))

fig, axes = plt.subplots(1, 2, figsize=(12, 4))
axes[0].hist(df['longitud_asunto'], bins=40, color=COLOR, edgecolor='white')
axes[0].set_title('Longitud del asunto')
axes[0].set_xlabel('Número de caracteres')
axes[0].set_ylabel('Frecuencia')
axes[1].hist(df['longitud_descripcion'], bins=50, color='#3c8c6c', edgecolor='white')
axes[1].set_title('Longitud de la descripción')
axes[1].set_xlabel('Número de caracteres')
axes[1].set_ylabel('Frecuencia')
fig.tight_layout()
fig.savefig(FIGURAS / '06_longitudes_texto.png', dpi=300, bbox_inches='tight')
plt.show()
```

La Figura 21 presenta la distribución de la longitud del asunto y de la descripción inicial de los tickets analizados.

Figura 21

Distribución de la longitud del asunto y de la descripción inicial



Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

La divergencia entre la media y la mediana de la descripción indica una distribución asimétrica influenciada por textos extensos. Sin embargo, estos casos no fueron eliminados ni considerados errores durante el EDA. En la preparación de los datos se deberá revisar si corresponden a descripciones válidas, contenido HTML, mensajes reenviados, firmas, historiales de correo u otra información que requiera limpieza.

Conclusiones del análisis exploratorio

El análisis exploratorio permitió determinar que el conjunto contiene 16.993 tickets y 19 variables, con una cobertura temporal comprendida entre diciembre de 2023 y agosto de 2026.

El principal problema identificado corresponde a la ausencia del tipo de incidencia en 11.127 registros. En consecuencia, solamente 5.866 tickets se encuentran inicialmente etiquetados para un proceso de clasificación supervisada.

También se comprobó la existencia de un desbalance entre las clases etiquetadas. Soporte Funcional, Soporte Software y Soporte Operativo concentran la mayoría de las observaciones, mientras que Soporte Comercial, Soporte Operativo para cobrar y Otras Incidencias poseen una representación reducida.

No se encontraron identificadores duplicados ni filas completamente repetidas. Sin embargo, 3.799 registros pertenecen a grupos con asuntos coincidentes, por lo que será necesario aplicar criterios adicionales antes de determinar la existencia de tickets duplicados.

El análisis de las longitudes textuales mostró que los asuntos son relativamente breves, mientras que las descripciones presentan una mayor variabilidad y algunos valores considerablemente extensos. Estos resultados justifican una revisión del contenido durante la limpieza textual.

En conclusión, el conjunto posee información suficiente para continuar con la preparación de los datos, pero requiere analizar el tratamiento de los tickets sin incidencia, las categorías minoritarias, los asuntos repetidos y las descripciones ausentes o excesivamente extensas. Estas decisiones se desarrollarán en la siguiente fase de CRISP-DM sin alterar la información original durante la comprensión de los datos.

3.9.3 Preparación de los datos

La preparación de los datos representa la tercera fase del enfoque metodológico CRISP-DM. Su finalidad es transformar la información examinada durante el análisis exploratorio en un conjunto consistente, anonimizado y adecuado para la capacitación de modelos de aprendizaje automático (Chapman et al., 2000).

En la presente investigación, esta fase se implementó mediante el notebook 03_preprocesamiento.ipynb y un conjunto de módulos reutilizables desarrollados en Python. Esta organización permite aplicar las mismas operaciones durante el entrenamiento y, posteriormente, al procesar tickets nuevos desde el sistema Actybase Web.

El archivo original tickets_eda.csv se mantuvo sin modificaciones en la carpeta data/raw. Todas las transformaciones se realizaron sobre copias en memoria y sus resultados se almacenaron en data/processed. De esta manera, se preservó la trazabilidad entre la información original, las decisiones de preparación y el conjunto final.

3.9.3.1 Construcción del dataset. La unidad de análisis fue definida como un ticket de soporte técnico. En consecuencia, cada fila del dataset representa un ticket y reúne la información disponible desde el momento en que el usuario registra el requerimiento.

El dataset inicial contiene 16.993 registros y 19 variables. Para construirlo se conservaron los siguientes elementos:

- Identificador interno del ticket.
- Fecha de creación.
- Canal de ingreso.
- Usuario solicitante.
- Organización cliente.
- Agente y departamento.
- Estado y prioridad.
- Asunto del ticket.
- Descripción inicial.
- Producto seleccionado.
- Código y descripción del tipo de incidencia.

El asunto y la descripción inicial constituyen las principales fuentes textuales para el modelo. El producto seleccionado por el usuario puede utilizarse como característica predictora, debido a que está disponible antes de generar la clasificación.

Por otra parte, el tipo de incidencia representa la variable objetivo. Por tanto, se utiliza únicamente como etiqueta durante el entrenamiento y la evaluación.

No se incorporaron respuestas de los técnicos, notas internas, tiempos de resolución, estados finales ni otros campos generados después de la creación del ticket. Esta exclusión evita que el conjunto incluya datos que no estarían accesibles al hacer una predicción en tiempo real.

Código:

```
df_raw = pd.read_csv(RAW, low_memory=False, parse_dates=['created'])
print(f'Registros originales: {len(df_raw):,}')
print(f'Variables originales: {df_raw.shape[1]}')
display(df_raw[['ticket_id','created','source','producto','categoria']].head())
```

La Figura 22 muestra la construcción inicial del conjunto de datos de tickets utilizado para las etapas posteriores de preparación y modelado.

Figura 22

Construcción inicial del dataset de tickets

```
Registros originales: 16,993
Variables originales: 19
```

	ticket_id	created	source	producto	categoria
0	1	2023-12-30 20:57:03	Web	Consulta general	Otras Incidencias
1	5	2024-02-07 09:04:41	Web	NaN	Soporte Operativo
2	6	2024-02-07 10:30:45	Web	Reportar Incidencia WEBCOOP	NaN
3	7	2024-02-07 15:26:26	Web	NaN	Soporte Operativo
4	8	2024-02-07 17:45:54	Web	NaN	NaN

Nota. Resultado obtenido a partir del respaldo local de Webcoopec Ticket (2026).

3.9.3.2 Integración de tablas. La información necesaria para construir el dataset se encontraba distribuida en diferentes tablas de osTicket. Por este motivo, fue necesario integrar las siguientes fuentes:

Tabla 19

Integración de tablas

Tabla	Información incorporada
ost_ticket	Identificador, fechas, canal, usuario, agente, departamento, estado y producto
ost_ticket__cdata	Asunto, prioridad y tipo de incidencia
ost_help_topic	Descripción del producto seleccionado
ost_user	Relación con el usuario y la organización

Tabla	Información incorporada
ost_department	Nombre del departamento
ost_ticket_status	Descripción del estado
ost_thread	Relación entre el ticket y su conversación
ost_thread_entry	Mensaje inicial del solicitante
ost_list_items	Descripción correspondiente al código de incidencia

Nota. Las tablas corresponden al respaldo local de Webcoopec Ticket (2026).

La integración se realizó mediante uniones izquierdas, tomando ost_ticket como tabla central. Este tipo de unión permitió conservar todos los tickets, incluso cuando una tabla relacionada no contenía información correspondiente.

Para obtener la descripción inicial se relacionaron las tablas ost_thread y ost_thread_entry. De cada ticket se seleccionó exclusivamente la primera entrada de tipo mensaje. Las respuestas posteriores y las notas internas fueron excluidas para impedir fuga de información.

El campo topic_id de ost_ticket fue interpretado como la categoría de producto seleccionada por el usuario. En cambio, el campo Incidencia de ost_ticket__cdata fue relacionado con ost_list_items para recuperar la etiqueta de la variable objetivo.

Código:

```
columnas_integradas = ['ticket_id','created','source','user_id','cliente_id','agente_id',
'departamento','estado','prioridad','subject','descripcion',
'producto','incidencia_codigo','categoria']
display(pd.DataFrame({'variable_integrada': columnas_integradas}))
```

La Figura 23 presenta las variables obtenidas después de integrar las tablas utilizadas para construir el conjunto de datos de análisis.

Figura 23

Variables obtenidas mediante la integración de las tablas de osTicket

	variable_integrada
0	ticket_id
1	created
2	source
3	user_id
4	cliente_id
5	agente_id
6	departamento
7	estado
8	prioridad
9	subject
10	descripcion
11	producto
12	incidencia_codigo
13	categoria

Nota. Resultado obtenido a partir de la integración de las tablas del respaldo local de Webcoopec Ticket (2026).

3.9.3.3 Limpieza de datos. La limpieza inicial se concentró en retirar elementos de formato que no aportan información semántica al proceso de clasificación.

Las descripciones almacenadas en osTicket podían contener etiquetas HTML y entidades codificadas. Por este motivo, se aplicaron las siguientes operaciones:

- Conversión de entidades HTML a caracteres legibles.
- Eliminación de etiquetas de presentación.
- Sustitución de saltos y espacios repetidos.
- Unión controlada del asunto con la descripción inicial.
- Conversión de valores textuales ausentes en cadenas vacías cuando correspondía.

La eliminación de etiquetas HTML no implicó la eliminación del contenido que se encontraba dentro de ellas. Por ejemplo, una expresión almacenada como Error al generar órdenes fue transformada en Error al generar órdenes.

3.9.3.4 Anonimización. Debido a que los tickets pueden contener información sensible de usuarios y organizaciones, se implementaron mecanismos de anonimización y seudonimización antes de construir el conjunto final.

Los identificadores internos fueron reemplazados por códigos generados mediante HMAC-SHA256. Este procedimiento permite mantener un identificador estable para trazabilidad sin conservar directamente el número original del ticket.

La clave utilizada para generar estos códigos se almacenó fuera del código fuente mediante una variable de entorno. Por consiguiente, no se encuentra incorporada en los notebooks ni en los módulos compartidos.

Dentro del asunto y la descripción se enmascararon los siguientes patrones:

- Direcciones de correo electrónico.
- Direcciones web.
- Direcciones IP.
- Números telefónicos.
- Números de documentos.

En lugar de eliminar completamente estas expresiones, se sustituyeron por marcadores como TOKEN_EMAIL, TOKEN_URL, TOKEN_IP, TOKEN_TELEFONO y TOKEN_DOCUMENTO. Este tratamiento permite conservar el tipo general de información sin exponer su valor.

La Figura 24 muestra un ejemplo del proceso de anonimización aplicado al contenido textual.

Figura 24

Ejemplo del proceso de anonimización textual

```
ejemplo_privado = 'Escribir a usuario@ejemplo.com desde 192.168.1.10 o llamar al 0991234567.'
print(anonimizar_texto(ejemplo_privado))
```

```
Escribir a TOKEN_EMAIL desde TOKEN_IP o llamar al TOKEN_DOCUMENTO .
```

3.9.3.5 Tratamiento de nulos. El tratamiento de valores nulos se realizó de acuerdo con el significado de cada variable.

De los 16.993 tickets originales, 11.127 no poseen un tipo de incidencia registrado. Estos casos no pueden utilizarse directamente para entrenar un modelo supervisado, debido a que carecen de la etiqueta que el modelo debe aprender.

Los tickets sin incidencia no fueron eliminados definitivamente. Se separaron del conjunto de entrenamiento y se conservaron en el archivo tickets_sin_etiqueta_auditoria.csv. Este archivo permitirá revisar los registros y considerar un proceso futuro de etiquetado manual o semiautomático.

Los 5.866 tickets que sí poseen una incidencia fueron conservados inicialmente como candidatos para el aprendizaje supervisado.

También se identificaron dos tickets sin descripción inicial. En estos casos, la descripción se representó mediante una cadena vacía y se conservó el asunto como fuente textual. Esta decisión evita descartar un ticket que todavía contiene información potencialmente útil.

Los 15 productos ausentes fueron representados mediante la etiqueta Producto no especificado. Este valor permite distinguir la ausencia del producto sin confundirla con una categoría válida.

Tabla 20

Tratamiento de valores nulos

Condición	Registros	Tratamiento
Tickets sin incidencia	11.127	Separados y conservados para auditoría
Tickets con incidencia	5.866	Conservados para preparación supervisada
Descripción inicial ausente	2	Sustituida por cadena vacía; se conserva el asunto

Condición	Registros	Tratamiento
Producto ausente	15	Marcado como “Producto no especificado”

3.9.3.6 Eliminación de duplicados. La identificación de duplicados se realizó después de unir y normalizar el asunto y la descripción inicial. No se consideró suficiente comparar únicamente los asuntos, debido a que diferentes usuarios pueden registrar títulos similares.

Primero se buscaron textos normalizados idénticos asociados con categorías diferentes. Se encontraron 11 registros en esta condición. Estos casos fueron separados para revisión, ya que un mismo contenido con etiquetas contradictorias introduce ambigüedad durante el aprendizaje.

Posteriormente, se identificaron textos idénticos que compartían la misma categoría. Se eliminaron 12 repeticiones exactas y se conservó la primera ocurrencia de acuerdo con la fecha de creación.

Esta regla evita que un mismo texto aparezca repetido dentro del conjunto y pueda distribuirse simultáneamente entre entrenamiento y prueba. Tal situación produciría una evaluación artificialmente optimista.

La Tabla 21 presenta el resultado del proceso de depuración de duplicados aplicado al conjunto de tickets etiquetados.

Tabla 21

Resultado del tratamiento de duplicados

Operación	Registros
Tickets etiquetados inicialmente	5.866
Registros con etiquetas contradictorias	11
Duplicados exactos eliminados	12
Tickets resultantes	5.843

3.9.3.7 Normalización del texto. La normalización tuvo como finalidad reducir diferencias de escritura que no modifican el significado del requerimiento.

Se aplicaron las siguientes operaciones:

- Unión del asunto y la descripción inicial.
- Normalización de caracteres Unicode.
- Conversión a minúsculas.
- Unificación de signos de puntuación y separadores.
- Eliminación de espacios repetidos.
- Conservación de letras acentuadas y de la letra ñ.

Las tildes se conservaron porque forman parte de la escritura correcta en español y pueden distinguir palabras diferentes. Asimismo, la normalización no eliminó expresiones negativas ni términos técnicos propios de los productos de Webcoopec.

La unión del asunto y la descripción permite que el modelo considere tanto el resumen del requerimiento como la explicación proporcionada por el usuario.

3.9.3.8 Tokenización. La tokenización implica segmentar el texto en unidades lingüísticas conocidas como tokens. Para este proceso se utilizó la biblioteca spaCy y el modelo de español es_core_news_sm.

A diferencia de una división basada únicamente en espacios, el tokenizador identifica palabras, signos y otras unidades considerando reglas lingüísticas del idioma español.

La tokenización constituye un paso previo para evaluar las stopwords y obtener los lemas. Las reglas fueron encapsuladas en el módulo preprocesamiento.py, de manera que puedan aplicarse de forma idéntica durante el entrenamiento y cuando Actybase Web reciba un ticket nuevo.

3.9.3.9 Stopwords. Las stopwords son palabras frecuentes que cumplen principalmente funciones gramaticales y que, en determinados problemas de clasificación, poseen una capacidad discriminativa reducida.

Durante el procesamiento se eliminaron las stopwords reconocidas por el modelo de español de spaCy. Sin embargo, se conservaron expresamente las palabras no, nunca y sin.

La conservación de estas expresiones responde a su importancia en el contexto técnico. Por ejemplo, las frases “genera comprobante” y “no genera comprobante” describen situaciones opuestas. Eliminar la negación podría alterar el significado y perjudicar la clasificación.

No se incorporó todavía una lista extensa de palabras propias del dominio. La posible eliminación de términos frecuentes como saludos o fórmulas de cortesía deberá justificarse mediante el análisis del vocabulario de entrenamiento.

3.9.3.10 Lematización. La lematización convierte las palabras flexionadas en su forma base o lema. Por ejemplo, las formas verbales generando, generó y generan pueden ser representadas a través de la forma lematizada del verbo generar.

Para esta tarea se utilizó el modelo lingüístico es_core_news_sm de spaCy. Durante el proceso se excluyeron signos de puntuación, espacios y stopwords, mientras que las negaciones fueron conservadas.

La lematización permite reducir la dispersión del vocabulario y agrupar diferentes formas gramaticales que comparten un significado principal. Esto resulta especialmente útil antes de aplicar TF-IDF.

Código:

```
procesador = ProcesadorEspanol()
ejemplo_serie = pd.Series(['Los usuarios no pudieron generar comprobantes correctamente.'])
print('Texto procesado:', procesador.transformar(ejemplo_serie).iloc[0])
```

```
print('Procesando el dataset completo...')
deduplicados['texto_procesado'] = procesador.transformar(deduplicados['texto_normalizado'])
print('Proceso lingüístico completado.')
```

La Figura 25 muestra un ejemplo del resultado obtenido durante el proceso de tokenización, eliminación de stopwords y lematización aplicado al texto de los tickets.

Figura 25

Ejemplo de tokenización, eliminación de stopwords y lematización

```
Texto procesado: usuario no poder generar comprobant correctamente
Procesando el dataset completo...
Proceso lingüístico completado.
```

3.9.3.11 Vectorización TF-IDF. Los algoritmos tradicionales de Machine Learning requieren una representación numérica de los textos. Por esta razón, se utilizó TF-IDF, cuyas siglas corresponden a *Term Frequency–Inverse Document Frequency*.

Esta técnica da más importancia a los términos de alta frecuencia en un ticket concreto, pero poco comunes en todos los documentos. De esta forma, se reduce la influencia de palabras comunes y se destacan términos potencialmente discriminativos.

Antes de ajustar TF-IDF, el dataset se fragmentó de forma estratificada en dos subconjuntos:

- 80 % para entrenamiento.
- 20 % para prueba.

La división estratificada permite conservar, en la medida de lo posible, la proporción de las categorías existentes.

El vectorizador fue ajustado exclusivamente con los textos del conjunto de entrenamiento. Posteriormente, el conjunto de prueba fue transformado utilizando el

vocabulario y las ponderaciones aprendidas durante el entrenamiento. Esta secuencia evita que la información del conjunto de prueba influya en la construcción de las características.

La configuración utilizada fue:

Código:

```
resultado_vectorizacion = vectorizar_y_guardar(deduplicados, random_state=42)
display(pd.Series(resultado_vectorizacion, name='resultado').to_frame())
```

La Figura 26 presenta los resultados obtenidos durante la vectorización TF-IDF, incluyendo la distribución de los datos de entrenamiento y prueba y el tamaño del vocabulario generado.

Figura 26

Vectorización TF-IDF

	resultado
train	4674
test	1169
vocabulario	19602
x_train_shape	(4674, 19602)
x_test_shape	(1169, 19602)

Tabla 22

Vectorización

Parámetro	Valor
Rango de n-gramas	Unigramas y bigramas
Frecuencia documental mínima	2
Frecuencia documental máxima	0,95
Máximo de características	20.000
Normalización	L2
Escalamiento sublineal	Activado
Semilla de reproducibilidad	42

El vocabulario resultante contiene 19.602 características. La matriz de entrenamiento posee 4.674 filas y 19.602 columnas, mientras que la matriz de prueba contiene 1.169 filas y el mismo número de características.

El vectorizador fue guardado en models/tfidf.pkl. Este artefacto deberá utilizarse sin volver a ajustarlo cuando Actybase Web procese tickets nuevos.

3.9.3.12 Dataset final. Después de separar los tickets sin etiqueta, revisar los conflictos y eliminar los duplicados exactos, el dataset final quedó conformado por 5.843 tickets.

La Tabla 23 presenta la distribución de los tickets por categoría después del proceso de preprocesamiento.

Tabla 23

Distribución por categoría después del preprocesamiento

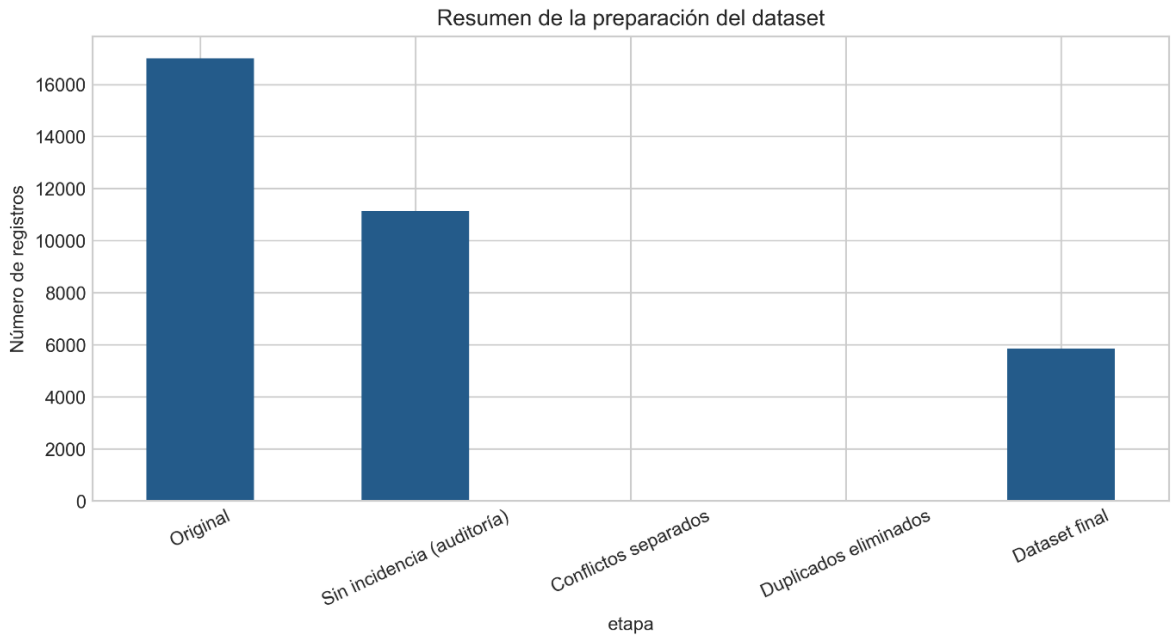
Tipo de incidencia	Registros
Soporte Funcional	2.551
Soporte Software	1.583
Soporte Operativo	1.482
Soporte Operativo para cobrar	139

Tipo de incidencia	Registros
Soporte Comercial	85
Otras Incidencias	3
Total	**5.843**

La Figura 27 presenta un resumen de las principales etapas de preparación del conjunto de datos utilizado en la investigación.

Figura 27

Resumen de la preparación del dataset

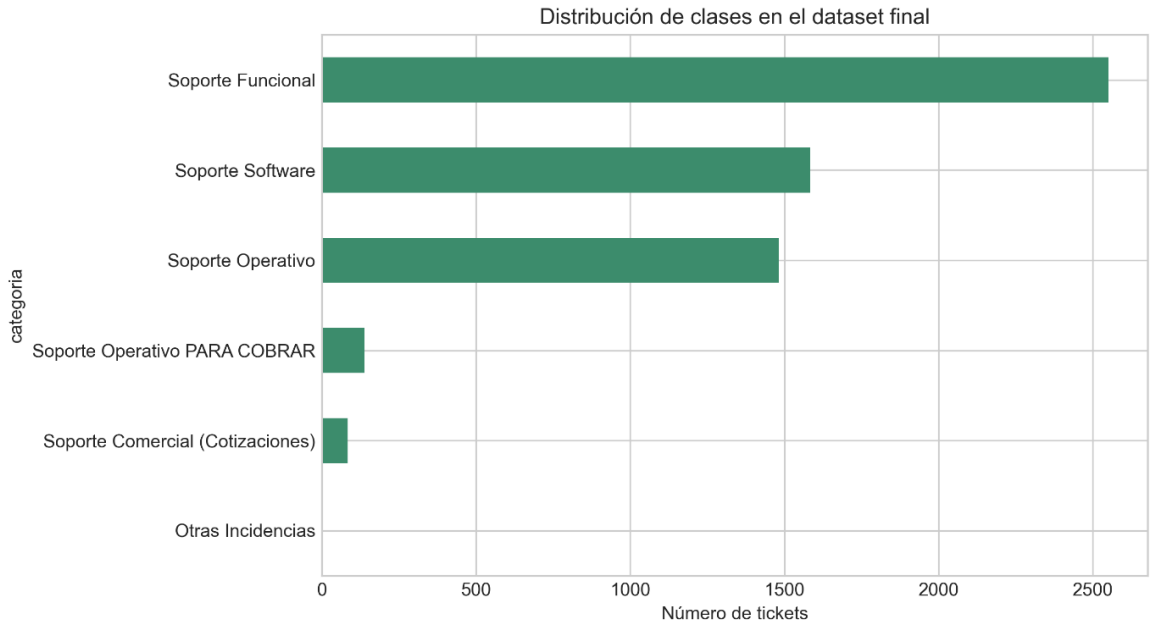


Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

La Figura 28 presenta la distribución de las clases que conforman el dataset final.

Figura 28

Distribución de las clases en el dataset final



Nota. Resultados obtenidos a partir del respaldo local de Webcoopec Ticket (2026).

El conjunto final fue dividido en 4.674 tickets de entrenamiento y 1.169 tickets de prueba. Cada registro contiene:

- Identificador seudonimizado.
- Fecha de creación.
- Canal de ingreso.
- Producto seleccionado.
- Categoría objetivo.
- Texto anonimizado.
- Texto normalizado.
- Texto tokenizado y lematizado.
- Partición asignada.

También se generaron las matrices X_train_tfidf.npz y X_test_tfidf.npz, así como los archivos y_train.csv y y_test.csv. El codificador de categorías se guardó en models/label_encoder.pkl.

La categoría Otras Incidencias conserva únicamente tres registros. Aunque fue mantenida para respetar la información original, esta cantidad resulta insuficiente para obtener una evaluación estadísticamente confiable. Su tratamiento deberá analizarse durante el modelado y validarse con los responsables de Webcoopec.

Finalmente, las funciones de limpieza, anonimización, normalización, lematización y vectorización fueron integradas en los módulos del proyecto. Esto permitirá que el futuro componente de Actybase Web aplique exactamente la misma secuencia a cada ticket nuevo antes de solicitar una predicción al modelo.

3.9.4 Modelado

La cuarta fase de la metodología CRISP-DM corresponde al modelado. En esta etapa se construyeron tres modelos de aprendizaje automático orientados a la categorización automática de tickets de soporte técnico a partir de su contenido textual.

Los algoritmos seleccionados para la investigación fueron Naive Bayes, Support Vector Machine y Random Forest. Estos algoritmos fueron entrenados bajo las mismas condiciones, utilizando el conjunto de entrenamiento y la representación numérica TF-IDF obtenida durante la preparación de los datos.

El proceso se desarrolló en el notebook 04_modelado.ipynb y en el módulo reutilizable entrenamiento.py. Esta separación permite conservar el análisis documentado en el notebook y, al mismo tiempo, disponer de funciones que podrán integrarse posteriormente con el sistema Actybase Web.

Durante esta fase no se realizó la comparación definitiva de los algoritmos ni se seleccionó un modelo ganador. El cálculo de accuracy, precisión, recall, F1 y matrices de confusión se desarrollará en la fase 3.9.5, correspondiente a la evaluación.

3.9.4.1 Conjunto utilizado para el entrenamiento. Los tres algoritmos fueron entrenados mediante la partición de entrenamiento construida en la fase anterior. Esta partición contiene 4.674 tickets representados mediante 19.602 características TF-IDF.

La matriz de entrenamiento presenta las siguientes dimensiones:

X_train = 4.674 filas × 19.602 características

y_train = 4.674 etiquetas

Cada fila de la matriz representa un ticket, mientras que cada columna corresponde a un término o combinación de términos identificados por el vectorizador TF-IDF.

El conjunto de prueba, compuesto por 1.169 tickets, permaneció completamente separado durante esta fase. Sus registros no fueron utilizados para configurar, entrenar o seleccionar ninguno de los algoritmos.

Código:

```
X_train, y_train = cargar_entrenamiento()
encoder = joblib.load(ROOT / 'models' / 'label_encoder.pkl')
print('Forma de X_train:', X_train.shape)
print('Forma de y_train:', y_train.shape)
print('Conjunto de prueba consultado: NO')
```

La Figura 29 muestra la carga de las matrices utilizadas para el entrenamiento de los modelos.

Figura 29

Carga de las matrices utilizadas para entrenar los categorizadores

```
Forma de X_train: (4674, 19602)
Forma de y_train: (4674,)
Conjunto de prueba consultado: NO
```

Esta separación evita que los modelos conozcan anticipadamente los casos que posteriormente serán utilizados para evaluar su capacidad de categorizar tickets no empleados durante el entrenamiento.

3.9.4.2 Distribución de las categorías de entrenamiento. La variable objetivo contiene seis tipos de incidencia. Su distribución dentro del conjunto de entrenamiento se presenta en la Tabla 24.

Tabla 24

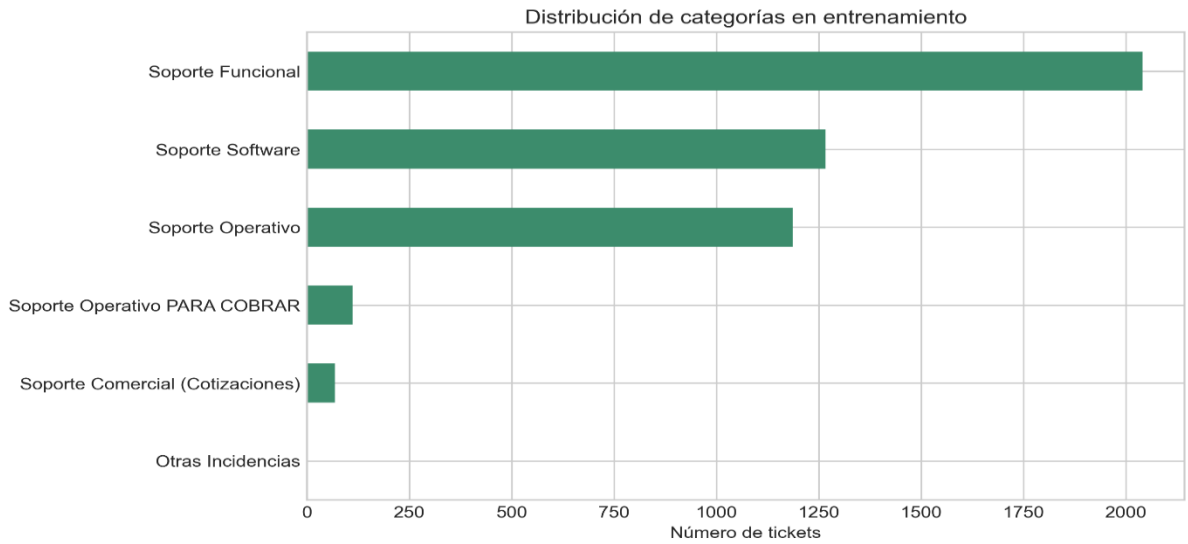
Distribución de las categorías en el conjunto de entrenamiento

Tipo de incidencia	Tickets
Soporte Funcional	2.041
Soporte Software	1.266
Soporte Operativo	1.186
Soporte Operativo para cobrar	111
Soporte Comercial	68
Otras Incidencias	2
Total	**4.674**

La Figura 30 presenta la distribución de los tipos de incidencia en el conjunto de entrenamiento utilizado para los modelos de clasificación.

Figura 30

Distribución de los tipos de incidencia en el conjunto de entrenamiento



La distribución evidencia un desbalance considerable. Las categorías Soporte Funcional, Soporte Software y Soporte Operativo concentran la mayor parte de los registros. Por el contrario, Soporte Comercial, Soporte Operativo para cobrar y Otras Incidencias disponen de una cantidad reducida de ejemplos.

El desbalance debe considerarse al interpretar posteriormente las métricas. Un categorizador podría alcanzar un número elevado de aciertos sobre las clases mayoritarias y presentar un comportamiento deficiente sobre las categorías con menos observaciones.

La categoría Otras Incidencias contiene únicamente dos casos en entrenamiento. Esta cantidad no permite que los algoritmos aprendan de forma estable sus características textuales. Por tanto, los resultados correspondientes a esta categoría deberán interpretarse con especial precaución durante la evaluación.

3.9.4.3 Selección de algoritmos. Los algoritmos incluidos en el estudio fueron seleccionados por representar enfoques diferentes de clasificación supervisada y por su aplicación en problemas de categorización textual.

Los tres algoritmos utilizados fueron:

- Complement Naive Bayes.
- Support Vector Machine con kernel lineal.
- Random Forest.

Cada modelo recibió la misma matriz TF-IDF y las mismas etiquetas de entrenamiento. De esta manera, la evaluación posterior permitirá efectuar una comparación bajo condiciones equivalentes.

No se incorporaron regresión logística ni otros algoritmos, debido a que la investigación delimitó la comparación a Naive Bayes, SVM y Random Forest.

3.9.4.4 Modelado mediante Naive Bayes. El algoritmo de Naive Bayes se fundamenta en el teorema de Bayes y se caracteriza por suposiciones de independencia condicional entre las variables correspondientes.

Para esta investigación se utilizó Complement Naive Bayes. Esta variante fue diseñada para optimizar el comportamiento de Naive Bayes ante conjuntos de texto con clases desbalanceadas (Rennie et al., 2003). Además, es compatible con las características no negativas generadas mediante TF-IDF.

El modelo fue configurado con un parámetro de suavizado alpha igual a 1,0.

Tabla 25

Configuración de Naive Bayes

Parámetro	Valor
Algoritmo	Complement Naive Bayes
Suavizado alpha	1,0
Registros de entrenamiento	4.674
Características	19.602

El modelo fue ajustado mediante la matriz X_train_tfidf y las categorías almacenadas en y_train. Una vez completado el entrenamiento, el artefacto fue guardado como:

models/modelo_naive_bayes.pkl

Este archivo conserva el modelo entrenado y permitirá utilizarlo posteriormente sobre el conjunto de prueba y durante la integración con Actybase Web, en caso de que sea seleccionado en la evaluación.

3.9.4.5 Modelado mediante Support Vector Machine. Support Vector Machine es un algoritmo que busca construir un hiperplano capaz de separar las observaciones pertenecientes a diferentes categorías (Cortes & Vapnik, 1995).

Para el tratamiento de los tickets se utilizó una SVM lineal, debido a que las matrices TF-IDF contienen un número elevado de características y una gran proporción de valores iguales a cero. En este tipo de representación, el kernel lineal suele ser más apropiado y computacionalmente eficiente que kernels de mayor complejidad.

El parámetro de regularización C fue establecido en 1,0. También se activó la ponderación balanceada de clases, con el objetivo de asignar una mayor importancia relativa a las categorías con menos registros.

Tabla 26

Configuración de Support Vector Machine

Parámetro	Valor
Algoritmo	SVM lineal
Parámetro C	1,0
Ponderación de clases	Balanceada
Semilla	42
Registros de entrenamiento	4.674
Características	19.602

La semilla 42 fue utilizada para mantener la reproducibilidad del procedimiento. El modelo entrenado fue almacenado como:

models/modelo_svm.pkl

3.9.4.6 Modelado mediante Random Forest. El algoritmo Random Forest es una técnica de ensamblaje que genera múltiples árboles de decisión y combina las predicciones de estos para clasificar datos en distintas categorías (Breiman, 2001).

A diferencia de Naive Bayes y SVM, Random Forest puede representar relaciones no lineales entre las características. Su inclusión permite comparar un modelo basado en árboles con algoritmos utilizados habitualmente en clasificación textual.

El modelo fue configurado con 300 árboles. En cada división se consideró la raíz cuadrada del número de características disponibles. También se utilizó una ponderación balanceada por submuestra para reducir el efecto del desbalance.

Tabla 27

Configuración de Random Forest

Parámetro	Valor
Número de árboles	300
Características por división	Raíz cuadrada
Ponderación	Balanceada por submuestra
Procesamiento paralelo	Activado
Semilla	42
Registros de entrenamiento	4.674
Características	19.602

El modelo entrenado fue almacenado como:

models/modelo_random_forest.pkl

Aunque Random Forest puede trabajar con la matriz TF-IDF, su estructura basada en árboles puede requerir más memoria y tiempo de procesamiento que los modelos lineales. Este aspecto será considerado junto con las métricas de categorización durante la evaluación.

3.9.4.7 Entrenamiento de los modelos. Los tres algoritmos fueron entrenados mediante la misma partición y bajo un procedimiento reproducible, como se muestra en la Figura 31.

Figura 31

Entrenamiento de los modelos de categorización automática

	modelo	archivo	registros_entrenamiento	caracteristicas	tiempo_entrenamiento_s
0	Naive Bayes	modelo_naive_bayes.pkl	4674	19602	0.0035
1	Support Vector Machine (SVM)	modelo_svm.pkl	4674	19602	0.0783
2	Random Forest	modelo_random_forest.pkl	4674	19602	3.8602

```
Conjunto de prueba consultado: False
Modelo seleccionado: None
```

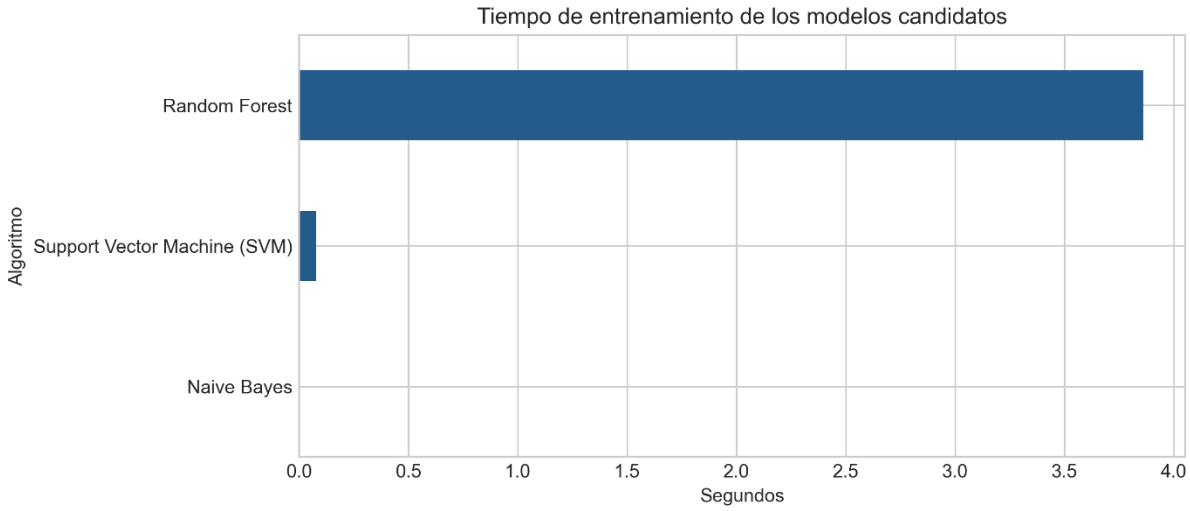
La tabla generada por el notebook muestra el algoritmo, el archivo creado, el número de registros, la cantidad de características y el tiempo requerido para completar el entrenamiento.

El tiempo de entrenamiento constituye una referencia técnica del entorno local y no representa una medida de la calidad del modelo. Un algoritmo más rápido no necesariamente categoriza mejor los tickets.

La Figura 32 presenta el tiempo requerido para entrenar los modelos candidatos en el entorno local utilizado para la investigación.

Figura 32

Tiempo requerido para entrenar los modelos candidatos



3.9.4.8 Artefactos generados. Como resultado de la fase de modelado se generaron tres archivos:

Tabla 28

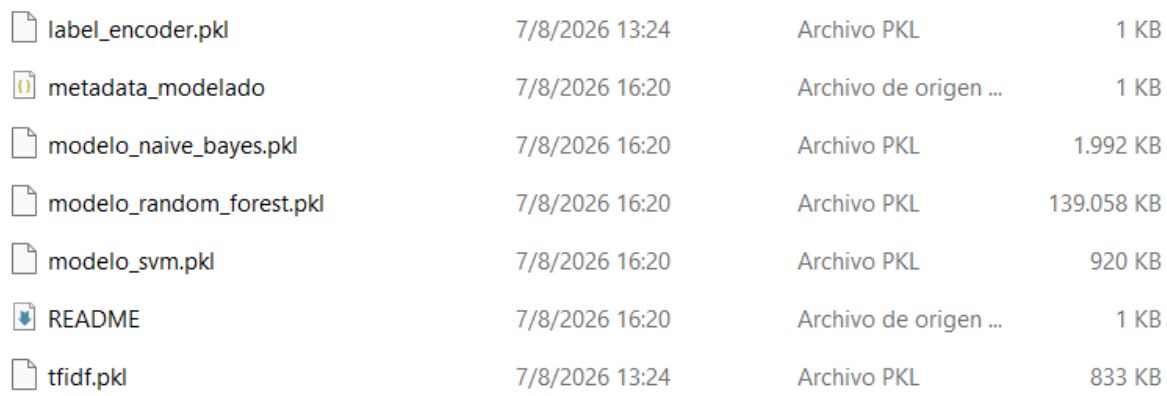
Artefactos Generados

Algoritmo	Archivo
Naive Bayes	modelo_naive_bayes.pkl
Support Vector Machine	modelo_svm.pkl
Random Forest	modelo_random_forest.pkl

La Figura 33 muestra los archivos y modelos almacenados como resultado del proceso de entrenamiento.

Figura 33

Modelos almacenados después del entrenamiento



También se generó el archivo metadata_modelado.json, que registra:

- Algoritmos entrenados.
- Semilla utilizada.
- Cantidad de registros.
- Número de características TF-IDF.
- Fecha de ejecución.
- Confirmación de que el conjunto de prueba no fue consultado.
- Ausencia de un modelo seleccionado.

3.9.4.9 Preparación para la integración con Actybase Web. Los tres modelos fueron desarrollados como componentes independientes del notebook. El módulo entrenamiento.py permite volver a entrenarlos sin copiar manualmente el código.

Una vez que la evaluación determine el algoritmo con mejor desempeño, el modelo seleccionado será almacenado como:

models/modelo_seleccionado.pkl

El módulo prediccion.py —empleado posteriormente por Actybase Web— cargará ese modelo junto con:

- El vectorizador tfidf.pkl.
- El codificador label_encoder.pkl.

- Las funciones de anonimización.
- Las reglas de normalización.
- El tokenizador y lematizador de español.

La finalidad del sistema consiste en categorizar automáticamente los tickets según su tipo de incidencia. La categoría asignada podrá ser revisada o corregida por el personal responsable cuando sea necesario.

3.9.4.10 Resultado de la fase de modelado. La fase de modelado permitió construir tres categorizadores automáticos basados en Naive Bayes, Support Vector Machine y Random Forest.

Los algoritmos fueron entrenados mediante 4.674 tickets y 19.602 características TF-IDF. Todos utilizaron la misma información y el mismo codificador de categorías.

El conjunto de prueba no fue consultado y no se calcularon métricas de evaluación durante esta fase. Tampoco se seleccionó un algoritmo ganador.

Como resultado, se generaron tres modelos candidatos que serán examinados en la siguiente fase mediante accuracy, precisión, recall, F1 y matrices de confusión. Esta evaluación permitirá determinar cuál ofrece el comportamiento más adecuado para su posterior integración con Actybase Web.

3.9.5 Evaluación

La quinta fase de la metodología CRISP-DM corresponde a la evaluación. Su propósito consiste en medir y comparar el comportamiento de los modelos desarrollados antes de seleccionar el algoritmo que continuará hacia la integración con Actybase Web (Chapman et al., 2000).

En esta fase se evaluaron los modelos Naive Bayes, Support Vector Machine y Random Forest mediante el conjunto de prueba construido durante la preparación de los datos. Este conjunto contiene 1.169 tickets que no fueron utilizados durante el entrenamiento.

El proceso se desarrolló mediante el notebook 05_evaluacion.ipynb y el módulo reutilizable evaluacion.py. Los tres algoritmos recibieron la misma matriz TF-IDF y fueron evaluados mediante las mismas etiquetas reales.

La evaluación no volvió a entrenar los modelos ni modificó el conjunto de prueba. Cada algoritmo se limitó a asignar una categoría a los tickets y posteriormente se comparó esta asignación con la categoría registrada históricamente.

3.9.5.1 Conjunto de prueba. El conjunto utilizado para la evaluación contiene 1.169 tickets y 19.602 características TF-IDF.

Sus dimensiones son:

X_test = 1.169 filas × 19.602 características

y_test = 1.169 etiquetas

La partición de prueba mantuvo las mismas categorías utilizadas durante el entrenamiento:

- Soporte Funcional.
- Soporte Software.
- Soporte Operativo.
- Soporte Operativo para cobrar.
- Soporte Comercial.
- Otras Incidencias.

Código:

```
from __future__ import annotations

import json
from datetime import datetime, timezone

import joblib
import matplotlib
matplotlib.use("Agg")
```

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy import sparse
from sklearn.metrics import (
    accuracy_score,
    balanced_accuracy_score,
    classification_report,
    confusion_matrix,
    f1_score,
    precision_score,
    recall_score,
)

from .conexion import ROOT

def cargar_evaluacion():
    """Carga test, codificador y los tres modelos definidos en la tesis."""
    x_test = sparse.load_npz(ROOT / "data" / "processed" / "X_test_tfidf.npz")
    y_test = pd.read_csv(ROOT / "data" / "processed" / "y_test.csv")["clase"].to_numpy()
    codificador = joblib.load(ROOT / "models" / "label_encoder.pkl")
    modelos = {
        "Naive Bayes": joblib.load(ROOT / "models" / "modelo_naive_bayes.pkl"),
        "Support Vector Machine (SVM)": joblib.load(ROOT / "models" / "modelo_svm.pkl"),
        "Random Forest": joblib.load(ROOT / "models" / "modelo_random_forest.pkl"),
    }
    return x_test, y_test, codificador, modelos

def calcular_metricas(y_real, y_estimado) -> dict:
    return {
        "accuracy": accuracy_score(y_real, y_estimado),
        "balanced_accuracy": balanced_accuracy_score(y_real, y_estimado),
                "precision_macro": precision_score(y_real, y_estimado, average="macro",
zero_division=0),
```

```
        "recall_macro": recall_score(y_real, y_estimado, average="macro", zero_division=0),
        "f1_macro": f1_score(y_real, y_estimado, average="macro", zero_division=0),
        "f1_weighted": f1_score(y_real, y_estimado, average="weighted", zero_division=0),
    }

def guardar_matriz(nombre: str, matriz: np.ndarray, clases: list[str]) -> None:
    figuras = ROOT / "outputs" / "figuras"
    figuras.mkdir(parents=True, exist_ok=True)
    fig, ax = plt.subplots(figsize=(9, 7))
    imagen = ax.imshow(matriz, cmap="Blues")
    fig.colorbar(imagen, ax=ax, fraction=0.046, pad=0.04)
    ax.set_xticks(range(len(clases)), clases, rotation=45, ha="right")
    ax.set_yticks(range(len(clases)), clases)
    ax.set_xlabel("Categoría asignada")
    ax.set_ylabel("Categoría real")
    ax.set_title(f"Matriz de confusión - {nombre}")
    umbral = matriz.max() / 2 if matriz.size else 0
    for i in range(matriz.shape[0]):
        for j in range(matriz.shape[1]):
            ax.text(j, i, int(matriz[i, j]), ha="center", va="center",
                    color="white" if matriz[i, j] > umbral else "black", fontsize=8)
    fig.tight_layout()
    seguro = {"Naive Bayes": "naive_bayes", "Support Vector Machine (SVM)": "svm",
"Random Forest": "random_forest"}[nombre]
    fig.savefig(figuras / f"matriz_confusion_{seguro}.png", dpi=300, bbox_inches="tight")
    plt.close(fig)

def evaluar_modelos() -> tuple[pd.DataFrame, str, dict]:
    x_test, y_test, codificador, modelos = cargar_evaluacion()
    metricas_dir = ROOT / "outputs" / "metricas"
    metricas_dir.mkdir(parents=True, exist_ok=True)
    clases = list(codificador.classes_)
    etiquetas = np.arange(len(clases))
    filas, predicciones = [], {}
```

```
    for nombre, modelo in modelos.items():
        y_estimado = modelo.predict(x_test)
        predicciones[nombre] = y_estimado
        filas.append({"modelo": nombre, **calcular_metricas(y_test, y_estimado)})
        reporte = classification_report(
            y_test, y_estimado, labels=etiquetas, target_names=clases,
            output_dict=True, zero_division=0,
        )
        pd.DataFrame(reporte).transpose().to_csv(
                metricas_dir / f"reporte_{nombre.lower().replace(' ', '_').replace('(', '').replace(')',
'')}.csv",
            encoding="utf-8-sig",
        )
        guardar_matriz(nombre, confusion_matrix(y_test, y_estimado, labels=etiquetas), clases)

    comparacion = pd.DataFrame(filas).sort_values(
        ["f1_macro", "balanced_accuracy", "f1_weighted"], ascending=False
    ).reset_index(drop=True)
        comparacion.to_csv(metricas_dir / "comparacion_modelos_test.csv", index=False,
encoding="utf-8-sig")

    ganador = str(comparacion.iloc[0]["modelo"])
    joblib.dump(modelos[ganador], ROOT / "models" / "modelo_seleccionado.pkl")
    metadata = {
        "modelo_seleccionado": ganador,
        "criterio_principal": "Mayor F1 macro en el conjunto de prueba",
        "criterios_secundarios": ["balanced_accuracy", "f1_weighted"],
        "registros_prueba": int(x_test.shape[0]),
        "caracteristicas": int(x_test.shape[1]),
        "fecha_utc": datetime.now(timezone.utc).isoformat(),
        "metricas": comparacion.iloc[0].drop("modelo").to_dict(),
    }
    (ROOT / "models" / "metadata_evaluacion.json").write_text(
```

```
        json.dumps(metadata, ensure_ascii=False, indent=2), encoding="utf-8"
    )

    # Auditoría sin texto ni identificadores personales.
    final = pd.read_csv(ROOT / "data" / "processed" / "test.csv", low_memory=False)
    auditoria = final[["ticket_uid", "categoria"]].copy()
    auditoria["categoria_asignada"] = codificador.inverse_transform(predicciones[ganador])
    auditoria["correcto"] = auditoria["categoria"] == auditoria["categoria_asignada"]
        auditoria.to_csv(metricas_dir / "auditoria_categorizacion_test.csv", index=False,
encoding="utf-8-sig")
    return comparacion, ganador, metadata

if __name__ == "__main__":
    tabla, ganador, metadata = evaluar_modelos()
    print(tabla.to_string(index=False))
    print("Modelo seleccionado:", ganador)
    print(json.dumps(metadata, ensure_ascii=False, indent=2))
```

La Figura 34 muestra la carga del conjunto independiente utilizado para evaluar los modelos entrenados.

Figura 34

Carga del conjunto independiente utilizado para evaluar los modelos

```
Forma de X_test: (1169, 19602)
Forma de y_test: (1169,)
Modelos evaluados: ['Naive Bayes', 'Support Vector Machine (SVM)', 'Random Forest']
```

Modelos evaluados:

- Naive Bayes
- Support Vector Machine (SVM)
- Random Forest

3.9.5.2 Métricas utilizadas. La comparación se realizó mediante diferentes métricas, debido a que la distribución de los tipos de incidencia presenta un desbalance considerable.

Las métricas utilizadas fueron:

- Accuracy.
- Exactitud balanceada.
- Precisión macro.
- Recall macro.
- F1 macro.
- F1 ponderado.
- Matriz de confusión.

El accuracy representa la proporción total de tickets categorizados correctamente:

Accuracy = categorizaciones correctas / total de tickets

Aunque esta métrica proporciona una visión general, puede resultar insuficiente en conjuntos desbalanceados. Un modelo puede alcanzar un accuracy relativamente elevado categorizando correctamente las clases mayoritarias y presentar un comportamiento deficiente sobre las minoritarias.

La precisión mide qué proporción de los tickets asignados a una categoría pertenece realmente a esa categoría.

El recall mide qué proporción de los tickets reales de una categoría fue identificada correctamente.

F1 integra precisión y recall a través de su media armónica. Esta métrica resulta útil cuando se persigue una armonización entre las dos partes.

F1 macro calcula F1 por separado para cada categoría y posteriormente obtiene un promedio asignando el mismo peso a todas las clases. Por esta razón, fue establecida como criterio principal para seleccionar el modelo.

F1 ponderado también calcula F1 por categoría, pero asigna un peso proporcional al número de observaciones existentes en cada clase.

La exactitud balanceada obtiene el promedio del recall de las categorías. Su utilización permite reducir la influencia de las clases mayoritarias.

3.9.5.3 Comparación general de los modelos. Los resultados obtenidos sobre los 1.169 tickets de prueba se presentan en la Tabla 29.

Tabla 29

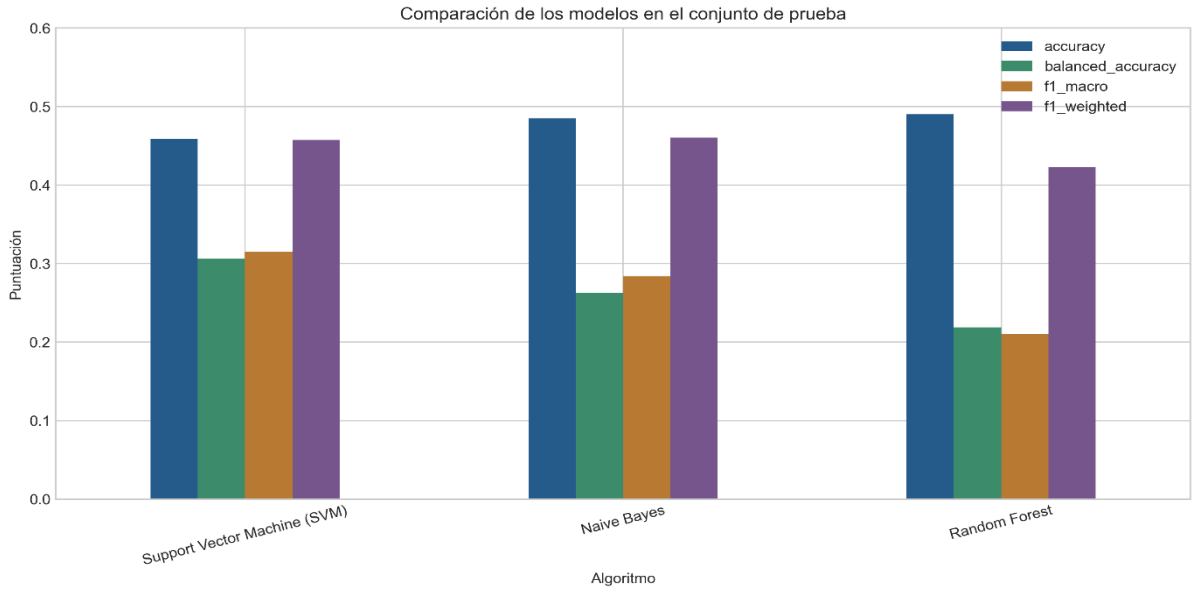
Comparación de los modelos sobre el conjunto de prueba

Modelo	Accuracy	Exactitud balanceada	Precisión macro	Recall macro	F1 macro	F1 ponderado
Support Vector Machine	0,459	0,306	0,326	0,306	0,315	0,457
Naive Bayes	0,485	0,263	0,402	0,263	0,284	0,460
Random Forest	0,490	0,219	0,426	0,219	0,210	0,423

La Figura 35 presenta la comparación del comportamiento de los modelos evaluados sobre el conjunto de prueba.

Figura 35

Comparación del comportamiento de los modelos sobre el conjunto de prueba



Random Forest obtuvo el mayor accuracy, con un valor de 0,490. Naive Bayes alcanzó 0,485 y SVM obtuvo 0,459.

Sin embargo, la comparación cambia al considerar el desempeño equilibrado entre las categorías. SVM obtuvo la mayor exactitud balanceada, con 0,306, y el mayor F1 macro, con 0,315.

Naive Bayes alcanzó un F1 macro de 0,284, mientras que Random Forest obtuvo 0,210. Esto indica que Random Forest concentró una mayor proporción de sus aciertos en las clases frecuentes y presentó mayores dificultades para reconocer las categorías minoritarias.

Por este motivo, el modelo no fue seleccionado únicamente por su accuracy. La naturaleza desbalanceada del conjunto exige considerar el rendimiento de cada categoría.

3.9.5.4 Resultados de Support Vector Machine. Support Vector Machine fue el algoritmo con mayor F1 macro y exactitud balanceada. Sus resultados por categoría se presentan en la Tabla 30.

Tabla 30

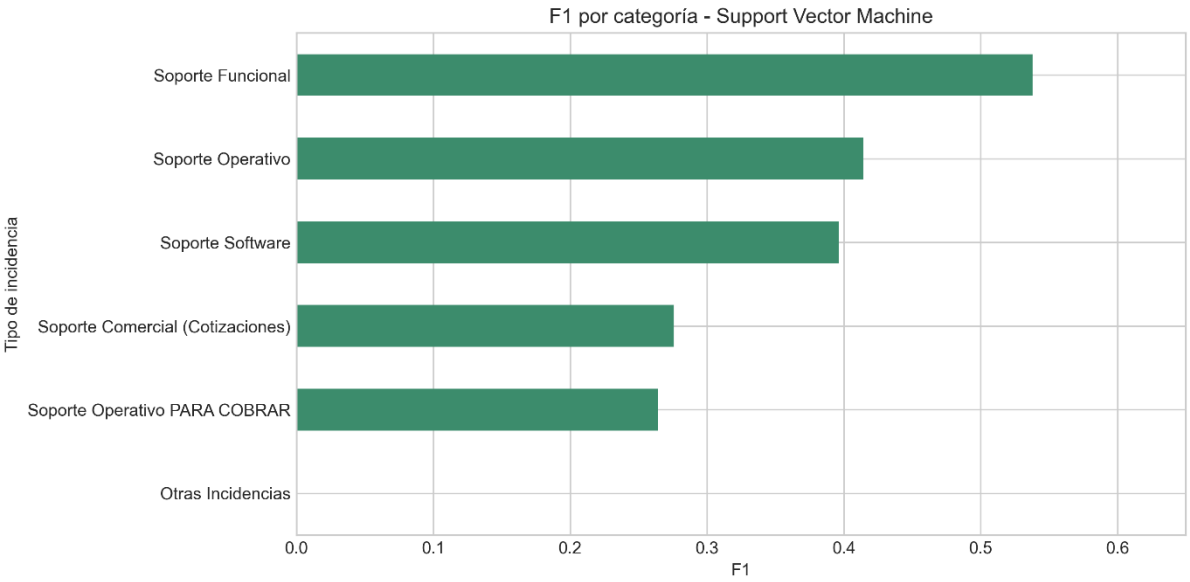
Métricas por categoría de Support Vector Machine

Tipo de incidencia	Precisión	Recall	F1	Tickets
Otras Incidencias	0,000	0,000	0,000	1
Soporte Comercial	0,333	0,235	0,276	17
Soporte Funcional	0,533	0,543	0,538	510
Soporte Operativo	0,416	0,412	0,414	296
Soporte Operativo para cobrar	0,280	0,250	0,264	28
Soporte Software	0,395	0,397	0,396	317

La Figura 36 presenta la medida F1 obtenida por categoría para el modelo Support Vector Machine.

Figura 36

Medida F1 por categoría obtenida mediante Support Vector Machine



La categoría con mejor resultado fue Soporte Funcional, con un F1 de 0,538. Posteriormente se ubicaron Soporte Operativo, con 0,414, y Soporte Software, con 0,396.

Las categorías minoritarias presentaron resultados inferiores. Soporte Comercial alcanzó un F1 de 0,276 y Soporte Operativo para cobrar obtuvo 0,264.

Otras Incidencias no fue reconocida correctamente. Esta categoría contiene únicamente dos observaciones en entrenamiento y una en prueba. La cantidad disponible resulta insuficiente para que el modelo aprenda un patrón generalizable.

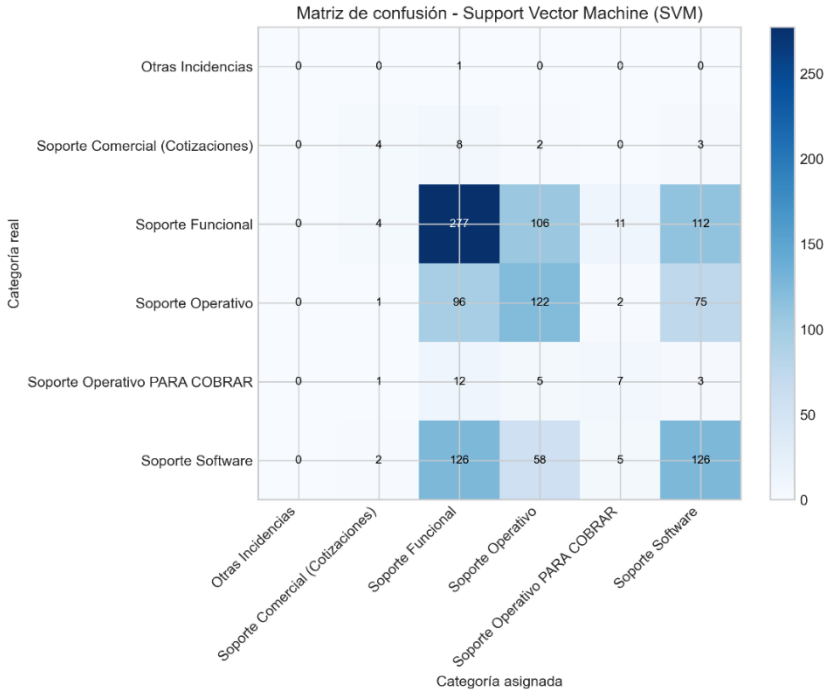
Los hallazgos evidencian que la calidad y la cantidad de las etiquetas históricas influyen directamente sobre la capacidad de categorización.

3.9.5.5 Matriz de confusión. La matriz de confusión permite observar las categorías asignadas correctamente y las confusiones producidas por el modelo.

La Figura 37 presenta la matriz de confusión obtenida para el modelo Support Vector Machine sobre el conjunto de prueba.

Figura 37

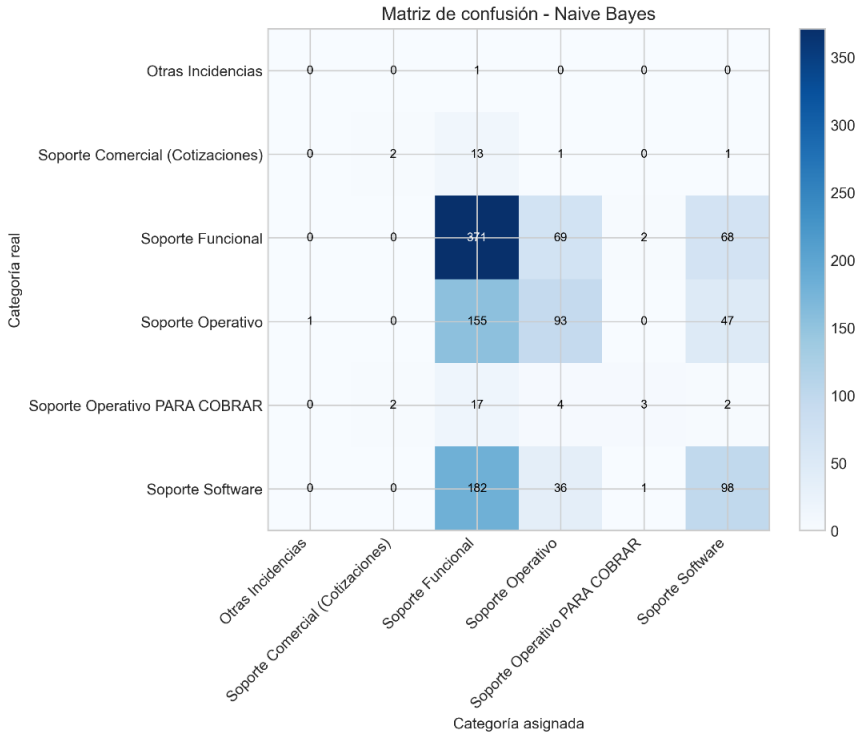
Matriz de confusión de Support Vector Machine



La Figura 38 presenta la matriz de confusión obtenida para el modelo Naive Bayes sobre el conjunto de prueba.

Figura 38

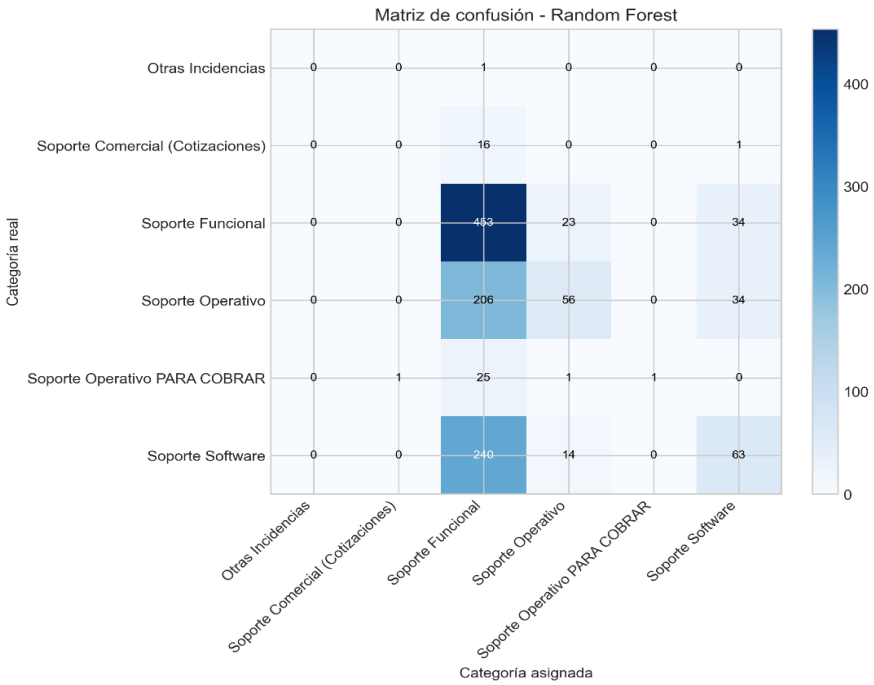
Matriz de confusión Naive Bayes



La Figura 39 presenta la matriz de confusión obtenida para el modelo Random Forest sobre el conjunto de prueba.

Figura 39

Matriz de confusión Random Forest



Las filas indican las categorías reales y las columnas, las asignadas por el modelo. Los valores en la diagonal principal representan categorizaciones correctas.

SVM categorizó correctamente 536 de los 1.169 tickets. Los 633 casos restantes fueron asignados a una categoría diferente de la etiqueta histórica.

Los resultados correctos por categoría fueron:

- 277 tickets de Soporte Funcional.
- 126 tickets de Soporte Software.
- 122 tickets de Soporte Operativo.
- 7 tickets de Soporte Operativo para cobrar.
- 4 tickets de Soporte Comercial.
- 0 tickets de Otras Incidencias.

Las principales confusiones se produjeron entre las tres categorías mayoritarias:

- 126 tickets de Soporte Software fueron asignados como Soporte Funcional.
- 112 tickets de Soporte Funcional fueron asignados como Soporte Software.
- 106 tickets de Soporte Funcional fueron asignados como Soporte Operativo.
- 96 tickets de Soporte Operativo fueron asignados como Soporte Funcional.
- 75 tickets de Soporte Operativo fueron asignados como Soporte Software.

Estas confusiones indican que los textos de diferentes categorías comparten vocabulario y expresiones similares.

3.9.5.6 Análisis de las causas de error. El comportamiento observado puede estar relacionado con diferentes factores.

En primer lugar, las categorías Soporte Funcional, Soporte Operativo y Soporte Software pueden compartir palabras como sistema, proceso, error, reporte, módulo, transacción o comprobante. La presencia de estos términos no siempre permite determinar por sí sola el tipo de incidencia.

En segundo lugar, algunos tickets pueden contener asuntos breves o descripciones insuficientes. Cuando el usuario no explica claramente el problema, el modelo dispone de poca información para asignar una categoría.

En tercer lugar, el etiquetado histórico puede contener diferencias de criterio. Problemas similares podrían haber sido clasificados de forma distinta por diferentes técnicos.

En cuarto lugar, las categorías minoritarias no disponen de suficientes ejemplos. Soporte Comercial, Soporte Operativo para cobrar y Otras Incidencias poseen una cantidad considerablemente inferior de registros.

Finalmente, TF-IDF representa la importancia de las palabras, pero no comprende completamente el contexto semántico ni las relaciones complejas entre las expresiones.

Estos factores explican por qué el modelo reconoce mejor las categorías con mayor cantidad de ejemplos y presenta dificultades con casos ambiguos o poco representados.

3.9.5.7 Selección del mejor modelo. Support Vector Machine fue seleccionado como el modelo más adecuado entre los tres algoritmos evaluados.

La selección se fundamentó en las siguientes razones:

- Obtuvo el mayor F1 macro, con 0,315.
- Alcanzó la mayor exactitud balanceada, con 0,306.
- Presentó el mejor recall macro.
- Mostró un desempeño más equilibrado entre las categorías.
- Es apropiado para matrices TF-IDF dispersas y de alta dimensión.
- Requiere menos recursos de almacenamiento que Random Forest.
- Superó a Naive Bayes y Random Forest en la métrica principal.

Aunque Random Forest alcanzó el mayor accuracy, su menor F1 macro indica que sus resultados se concentraron principalmente en las clases más frecuentes.

Naive Bayes presentó resultados intermedios. Su precisión macro fue superior, pero su recall macro y F1 macro fueron inferiores a los de SVM.

El modelo SVM seleccionado fue almacenado como:

models/modelo_seleccionado.pkl

3.9.5.8 Ejemplos sintéticos de categorización. Después de seleccionar el modelo se realizaron pruebas con textos sintéticos. Estos ejemplos fueron creados exclusivamente para demostrar el funcionamiento técnico y no pertenecen a usuarios reales.

Tabla 31

Ejemplos sintéticos procesados por el modelo seleccionado

Asunto sintético	Categoría asignada
Error al guardar comprobante	Soporte Software
Solicitud de cotización	Soporte Comercial

Asunto sintético	Categoría asignada
No cuadra saldo	Soporte Operativo
Cambio de parámetro	Soporte Funcional
Proceso para cobrar	Soporte Funcional

La Figura 40 presenta ejemplos sintéticos utilizados para verificar el funcionamiento de la categorización automática mediante el modelo SVM.

Figura 40

Ejemplos de categorización automática mediante el modelo SVM

Los siguientes textos fueron creados únicamente para demostrar el funcionamiento técnico. No pertenecen a usuarios reales y no forman parte de las métricas.

```
pipeline = PipelineInferencia()
ejemplos = [
    ('Error al guardar comprobante', 'El sistema muestra un error al intentar guardar el comprobante'),
    ('Solicitud de cotización', 'Por favor enviar cotización para desarrollar el nuevo reporte'),
    ('No cuadra saldo', 'El saldo contable no coincide al cerrar el día'),
    ('Cambio de parámetro', 'Necesitamos configurar el porcentaje de interés'),
    ('Proceso para cobrar', 'Se realizó soporte operativo que debe facturarse al cliente'),
]
resultados = []
for asunto, descripcion in ejemplos:
    resultados.append({'asunto_sintetico': asunto, 'categoria_asignada': pipeline.predecir_categoria(asunto, descripcion)})
display(pd.DataFrame(resultados))
```

	asunto_sintetico	categoria_asignada
0	Error al guardar comprobante	Soporte Software
1	Solicitud de cotización	Soporte Comercial (Cotizaciones)
2	No cuadra saldo	Soporte Operativo
3	Cambio de parámetro	Soporte Operativo
4	Proceso para cobrar	Soporte Funcional

Los ejemplos no forman parte de las métricas de evaluación y no deben interpretarse como evidencia de exactitud.

El caso “Proceso para cobrar” fue asignado como Soporte Funcional, a pesar de contener una referencia a un procedimiento cobrable. Este resultado ilustra que el modelo todavía puede cometer errores y que la categoría asignada debe mantenerse editable.

3.9.5.9 Limitaciones del modelo seleccionado. Aunque SVM obtuvo el mejor equilibrio entre los algoritmos comparados, su accuracy de 0,459 y su F1 macro de 0,315 muestran que el modelo todavía posee oportunidades importantes de mejora.

Las principales limitaciones son:

- Cantidad reducida de tickets etiquetados.
- Desbalance extremo entre categorías.
- Tres registros disponibles para Otras Incidencias.
- Posibles inconsistencias del etiquetado histórico.
- Vocabulario compartido entre diferentes tipos de incidencia.
- Descripciones breves o ambiguas.
- Utilización exclusiva de TF-IDF para representar el texto.
- Ausencia inicial de variables complementarias validadas.

Por tanto, el modelo no debe utilizarse todavía para reemplazar completamente la decisión humana.

Su aplicación más apropiada consiste en presentar una categoría sugerida que pueda ser confirmada o corregida. Las correcciones generadas por los usuarios podrán almacenarse y utilizarse posteriormente para ampliar el conjunto de entrenamiento.

3.9.5.10 Resultado de la fase de evaluación. La fase de evaluación permitió comparar Naive Bayes, Support Vector Machine y Random Forest mediante 1.169 tickets independientes.

Random Forest obtuvo el mayor accuracy, pero Support Vector Machine alcanzó el mejor F1 macro y la mayor exactitud balanceada. Por esta razón, SVM fue seleccionado como el modelo más adecuado para continuar hacia la integración experimental con Actybase Web.

La evaluación también permitió identificar que las principales dificultades se concentran en las categorías minoritarias y en las confusiones entre Soporte Funcional, Soporte Operativo y Soporte Software.

El modelo seleccionado fue almacenado de manera independiente junto con el vectorizador TF-IDF, el codificador de etiquetas y las funciones de procesamiento textual.

La siguiente fase estará orientada al despliegue experimental, donde se construirá el flujo que recibe el asunto y la descripción de un ticket, procesa el texto y presenta una categoría sugerida dentro del entorno de Actybase Web.

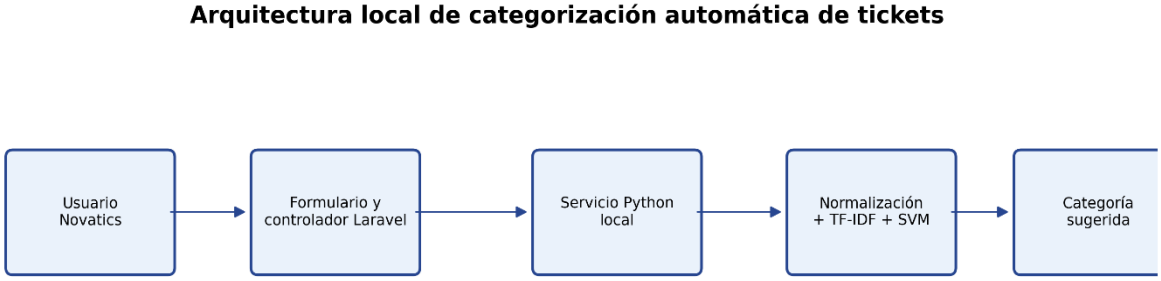
3.9.6 Despliegue

La fase de despliegue permitió incorporar el modelo de categorización automática de tickets en el entorno local del sistema Actybase Web. Para esta integración se implementó una arquitectura desacoplada compuesta por la aplicación web desarrollada en Laravel y un servicio local de inferencia desarrollado en Python. Este servicio carga los artefactos generados durante las fases de preparación, modelado y evaluación: el vectorizador TF-IDF, el codificador de la variable objetivo y el modelo Support Vector Machine (SVM) seleccionado.

3.9.6.1 Arquitectura de la solución. El flujo implementado inicia cuando el usuario accede al módulo de Analítica de Datos. En esta pantalla se ingresan el asunto y la descripción del requerimiento. Actybase Web valida los campos y envía la solicitud al servicio Python mediante una comunicación HTTP local. A continuación, el servicio aplica el mismo procedimiento de anonimización, normalización, lematización y vectorización utilizado durante el entrenamiento. Finalmente, el modelo SVM procesa la representación TF-IDF y devuelve la categoría de incidencia sugerida. El flujo de esta arquitectura se presenta en la Figura 41.

Figura 41

Arquitectura de integración entre Actybase Web y el servicio local de categorización

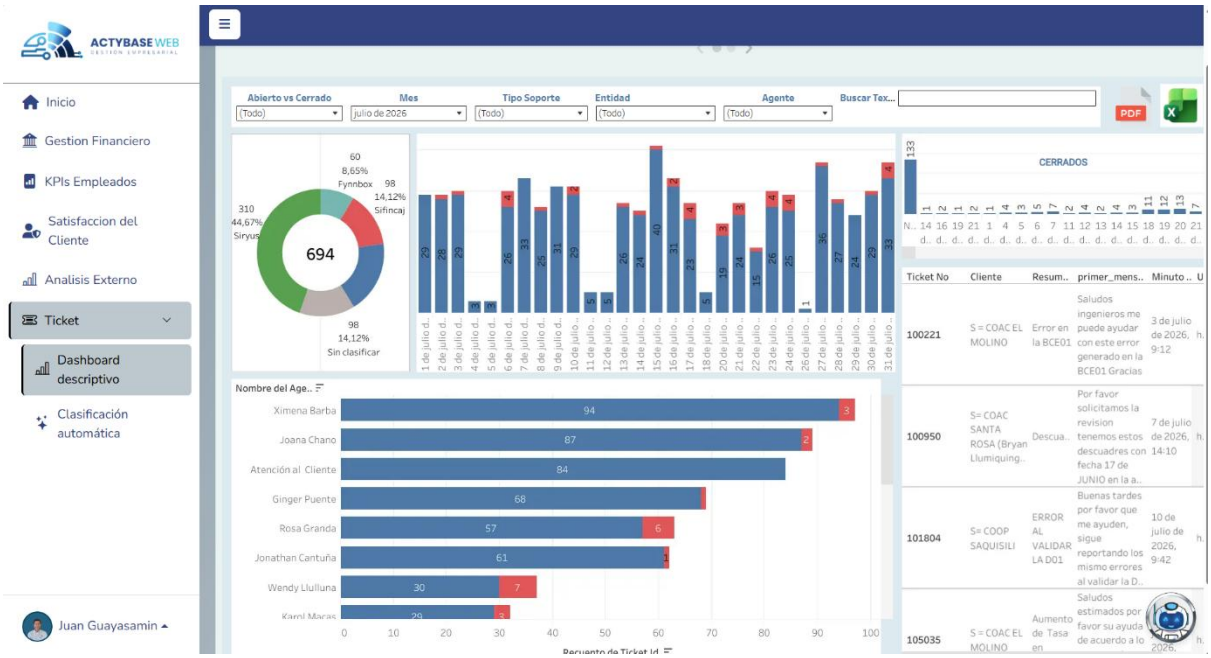


3.9.6.2 Integración con Actybase Web. Se modificó únicamente el módulo Analítica de Datos, manteniendo sin cambios las funcionalidades preexistentes. La opción Ticket se organizó como un submenú que contiene: a) Dashboard descriptivo, correspondiente a la integración existente con Tableau; y b) Clasificación automática, correspondiente al nuevo componente de PLN y aprendizaje automático. Esta separación permite conservar el análisis descriptivo y, simultáneamente, incorporar la categorización automática como una función independiente.

La Figura 42 presenta el submenú Ticket del módulo Analítica de Datos de Actybase Web, desde el cual se accede al dashboard descriptivo y a la funcionalidad de clasificación automática.

Figura 42

Submenú Ticket del módulo Analítica de Datos



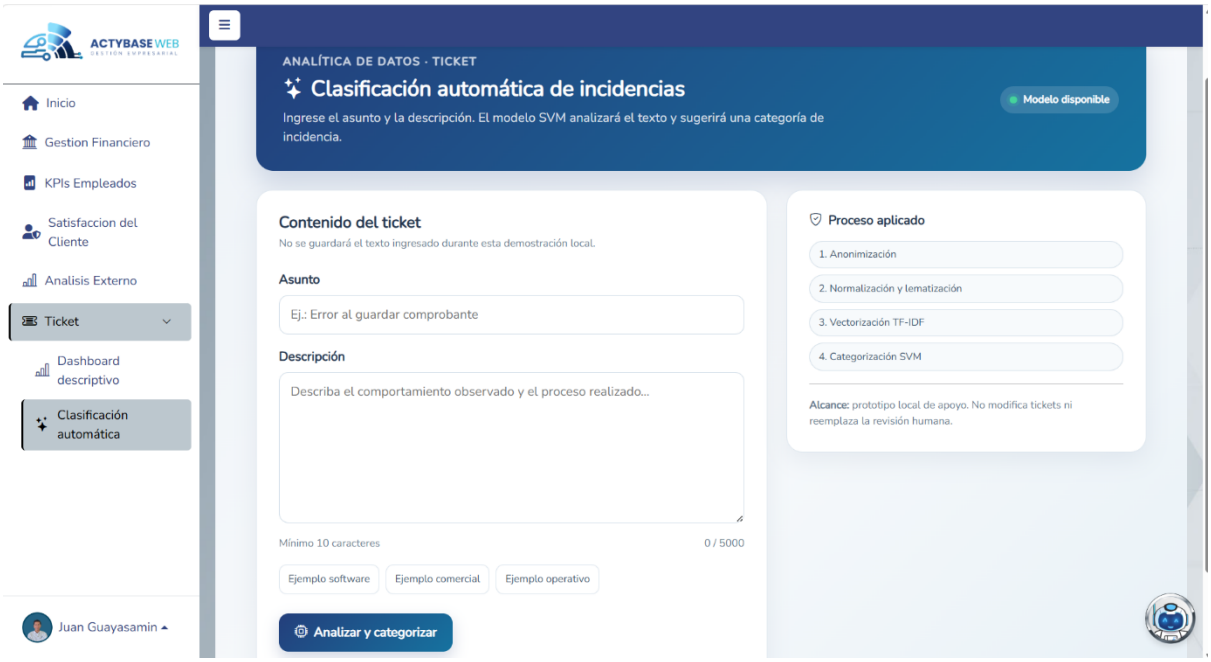
El formulario de clasificación presenta campos para el asunto y la descripción del ticket, el estado del servicio y ejemplos sintéticos de prueba. La interfaz no registra

automáticamente el ticket ni sustituye la decisión del personal responsable; presenta una recomendación que debe ser revisada y confirmada.

La Figura 43 presenta la interfaz implementada para la categorización automática de tickets.

Figura 43

Interfaz para la categorización automática de tickets



3.9.6.3 Servicio de inferencia. El servicio de inferencia fue restringido a la dirección local 127.0.0.1. Dispone de un endpoint de verificación de estado y otro para clasificar el texto. La solicitud incluye exclusivamente el asunto y la descripción requeridos para la demostración. Para preservar la privacidad, el servicio no almacena el contenido recibido ni genera registros con datos del ticket.

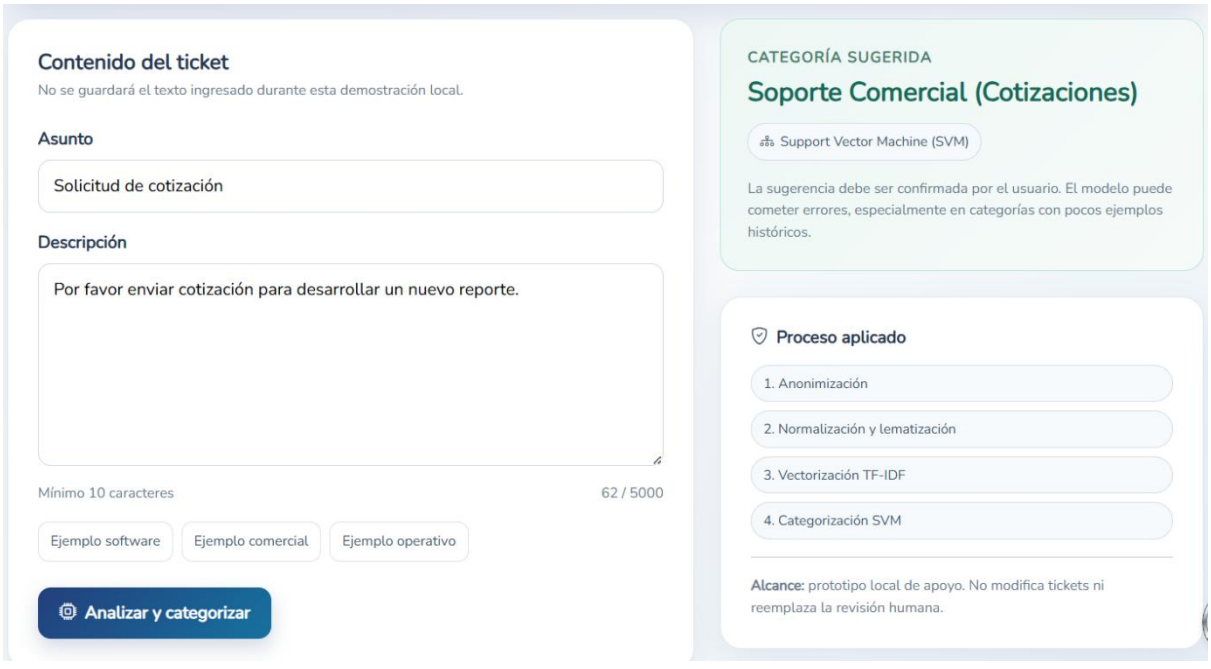
El proceso de inferencia conserva la secuencia metodológica utilizada en el entrenamiento: unión del asunto y la descripción, anonimización de datos sensibles, normalización del texto, lematización, transformación mediante TF-IDF y aplicación del clasificador SVM. De esta manera se evita una divergencia entre el procesamiento empleado en la evaluación y el utilizado durante el despliegue.

3.9.6.4 Validación funcional. La validación se realizó en el entorno local mediante tickets sintéticos. Primero se comprobó la disponibilidad del servicio y la carga del modelo SVM. Posteriormente, se ingresó un asunto y una descripción de prueba desde Actybase Web y se verificó que el sistema devolviera una categoría de incidencia. La prueba técnica del servicio produjo el estado “ok” y el flujo integrado respondió correctamente. Además, el acceso directo sin una sesión activa fue rechazado por los controles de autenticación del sistema.

La Figura 44 presenta el resultado de la categorización automática aplicada a un ticket sintético.

Figura 44

Resultado de la categorización automática de un ticket sintético



3.9.6.5 Controles y consideraciones operativas. La solución incluye validación de los campos, protección mediante la autenticación existente en Actybase Web, control de errores cuando el servicio no se encuentra disponible y limitación del tamaño de las solicitudes. Asimismo, el servicio se ejecuta únicamente en el equipo local y no expone el modelo directamente a Internet. Los ejemplos de la demostración son sintéticos y no contienen información identificable de clientes.

Debido a que en la evaluación el modelo seleccionado obtuvo una exactitud de 45,9 %, un F1 macro de 0,315 y una exactitud balanceada de 0,306, la categoría se presenta como una sugerencia sujeta a supervisión humana. Esta decisión reduce el riesgo de que una clasificación incorrecta se aplique de manera irreversible y permite recopilar correcciones para futuras iteraciones del modelo.

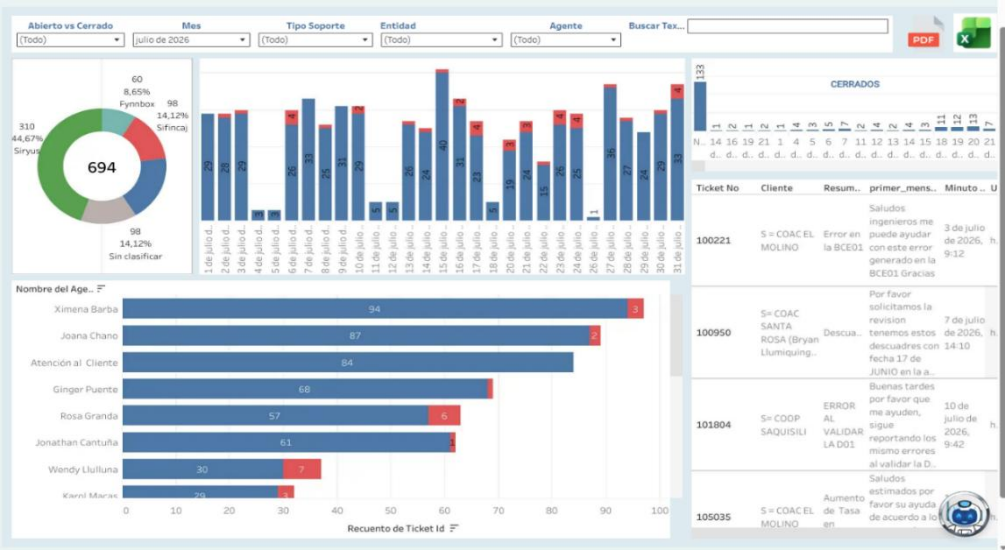
3.9.6.6 Resultado de la fase. El despliegue local demostró la viabilidad técnica de integrar un modelo de PLN con el sistema Actybase Web. La aplicación también muestra los indicadores descriptivos de Tableau. Una vez realizada una categorización adecuada, estos indicadores reflejarán de mejor manera la realidad del negocio con respecto a los tickets de soporte técnico. Además, la aplicación incorpora una interfaz independiente para categorizar tickets.

La arquitectura propuesta facilita la actualización posterior del modelo sin modificar la lógica principal del sistema. Como trabajo de continuidad se recomienda registrar, con autorización y controles de privacidad, la categoría sugerida y la categoría finalmente confirmada para monitorear el desempeño, detectar cambios en la distribución de datos y planificar nuevos entrenamientos.

La Figura 45 presenta el análisis descriptivo de los tickets mediante el dashboard implementado en Webcoopec Ticket.

Figura 45

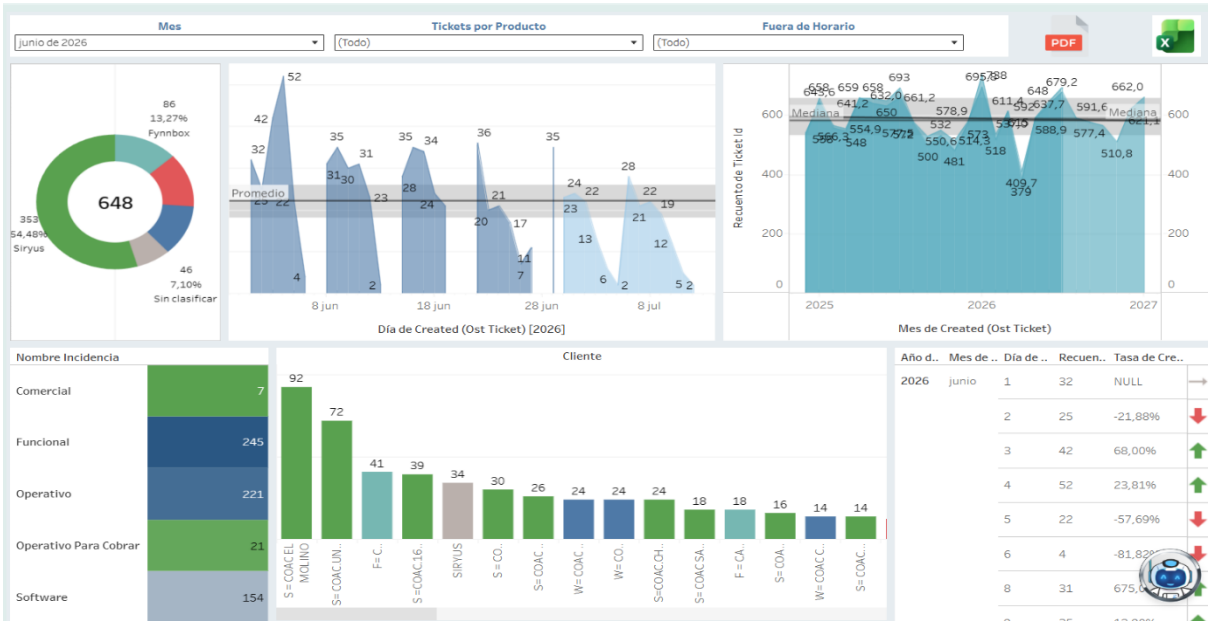
Análisis descriptivo Webcoopec Ticket



La Figura 46 presenta una vista complementaria del análisis descriptivo de los tickets, con indicadores relacionados con producto, tipo de incidencia, clientes y comportamiento temporal.

Figura 46

Vista complementaria del análisis descriptivo de Webcoopec Ticket



Código:

1. **Rutas GET y POST de `tickets/clasificacion-automatica`.**

```
Route::middleware(['auth', 'checktime', 'password_expired'])
    ->group(function () {
        Route::get(
            'tickets/clasificacion-automatica',
            [ClasificacionTicketController::class, 'index']
        )->name('analisis.clasificacionTicket');

        Route::post(
            'tickets/clasificacion-automatica',
            [ClasificacionTicketController::class, 'clasificar']
        )->name('analisis.clasificacionTicket.procesar');
    });
```

La ruta GET presenta la interfaz de clasificación automática, mientras que la ruta POST recibe el asunto y la descripción del ticket para procesarlos. Ambas rutas utilizan los

mecanismos de autenticación, control de sesión y vigencia de contraseña existentes en Novatics.

2. **Llamada HTTP del controlador Laravel al endpoint `/clasificar`.**

```
$datos = $request->validate([
    'asunto' => ['required', 'string', 'min:3', 'max:255'],
    'descripcion' => ['required', 'string', 'min:10', 'max:5000'],
]);

$respuesta = Http::acceptJson()
    ->timeout(config('services.ticket_ml.timeout', 60))
    ->post(
        config('services.ticket_ml.url') . '/clasificar',
        $datos
    );

if (!$respuesta->successful()) {
    return back()->withInput()->withErrors([
        'servicio' => 'El servicio de categorización no pudo procesar el ticket.',
    ]);
}

$resultado = $respuesta->json();
```

El controlador valida la longitud y el tipo de los campos antes de transmitirlos al servicio local de Machine Learning. La comunicación se realiza mediante una solicitud HTTP POST en formato JSON al endpoint /clasificar. Finalmente, Laravel comprueba que la respuesta sea satisfactoria y obtiene la categoría sugerida.

3. **Servicio Python que transforman el texto con TF-IDF y ejecutan `model.predict`.**

```
def categorize(self, subject: str, description: str) -> dict:
    # Aplicación del mismo preprocesamiento utilizado
    # durante el entrenamiento del modelo.
    processed_text = self._process(subject, description)

    # Transformación del texto mediante el vectorizador
    # TF-IDF previamente entrenado.
    feature_matrix = self.vectorizer.transform([processed_text])

    # Categorización automática mediante el modelo SVM.
    encoded_category = int(
        self.model.predict(feature_matrix)[0]
    )

    # Conversión de la etiqueta numérica a su nombre original.
    predicted_category = str(
        self.encoder.inverse_transform(
            [encoded_category]
        )[0]
    )

    return {
        "categoria": predicted_category,
        "modelo": "Support Vector Machine (SVM)",
        "requiere_confirmacion": True,
    }
```

El servicio aplica al ticket el mismo procesamiento lingüístico utilizado durante el entrenamiento. Posteriormente, el texto se transforma en una representación numérica mediante TF-IDF y se entrega al modelo SVM para obtener la categoría de incidencia. El

codificador convierte el resultado numérico en su etiqueta original y la respuesta indica que la sugerencia requiere confirmación humana.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

La investigación llevada a cabo evidenció que el uso de técnicas de Procesamiento de Lenguaje Natural (PLN) junto con Machine Learning representa una opción factible para la automatización en la clasificación de tickets de soporte técnico dentro de la mesa de ayuda de Webcoopec System. A partir del análisis de información histórica proveniente de la plataforma Webcoopec Ticket, fue posible desarrollar un modelo de clasificación supervisada orientado a predecir automáticamente la categoría funcional de nuevos requerimientos, contribuyendo a optimizar el proceso de categorización y fortaleciendo la gestión operativa de la organización.

Con respecto al primer objetivo específico, se llevó a cabo un análisis exploratorio de los datos históricos contenidos en la base de datos del sistema de gestión de tickets, identificando las principales variables utilizadas durante la operación de la mesa de ayuda, así como la distribución de categorías, estados, prioridades, clientes, agentes y demás características relevantes del conjunto de datos. Este proceso permitió detectar registros incompletos, categorías obsoletas, valores inconsistentes y oportunidades de mejora en la calidad de la información, proporcionando una base confiable para el desarrollo del modelo predictivo.

Respecto al segundo objetivo específico, se diseñó e implementó un proceso de preparación de datos fundamentado en técnicas de Procesamiento de Lenguaje Natural. Dicho proceso incluyó la integración de datos provenientes de diferentes tablas de la base de datos, la limpieza y normalización del contenido textual, la eliminación de información irrelevante, el tratamiento de valores nulos, la tokenización, la lematización y la representación vectorial mediante TF-IDF. Estas actividades permitieron transformar información textual no

estructurada en variables numéricas aptas para el entrenamiento de algoritmos de aprendizaje supervisado.

En cumplimiento del tercer objetivo específico, se entrenaron y compararon diferentes algoritmos de Machine Learning para establecer cuál mostró el mejor rendimiento en la clasificación automática de tickets. La evaluación se realizó mediante métricas ampliamente aceptadas en problemas de clasificación supervisada, tales como Accuracy, Precision, Recall, F1-Score y Matriz de Confusión, permitiendo seleccionar el modelo con mayor capacidad de generalización para el contexto operativo de Webcoopec System.

En relación con el cuarto objetivo específico, se desarrolló un prototipo funcional que demuestra la factibilidad de integrar el modelo de clasificación automática dentro de la plataforma Webcoopec Ticket. La incorporación del modelo permite asistir el proceso de categorización de nuevas incidencias, disminuyendo la dependencia de la clasificación manual, promoviendo una mayor estandarización de criterios y reduciendo la probabilidad de errores derivados de la intervención humana.

Uno de los aportes fundamentales de esta investigación radica en evidenciar que la clasificación automática representa únicamente la primera etapa de un proceso integral de analítica de datos. Una vez que los tickets son categorizados de manera consistente, la información puede ser utilizada para construir indicadores descriptivos relacionados con clientes, agentes de soporte, categorías de atención, tiempos de respuesta, productividad, tickets sujetos a facturación, distribución de la carga operativa y comportamiento histórico de la mesa de ayuda. Esto permite transformar los registros históricos en información valiosa para respaldar decisiones estratégicas y mejorar continuamente los procesos organizacionales.

Finalmente, la investigación confirma que la metodología CRISP-DM constituye un marco adecuado para el desarrollo de proyectos de Ciencia de Datos aplicados a entornos empresariales, ya que permitió estructurar de forma ordenada las fases de comprensión del

negocio, comprensión de los datos, preparación de la información, modelado, evaluación y despliegue del modelo predictivo. La aplicación de esta metodología garantizó la trazabilidad entre el problema identificado, los objetivos planteados y los resultados obtenidos, consolidando una propuesta tecnológica que combina Inteligencia Artificial, Procesamiento de Lenguaje Natural y Analítica de Datos para contribuir a la optimización de la gestión operativa de Webcoopec System.

4.2 Limitaciones

Durante el transcurso de esta investigación, se identificaron múltiples limitaciones que deben ser tenidas en cuenta al interpretar los resultados obtenidos y al planificar futuras investigaciones relacionadas con la clasificación automática de tickets de soporte técnico.

4.2.1 Dependencia de la calidad de los datos históricos

El desempeño del modelo de Machine Learning depende directamente de la calidad de los datos históricos registrados en la plataforma Webcoopec Ticket. Se identificaron registros con categorías no definidas, información incompleta, inconsistencias en la clasificación manual y diferencias en los criterios utilizados por los analistas de soporte durante distintos períodos de operación, lo que requirió procesos adicionales de depuración y normalización antes del entrenamiento del modelo.

4.2.2 Alcance limitado al contexto organizacional de Webcoopec System

La investigación fue desarrollada utilizando información proveniente exclusivamente de la mesa de ayuda de Webcoopec System. En consecuencia, el modelo fue entrenado con terminología, procesos operativos y categorías propias de la organización, por lo que su aplicación directa en otras empresas requerirá un proceso previo de adaptación y reentrenamiento utilizando datos representativos del nuevo entorno.

4.2.3 Variabilidad del lenguaje utilizado por los usuarios

Los tickets de soporte exhiben discrepancias significativas en la manera en que los usuarios articulan sus necesidades. El uso de abreviaturas, errores ortográficos, términos técnicos, lenguaje coloquial y descripciones incompletas puede influir en la habilidad del modelo para reconocer la categoría adecuada, especialmente cuando la información proporcionada es insuficiente.

4.2.4 Evolución permanente de las categorías de soporte

Las categorías utilizadas por la mesa de ayuda pueden modificarse con el tiempo debido a cambios en los procesos internos, incorporación de nuevos módulos del sistema o actualización de los servicios ofrecidos por la empresa. Esta situación implica que el modelo deberá ser actualizado y reentrenado periódicamente para mantener un adecuado nivel de desempeño.

4.2.5 Restricciones temporales del desarrollo

Por la cantidad de tiempo para realizar la investigación, el estudio se centró en la implementación y evaluación de modelos clásicos de aprendizaje supervisado, sin incorporar modelos basados en arquitecturas profundas o grandes modelos de lenguaje (Large Language Models - LLM), cuya evaluación podría constituir una línea de investigación futura.

4.2.6 Integración del modelo como prototipo funcional

La investigación permitió desarrollar un prototipo funcional para la clasificación automática de tickets; sin embargo, su implementación definitiva dentro del entorno de producción de Webcoopec System requerirá fases adicionales de validación operativa, pruebas con usuarios finales, monitoreo del desempeño y mantenimiento continuo para garantizar su funcionamiento en condiciones reales.

4.2.7 Protección y anonimización de los datos

Por tratarse de información proveniente de una mesa de ayuda empresarial, fue necesario aplicar procesos de anonimización y limitar el uso de determinados atributos que contenían datos personales o información sensible. Si bien estas medidas garantizan el cumplimiento de la Ley Orgánica de Protección de Datos Personales (2021) y las políticas internas de seguridad de la información, también restringen el uso de algunas variables que podrían aportar información adicional al modelo.

4.3 Recomendaciones

Con base en los resultados obtenidos de la investigación, se plantean las siguientes recomendaciones:

4.3.1 Mantener un proceso continuo de actualización del modelo

Se recomienda reentrenar periódicamente el modelo de Machine Learning utilizando nuevos tickets generados por la mesa de ayuda, con el fin de mantener su capacidad predictiva frente a cambios en la terminología, nuevos módulos del sistema o modificaciones en las categorías de soporte.

4.3.2 Fortalecer la calidad de la clasificación manual

Es recomendable establecer criterios estandarizados para la categorización de tickets y capacitar continuamente al personal de soporte técnico, ya que la calidad de las etiquetas históricas influye directamente en el desempeño de los modelos supervisados.

4.3.3 Integrar el modelo en el entorno de producción

Se recomienda incorporar el modelo de clasificación automática dentro de la plataforma Webcoopec Ticket como una herramienta de apoyo al personal de soporte, permitiendo sugerir automáticamente la categoría del ticket durante su registro y manteniendo la posibilidad de validación por parte del analista.

4.3.4 Implementar tableros de inteligencia de negocios

Se recomienda aprovechar la información clasificada automáticamente para construir dashboards en Tableau que permitan monitorear indicadores relacionados con categorías, tiempos de atención, productividad de agentes, tickets cobrables, comportamiento de clientes y tendencias operativas, fortaleciendo la toma de decisiones basada en datos.

4.3.5 Evaluar modelos de inteligencia artificial más avanzados

Como línea futura de investigación, se recomienda comparar el desempeño de los modelos clásicos de Machine Learning con arquitecturas basadas en Deep Learning y Modelos de Lenguaje de Gran Escala (LLM), con el propósito de analizar posibles mejoras en la precisión de clasificación.

4.4 Aportes de la investigación

Este estudio proporciona aportes relevantes tanto para el ámbito académico como para las prácticas profesionales, al demostrar la aplicación de técnicas de Procesamiento de Lenguaje Natural y Aprendizaje Automático en un contexto empresarial real, dirigido al desarrollo de software en el sector financiero popular y solidario.

Entre los principales aportes se destacan los siguientes:

- Se desarrolló un modelo de clasificación automática de tickets de soporte técnico basado en aprendizaje supervisado, utilizando información histórica proveniente de la plataforma Webcoopec Ticket.
- Se implementó un proceso metodológico integral de acuerdo con el marco CRISP-DM, que abarca las etapas de comprensión del negocio, análisis exploratorio de datos, preparación de datos, modelado, evaluación e implementación.
- Se creó un conjunto de datos unificado a partir de varias tablas de la base de datos de osTicket, aplicando procesos de integración, limpieza, anonimización y transformación de datos textuales.

- Se compararon diferentes algoritmos de Machine Learning mediante métricas objetivas de evaluación, permitiendo seleccionar el modelo con mejor desempeño para el problema de clasificación planteado.
- Se desarrolló un prototipo funcional que demuestra la viabilidad de integrar Inteligencia Artificial dentro del sistema de gestión de tickets de Webcoopec System, apoyando la clasificación automática de nuevas incidencias.
- La investigación establece las bases para la implementación de analítica descriptiva sobre la información clasificada, permitiendo generar dashboards e indicadores que faciliten el análisis del comportamiento de clientes, agentes de soporte, categorías, tickets cobrables, tiempos de atención y tendencias operativas.
- Finalmente, el estudio constituye una referencia metodológica para futuras investigaciones orientadas a la aplicación de Ciencia de Datos, Procesamiento de Lenguaje Natural y Machine Learning en organizaciones ecuatorianas que administran grandes volúmenes de información textual mediante sistemas de mesa de ayuda.

Referencias

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS Inc. https://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/14.2/es/CRISP-DM.pdf

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. https://doi.org/10.1007/BF00994018

Das, M., Kamalanathan, S., & Alphonse, P. J. A. (2023). *A comparative study on TF-IDF feature weighting method and its analysis using unstructured dataset* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2308.04037

Feng, L., Senapati, J., & Liu, B. (2022). *TaDaa: Real time ticket assignment deep learning auto advisor for customer support, help desk, and issue ticketing systems* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2207.11187

Han, J., Pei, J., & Tong, H. (2022). *Data mining: Concepts and techniques* (4th ed.). Morgan Kaufmann.

Jurafsky, D., & Martin, J. H. (2026). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models* (3rd ed.) [Online manuscript]. https://web.stanford.edu/~jurafsky/slp3/

Ley Orgánica de Protección de Datos Personales. (2021). *Registro Oficial No. 459, Quinto Suplemento, 26 de mayo de 2021*.

Liu, F., He, X., Zhang, T., Chen, J., Li, Y., Yi, L., Zhang, H., Wu, G., & Shi, R. (2025). TickIt: Leveraging large language models for automated ticket escalation. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion '25)* (pp. 343–354). Association for Computing Machinery. https://doi.org/10.1145/3696630.3728558

Manrique-Gómez, L., & Borja Gómez, J. H. (2025). Ciencia de datos para la historia: Datificar las fuentes para una historia (predictiva). *Historia y Grafía*, (64), 97–145. https://doi.org/10.48102/hyg.vi64.541

Molino, P., Zheng, H., & Wang, Y.-C. (2018). COTA: Improving the speed and accuracy of customer support through ranking and deep networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 586–595). Association for Computing Machinery. https://doi.org/10.1145/3219819.3219851

Oancea, B. (2025). *Text classification using machine learning methods* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2502.19801

osTicket. (s. f.). *osTicket documentation*. Recuperado el 12 de agosto de 2026, de https://docs.osticket.com/

PeopleCert. (s. f.). *ITIL 4 Practitioner: Service Desk*. Recuperado el 12 de agosto de 2026, de https://www.peoplecert.org/browse-certifications/it-governance-and-service-management/ITIL-1/itil4-practices-service-desk-3706

Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of Naive Bayes text classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)* (pp. 616–623). AAAI Press. https://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf

Sánchez Trujillo, M. G., & Pérez Hernández, J. Á. (2021). Metodología CRISP-DM en la gestión de proyecto de Data Mining: Caso enfermedades dermatológicas. En *International Conference on Project Management 2021*. Universidad EAN. https://repository.universidadean.edu.co/bitstreams/d5d07653-c175-4bfe-9745-d8d9e8cc16ad/download

Superintendencia de Economía Popular y Solidaria. (2025, 16 de diciembre). *Resolución Nro. SEPS-INSEPS-IGS-DNGRT-2025-0214*. https://seps.gob.ec/?download_id=35839&sdm_process_download=1

Truss, M., & Boehm, S. (2024). *AI-based classification of customer support tickets: State of the art and implementation with AutoML* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2406.01789