



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

**TITULO DEL TRABAJO DE TITULACIÓN**

Desarrollo de una interfaz basada en algoritmos de procesamiento de lenguaje natural (NPL) para la gestión de conocimiento.

**AUTOR**

**Tomala Tigreros, James Ángel**

**PROYECTO UIC**

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

**TUTOR**

Ing. Coronel Suárez Marjorie, Mgti.

**Santa Elena, Ecuador**

**Año 2024**



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

**TRIBUNAL DE SUSTENTACIÓN**

---

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

---

Ing. Marjorie Coronel Suárez, Mgt.  
**TUTOR**

---

Ing. Walter Orozco Iguasnia, Msc.  
**DOCENTE ESPECIALISTA**

---

Ing. Marjorie Coronel Suárez, Mgt.  
**DOCENTE GUÍA UIC**



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

**CERTIFICACIÓN**

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

La Libertad, a los 04 días del mes de diciembre del año 2024

**TUTOR**



Firmado electrónicamente por:  
MARJORIE ALEXANDRA  
CORONEL SUAREZ

---

**Ing. Coronel Suárez Marjorie, Mgti**



**UPSE**

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

**Yo, Tomala Tigreros, James Ángel**

**DECLARO QUE:**

El trabajo de Titulación, desarrollo de una interfaz basada en algoritmos de procesamiento de lenguaje natural (NPL) para la gestión de conocimiento previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

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

La Libertad, a los 04 días del mes de diciembre del año 2024

**EL AUTOR**

---

**James Ángel Tomala Tigreros**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

**CERTIFICACIÓN DE ANTIPLAGIO**

Certifico que después de revisar el documento final del trabajo de titulación denominado (Título del ensayo), presentado por el estudiante, Tomala Tigreros James Ángel fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 8%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



**TUTOR**



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

**Ing. Coronel Suárez Marjorie, Mgti**



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

**AUTORIZACIÓN**

**Yo, Tomala Tigreros, James Ángel**

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

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

La Libertad, a los 04 días del mes de diciembre del año 2024

**EL AUTOR**

A handwritten signature in blue ink, appearing to read "James Ángel Tomala Tigreros", is written over a light blue grid background.

---

**James Ángel Tomala Tigreros**

## AGRADECIMIENTO

Quiero expresar mi sincero agradecimiento a todas las personas que contribuyeron de manera invaluable en la realización de este proyecto.

En primer lugar, a mi familia por su constante aliento, comprensión y amor incondicional. Su apoyo emocional fue esencial para mantenerme enfocado y motivado a lo largo de este arduo proceso.

A mi tutora Ing. Marjorie Coronel, por su orientación experta y su apoyo a lo largo de todo este proceso. Sus conocimientos y consejos fueron fundamentales para dar forma y enfocar adecuadamente mi investigación.

Por último, agradecer profundamente a mi grupo de amigos, cuyo respaldo, colaboración y motivación fueron cruciales para completar este proyecto. Su ayuda y compañía en los momentos más desafiantes hicieron que este camino fuera mucho más llevadero y significativo.

*James Ángel, Tomala Tigreros*

## DEDICATORIA

Dedico este trabajo a las personas que han sido mi fuente de inspiración y apoyo a lo largo de este viaje académico.

A mi madre, mi padre, mi tío materno, cuyo amor inquebrantable y sacrificio me han brindado las bases para alcanzar mis metas. A usted, le dedico este logro con gratitud y cariño.

A todas las personas que contribuyeron de alguna manera en este proceso, les agradezco por ser parte de este logro. Este trabajo está dedicado a cada uno de ustedes como muestra de mi agradecimiento y afecto

*James Ángel, Tomala Tigreros*

# ÍNDICE GENERAL

TÍTULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD.	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGUREAS	XIV
RESUMEN	XVI
ABSTRACT.	XVI
INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN	2
1.1. Antecedentes	2
1.2. Descripción del Proyecto	4
1.3. Objetivos del Proyecto	8
1.4. Justificación del proyecto	9
1.5. Alcance del proyecto	11
1.6. Metodología del Proyecto	12
1.6.1. Metodología de la investigación	12

1.6.2.	Beneficiarios del proyecto	12
1.6.2.1.	Población	13
1.6.2.1.1.	Población de algoritmos	13
1.6.2.2.	Muestra	13
1.6.2.2.1.	Muestra de algoritmos	13
1.6.3.	Variables	13
1.6.4.	Análisis de recolección de datos	14
1.6.4.1.	Revisión de la literatura	14
1.6.4.2.	Revisión de dataset's en línea	14
1.7.	Metodología de desarrollo	14
CAPÍTULO 2. PROPUESTA...		17
2.1.	Marco Contextual	17
2.2.	Marco conceptual	17
2.2.1	Conocimiento táctico y explicativo	17
2.2.2	Gestión del conocimiento	17
2.2.2.1	Modelos de gestión del conocimiento	18
2.2.3	Machine Learning	19
2.2.5	Clasificación de Métricas de Evaluación de Modelos de Clasificación	22
2.2.6	Base de datos	23
2.2.7	Base de conocimiento	24
2.2.8	Modelos basados en Transformers:	24
2.2.9	Embeddings	25
2.2.10	SVM	25
2.2.11	Django	25
2.2.12	Python	26
2.2.13	Java script	26
2.3.	Marco Teórico	26
2.3.1	Gestión del conocimiento basado en el Modelo Nonaka y Takeuchi	26
2.3.2	Gestión de servicios en contexto empresariales	27
2.3.3	Machine Learning (ML) en servicios de T.I	28

2.4 Requerimientos	30
2.4.1 Requerimientos Funcionales	30
2.4.2 Requerimientos No Funcionales	33
2.5. Componente de la Propuesta	34
2.5.1 Revisión bibliográfica y análisis documental	34
2.5.2 Diseño Conceptual del Sistema:	36
2.5.2.1 Arquitectura del Sistema	36
2.5.2.2 Diseño la base de datos	36
2.5.2.3 Diccionario de datos	37
2.5.2.4 Diseño de Interfaces	40
2.5.3. Desarrollo del prototipo	46
2.5.4 Simulación y Evaluación	53
2.6 Pruebas	58
2.7 Resultados	62
2.7.1 Análisis de resultados	62
2.7.2 Resultados finales	63
CONCLUSIONES	65
RECOMENDACIONES	67
REFERENCIAS.	69
ANEXOS	78

## ÍNDICE DE TABLAS

Tabla 1 Beneficios de usar comparación semántica	10
Tabla 2 Requerimientos funcionales - Módulo NLP	31
Tabla 3 Requerimientos funcionales - Módulo de atención al usuario	32
Tabla 4 Requerimientos funcionales - Módulo de registro de conocimiento	33
Tabla 5 Requerimientos funcionales - Módulo de almacenamiento	33
Tabla 6 Requerimientos no funcionales	34
Tabla 7 Componentes de la tabla principal	38
Tabla 8 Componentes de la tabla embeddings	38
Tabla 9 Componentes tabla historial_conocimiento	39
Tabla 10 Componentes tabla sugerencias	39
Tabla 11 Componentes tabla ranking	40
Tabla 12 Ejemplo de datos generados con el modelo GPT	52
Tabla 13 Caso de uso consultas generales	55
Tabla 14 Caso de uso para proceso de carga individual	57
Tabla 15 Caso de uso para realizar una carga masiva	58
Tabla 16 Caso de prueba de comparación semántica consulta	59
Tabla 17 Caso de prueba comparación semántica respuesta	60
Tabla 18 Caso de prueba actualización de registros	61
Tabla 19 Caso de prueba de uso generar o actualizar embeddings	62
Tabla 20 Análisis de pruebas	63
Tabla 21 Tabla de resultados F1 Score del modelo SVM	80

Tabla 22 Datos sintéticos generados para la búsqueda de la mejor estrategia para la interacción NLP	82
Tabla 23 Resultados del uso de TF-IDF más función coseno	83
Tabla 24 Tabla de cluster asignados a las instrucciones originales	84
Tabla 25 Asignación de cluster a instrucciones sintéticas por el algoritmo Kmeans	84
Tabla 26 Validación de similitud por el modelo T5 y Bert-Base-Spanish	85
Tabla 27 Dataset en línea analizados	91

## ÍNDICE DE FIGUREAS

Figura 1 Infraestructura del proyecto	7
Figura 2 Estructura del modelo Kerschberg	19
Figura 3 Cuadro comparativo de modelos de gestión del conocimiento	35
Figura 4 Arquitectura del modelo Kerschberg adaptada para el Proyecto	36
Figura 5 Arquitectura cliente/servidor en tres capas	36
Figura 6 Arquitectura de la base de datos	37
Figura 7 Interfaz inicial al acceder al portal del conocimiento	41
Figura 8 Segunda fase luego de clasificar y comparar la consulta	41
Figura 9 Tercera fase una vez elegida una respuesta	41
Figura 10 Ultima fase dedicada a generar una sugerencia para la respuesta actual	42
Figura 11 Fase inicial de la interfaz de carga masiva	43
Figura 12 Fase final en el caso de existir un registro con similaridad	43
Figura 13 Fase inicial la interfaz de carga masiva	43
Figura 14 Fase luego del acondicionamiento de datos	44
Figura 15 Fase inicial de la interfaz de actualización	45
Figura 16 Modal para la actualización de registros	45
Figura 17 Fase inicial de la interfaz de generación de embeddings	46
Figura 18 Fase de visualización de los registros a los que se le genero los embeddings	46
Figura 19 Caso de uso portal del conocimiento	53
Figura 20 Caso de uso portal de gestión del conocimiento	54
Figura 21 Caso de uso general para las consultas del usuario	54
Figura 22 Caso de uso para ingresar un registro de forma individual	56

Figura 23 Caso de uso para ingresar varios registros	57
Figura 24 Código usado para entrenar el modelo de maquinas SVM	80
Figura 25 Matriz de confusión del modelo SVM	81
Figura 26 Código usado para genrar la clasificación con el modelo SVM	86
Figura 27 Código para la comparación semántica de la consulta del usuario	87
Figura 28 Código de la comparación semántica para respuestas	88
Figura 29 Código para generar embeddings	89
Figura 30 Código para insertar embeddings	90
Figura 31 Código para actulizar embeddings	90
Figura 32 Código para separar registros con respuestas estructuradas y planas	92
Figura 33 Eliminación de placeholders	92
Figura 34 Código para hacer la traducción colocando el modelo usando GPU como CPU	94
Figura 35 Carga del dataset al modelo GPT 4min usando la interfaz de ChatGPT	94
Figura 36 Evidencia de generación de datos sintéticos	95

## RESUMEN

Este proyecto de titulación se enfoca en el desarrollo de un prototipo de interfaz web para la gestión automatizada del conocimiento, optimizando la recuperación de información sobre soporte técnico mediante procesamiento de lenguaje natural (NLP). Se utilizó el modelo Kerschberg para definir la arquitectura de la base de datos e interfaces y se entrenó un modelo de máquinas de soporte vectorial (SVM) con un 96% de precisión en clasificación de texto. Se generó un dataset sintético mediante ChatGPT simulando incidentes reales de soporte, lo que permitió evaluar la similitud en consultas y respuestas. La interfaz facilita la interacción del usuario mediante consultas en lenguaje natural obteniendo respuestas rápidas mejorando la eficiencia del sistema, reduciendo el tiempo de respuesta y optimizando la gestión del conocimiento, permitiendo una mayor accesibilidad y precisión en la recuperación de información relevante.

**Palabras claves:** Modelos de gestión del conocimiento, Embeddings, Comparación semántica.

## ABSTRACT

This thesis project focuses on the development of a web interface prototype for automated knowledge management, optimizing the retrieval of information on technical support through natural language processing (NLP). The Kerschberg model was used to define the architecture of the database and interfaces, and a support vector machine (SVM) model was trained with 96% accuracy in text classification. A synthetic dataset was generated using ChatGPT to simulate real technical support incidents, which allowed for the evaluation of similarities in queries and responses. The interface facilitates user interaction through natural language queries, providing quick responses, improving system efficiency, reducing response time, and optimizing knowledge management, thereby allowing for greater accessibility and accuracy in retrieving relevant information.

**Keywords:** Knowledge management models, Embeddings, Semantic comparison.

# INTRODUCCIÓN

En la actualidad el volumen de conocimiento producido a través de la experiencia recogido de manuales e informes se incrementa de manera exponencial lo que presenta retos para su administración y uso. Un sistema de administración del saber unifica procesos, simplificando la solución de problemas. De igual manera el uso del procesamiento de lenguaje natural (NLP) contribuye a superar este reto de organizar el conocimiento no estructurado, como apuntes de apoyo lo que acelera las búsquedas y potencia la precisión en las respuestas.

Este proyecto de titulación se enfoca en desarrollar una interfaz web empleará técnicas de procesamiento de lenguaje natural con el fin de proporcionar a los usuarios una interacción eficiente con una base de datos la meta es que los usuarios puedan acceder y solucionar de manera independiente problemas habituales de soporte técnico de nivel 1. Esto disminuirá el esfuerzo del equipo de asistencia, sino que también potenciará la experiencia del usuario al simplificar la solución independiente de problemas.

En el primer capítulo se realizó una revisión bibliográfica sobre los problemas generados por la inadecuada o inexistente gestión del conocimiento que se sirven de sustento para el desarrollo del proyecto. Se presentan los objetivos del proyecto la necesidad del mismo y los métodos utilizados para desarrollar la propuesta tecnológica.

En el segundo capítulo se examinan los aspectos contextuales, conceptuales y teóricos relacionados con la propuesta tecnológica basados en los problemas de gestión del conocimiento identificados en la revisión bibliográfica. Son expuestos los conceptos fundamentales para entender la aplicación del procesamiento de lenguaje natural (NLP) en la administración del conocimiento y se examinan las teorías que respaldan el proyecto. También se especifican los requerimientos la solución sugerida y el estudio del rendimiento previsto.

# CAPÍTULO 1. FUNDAMENTACIÓN

## 1.1. Antecedentes

Los desafíos que se presentan al no existir una correcta gestión del conocimiento son la dificultad para compartir y aprovechar el conocimiento, pérdida de conocimiento valioso cuando empleados dejan la empresa, duplicación de esfuerzos debido a la falta de accesibilidad del conocimiento existente [1]. Esto trae consigo que una empresa tenga una menor capacidad de innovación y adaptación a los cambios.

La gestión del conocimiento enfrenta barreras y desafíos significativos en el entorno actual. Según el artículo "Gestión del conocimiento empresarial en la Internet de las cosas", el principal reto es integrar grandes volúmenes de datos en sistemas de conocimiento organización, existe la dificultad de organizar y convertir esos datos en conocimiento útil debido al procesamiento lento y a la creciente cantidad de información generada [2].

La inexistencia de una plataforma para gestionar el conocimiento puede derivar en varios inconvenientes como el tiempo invertido por los colaboradores en la búsqueda o recuperación de conocimiento que ya existe, pero que es difícil de localizar en los archivos de la organización, aumentando el costo en tiempo de investigación por parte del personal para la resolución de un problema similar que pudo ser ya resuelto en el pasado [3].

En el artículo "Natural language processing methods for knowledge management—Applying document clustering for fast search and grouping of engineering documents" Los autores destacan las principales problemáticas que enfrentan los arquitectos de productos, las cuales incluyen: dificultades para encontrar rápidamente los documentos relevantes, debido al hecho de que los sistemas TI actuales carecen de las capacidades necesarias para recuperar con precisión documentos pertinentes a partir de datos no estructurados (como texto libre) además cuando se realizan búsquedas en grandes bases de datos de Solicitudes de Cambio de Ingeniería (ECR), los resultados devuelven cientos de informes, lo que convierte en una tarea que consume mucho tiempo identificar los documentos más relevantes [4].

Los incidentes de nivel 1 suelen estar relacionados con problemas recurrentes que los usuarios enfrentan a diario, como dificultades de acceso, configuraciones básicas de software o errores simples de operación. La ausencia de herramientas efectivas para centralizar y gestionar el conocimiento sobre estas soluciones puede generar tiempos de respuesta más lentos y provocar la duplicación de esfuerzos por parte de los técnicos [5].

A partir de la revisión bibliográfica, se evidencia que las organizaciones enfrentan un gran desafío para realizar una correcta gestión del conocimiento, dentro de los desafíos que enfrentan es la centralización del conocimiento esto hace necesario el desarrollo de soluciones tecnológicas que potencien la comunicación entre una base de conocimientos y el usuario. Este trabajo propone investigar soluciones efectivas mediante el uso de modelos de procesamiento de lenguaje natural (NLP) para acceder al conocimiento que reposa en la base de conocimientos.

La automatización de la gestión del conocimiento está transformando la manera en que las organizaciones acceden, organizan y utilizan la información, este enfoque solo mejora la eficiencia en la recopilación y recuperación de datos, sino que también optimiza la colaboración interna, facilita la toma de decisiones y otorga ventajas competitivas en sectores industriales [6].

Los sistemas de recuperación de información que utilizan técnicas como la indexación que es el proceso de asignar términos específicos a los documentos y clasificación que es el proceso de asignar categorías temáticas a los documentos, ayuda a los usuarios a encontrar documentos relacionados con un tema específico o información relevante. Lo anteriormente descrito se puede lograr con el uso de técnicas de NLP que puede mejorar significativamente la clasificación y recuperación de información, al permitir un procesamiento más profundo y contextual del lenguaje utilizado en los documentos, ayudando a los usuarios a encontrar la información que necesitan de manera más eficiente [7].

El uso de modelos capaces de generar embeddings con una representación semántica precisa, junto con la incorporación de modelos de lenguaje de última generación y sistemas de recuperación de información basados en NLP han sido algunas de las principales mejoras en la recuperación de información. Demostrado un mejor rendimiento en comparación con los enfoques tradicionales, la capacidad de los modelos de

Transformer de modelar eficientemente los datos secuenciales y entender a todas las posiciones del texto de manera simultánea [8].

Para la solución de problemas recurrentes, el uso de sistemas de clasificación automática son herramientas útiles, el uso de una lógica matemática permite la formalización del razonamiento y validación de conclusiones a través de métodos de deducción proporciona un enfoque mecánico al análisis de problemas complejos, actúa como puerta de entrada al machine learning, facilitando su implementación de soluciones más efectivas, la clasificación automática ordena la información en estructuras reconocidas [9]. Ambas herramientas trabajan juntas para proporcionar una estrategia sistemática y eficiente para poder resolver el problema de centralización del conocimiento.

Por lo antes expuesto, se propone desarrollar una interfaz web basada en algoritmos de procesamientos de lenguaje natural (NLP) para la centralización del conocimiento y automatización de procesos de la gestión del conocimiento. Con esta solución los usuarios podrán acceder de forma rápida al conocimiento que reposa en la base de conocimientos facilitando la resolución de incidentes de soporte técnico de nivel 1 sin necesidad de ayuda técnica directa.

## **1.2. Descripción del Proyecto**

En los entornos de soporte técnico, la gestión eficaz de incidentes y la resolución rápida de problemas se han vuelto esenciales para mantener la productividad y la satisfacción del cliente [10]. Las organizaciones enfrentan desafíos relacionados con la pérdida de conocimiento crítico, en particular cuando el personal clave abandona la empresa o cuando las soluciones a incidentes recurrentes no están documentadas adecuadamente [11].

A medida que los incidentes de nivel 1 suelen involucrar problemas frecuentes y de fácil resolución, representan una oportunidad ideal para implementar soluciones automatizadas que reduzcan el tiempo y esfuerzo invertidos en su gestión. Los métodos tradicionales, como los manuales de resolución de problemas o las búsquedas en bases de datos no estructuradas, a menudo resultan ineficientes, ya que dependen de la experiencia del técnico y no siempre garantizan una respuesta oportuna.

En este contexto surge la necesidad de una solución que permita tanto centralizar como gestionar el conocimiento sobre incidentes técnicos de manera accesible y eficaz. Entonces se plantea realizar el desarrollo de una interfaz basada en modelos NLP como solución tecnológica para la gestión del conocimiento permitiendo a los usuarios acceder de manera rápida al mismo tiempo que precisa, a la información relevante.

Para enfrentar estos retos, se implementará un enfoque fundamentado en un modelo de administración del conocimiento diseñado para organizar, centralizar y simplificar la recuperación de información. Este modelo facilitará la organización de la base de datos para mejorar la administración del saber, estableciendo además las interfaces requeridas para automatizar los procesos relacionados, favoreciendo de esta manera la recuperación de datos y la solución de problemas. Se integrarán técnicas de procesamiento de lenguaje natural (NLP) para mejorar la interacción con esta base proporcionando a los usuarios respuestas rápidas, relevantes, pero también contextualmente precisas.

Considerando todas las tecnologías que se han introducido en los últimos años es imprescindible resaltar la Inteligencia Artificial (IA). Se refiere al empleo de sistemas de computación para llevar a cabo actividades que usualmente necesitan inteligencia humana, como el aprendizaje, la toma de decisiones, así como de la solución de problemas [12]. Dentro del marco de asistencia técnica, esta tecnología es fundamental para incrementar la eficacia y exactitud en la solución de incidentes. Este proyecto se favorece de estas competencias mediante la implementación de módulos específicos diseñados para centralizar, procesar y suministrar datos relevantes en tiempo real.

## **Módulos del Proyecto**

### **Revisión Bibliográfica y Análisis Documental**

Se centra en investigar literatura relevante para establecer fundamentos teóricos y determinar las mejores prácticas en sistemas de gestión de conocimiento y NLP. Se aborda la importancia de la creación de datos sintéticos como una estrategia para superar limitaciones de datos permitiendo generar ejemplos adicionales para entrenar modelos sin comprometer la calidad siendo crucial para asegurar que el modelo pueda manejar una amplia variedad de consultas y respuestas. Asimismo se destaca la elección del dataset dando prioridad a aquellos que muestren respuestas organizadas y clasificadas, lo que

simplifica la aplicación de algoritmos de clasificación y potencia la exactitud en la recuperación de datos significativos en situaciones particulares como el apoyo técnico.

### **Diseño Conceptual del Sistema**

Este módulo se centra en adaptar el modelo de gestión del conocimiento seleccionado para estructurar una base de datos optimizada, diseñando tablas que permitan almacenar, organizar y recuperar información relevante de manera eficiente. La estructura estará alineada con las necesidades de un sistema que facilite consultas basadas en lenguaje natural, enfocándose en la resolución de incidentes de soporte técnico. Además, las interfaces se desarrollarán del sistema específicamente para garantizar una interacción fluida entre el usuario y la base de datos, priorizando la usabilidad y la experiencia del usuario. Este diseño conceptual garantizará que los componentes del sistema trabajen de manera integrada para responder de forma precisa y ágil a las consultas realizadas.

### **Desarrollo de Prototipo**

La solución se implementará mediante una interfaz web diseñada para facilitar tanto la consulta como el mantenimiento del conocimiento organizacional. Esta interfaz estará dividida en dos componentes principales garantizando una gestión eficiente y tiempos de respuesta mínimos. A continuación, se describen las funcionalidades de cada componente:

- Sistema de respuestas automáticas basado en similitud semántica:
- Permite a los usuarios interactuar con la base de datos para realizar consultas de manera eficiente.
- Utiliza técnicas avanzadas de procesamiento de lenguaje natural para identificar y mostrar las tres respuestas más relevantes relacionadas con la consulta del usuario.
- Está diseñado para optimizar el tiempo de resolución de problemas recurrentes asegurando una experiencia intuitiva y rápida.

### **Gestión del conocimiento:**

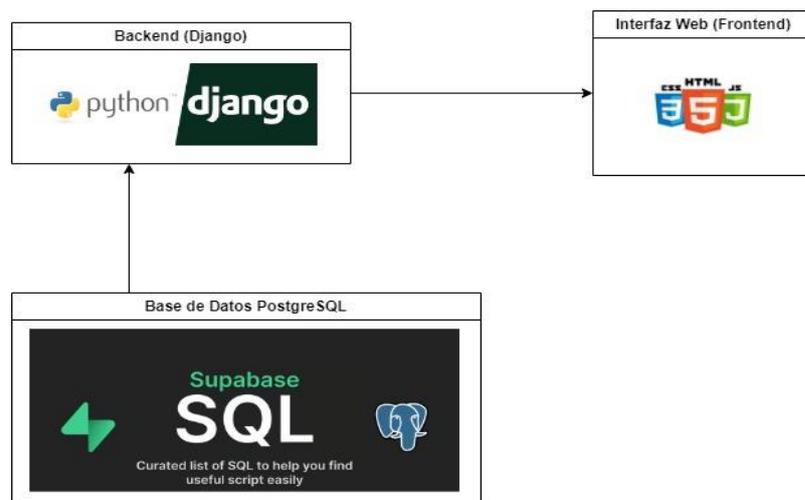
- Ofrece herramientas para agregar, actualizar y gestionar el conocimiento en la base de datos de manera estructurada.
- Facilita la ampliación y mejora continua del sistema de conocimiento asegurando su relevancia y actualidad.
- Está enfocado en usuarios con roles administrativos para mantener la calidad y precisión de los datos.

## Simulación y Evaluación

Se enfocará en garantizar que la interfaz y los algoritmos implementados cumplan con los objetivos establecidos en cuanto a eficiencia. Para ello se realizarán pruebas funcionales que evalúen la interacción de los usuarios con el sistema en ambos componentes: consultas automáticas basadas en similitud semántica y mantenimiento del conocimiento.

Igualmente se implementarán escenarios simulados con problemas recurrentes para medir la relevancia de las respuestas generadas y el tiempo de respuesta del sistema. Los resultados serán analizados para identificar áreas de mejora y optimizar tanto el diseño como el rendimiento del prototipo antes de su implementación final.

Dado los módulos mencionados se procede a mostrar la infraestructura general del proyecto a ser desarrollado demostrando cómo se intercambia información entre los distintos componentes del sistema:



*Figura 1 Infraestructura del proyecto*

Con el fin de llevar a cabo el proyecto, se emplearán las siguientes tecnologías y herramientas de desarrollo de software:

Base de datos

- PostgreSQL

Entornos de ejecución

- Django

## Frameworks

- Bootstrap
- Font Awesome
- UIverse

## Lenguajes de programación

- Python
- JavaScript
- HTML
- CSS

## Editor de código

- Visual Studio Code

### **1.3. Objetivos del Proyecto**

#### **Objetivo General:**

Desarrollar una interfaz web que permita la gestión de conocimientos sobre soluciones a incidentes de soporte técnico, empleando técnicas de procesamiento de lenguaje natural y gestores de bases de datos, con el propósito de optimizar la gestión y recuperación de información.

#### **Objetivos específicos:**

- Realizar una revisión bibliográfica de las mejores prácticas y enfoques actuales en sistemas de gestión de conocimiento y tecnologías de procesamiento del lenguaje natural.
- Seleccionar estrategias que permitan optimizar la organización y accesibilidad de la información en la implementación de algoritmos de procesamiento de lenguaje natural.
- Generar un dataset de datos sintéticos basado en información real de incidentes de soporte técnico
- Desarrollar un prototipo de interfaz que utilice procesamiento de lenguaje natural para facilitar consultas a una base de conocimientos.

#### **1.4. Justificación del proyecto**

“La sociedad del conocimiento ha tomado una importancia relevante, ya que las organizaciones que incorporan y administran el conocimiento en su quehacer son capaces de ser competitivas y adaptarse a los cambios en el entorno; El objetivo principal del proceso de gestión del conocimiento consiste en mejorar la calidad de las decisiones que se toman en las organizaciones, esto se logra gracias a que se dispone de información fiable y válida que es entregada de manera oportuna y segura” [12]. Este proceso permite optimizar la eficiencia organizacional al evitar la redistribución innecesaria de recursos y asegurar que el conocimiento acumulado esté disponible para futuras necesidades.

El uso de una interfaz web y herramientas de inteligencia artificial optimiza la gestión de soporte técnico al permitir una centralización eficiente de las solicitudes y demandas de los usuarios, al sistematizar las tareas rutinarias y operativas, se asegura un mejor seguimiento de los plazos y se evita la pérdida de información crucial, con la implementación de este tipo de sistemas contribuye a una mejora en el desempeño general del proceso de soporte técnico, aumentando la satisfacción de los usuarios al ofrecer respuestas más rápidas y precisas. [13].

En el artículo titulado “Automatización de Procesos y Eficiencia Operativa mediante Inteligencia Artificial en la Administración” se llega a concluir que el automatizar tareas repetitivas, como el manejo de incidentes comunes, se logra un aumento en la eficiencia operativa de hasta un 20%, según los estudios reportados, la capacidad del sistema para responder rápidamente y con precisión reduce la necesidad de intervención humana, disminuyendo la carga de trabajo y permitiendo que los recursos humanos se enfoquen en actividades de mayor valor estratégico [14].

En el estudio titulado “KnowID: An Architecture for Efficient Knowledge-Driven Information and Data Access” se detalla que los beneficios de integrar base de datos con ontologías y modelos conceptuales para mejorar el acceso a la información mediante una arquitectura orientada al conocimiento son: El conocimiento intensional representado en modelos conceptuales con datos almacenados en bases de datos facilita el manejo más eficiente de la información técnica, asegurando que las soluciones a incidentes comunes sean preservadas y reutilizables, evitando la pérdida de información valiosa [15].

El uso de herramientas como ServiceNow justifica la implementación de proyectos enfocados en la gestión del conocimiento al permitir la automatización, digitalización y optimización de procesos empresariales. Estas plataformas eliminan los flujos de trabajo manuales y propensos a errores maximizando la eficiencia y asegurando un acceso centralizado a información crítica [17]. Promueven la creación de soluciones novedosas y lucrativas, acelerando la transformación digital, optimizando tanto la experiencia de los trabajadores como de los consumidores.

La implementación de Procesamiento de Lenguaje Natural (NLP), técnicas de agrupación de documentos no solo permite una mejor organización, así como la recuperación de información relevante en soporte técnico, sino que también resuelve problemas asociados a la gestión del conocimiento al clasificar o agrupar automáticamente incidentes similares los equipos técnicos pueden acceder rápidamente a soluciones previas sin tener que realizar búsquedas manuales extensas [18]. Se reduce considerablemente el tiempo necesario para solucionar incidentes recurrentes de esa manera permite a los técnicos se enfoquen en problemas más complicados mejorando así la eficiencia en la eficiencia operativa.

Como se señala en “Detecting Semantic Similarity Of Documents Using Natural Language Processing” el uso de una comparación semántica para optimizar sistemas de respuestas a preguntas puede tener los siguientes beneficios:

<b><i>CAMPO</i></b>	<b><i>BENEFICIO</i></b>
<b><i>Relevancia de las respuestas</i></b>	Al analizar el significado semántico de las preguntas y las respuestas se puede mejorar la capacidad del sistema para identificar las respuestas más relevantes y precisas [19].
<b><i>Sutilezas en el lenguaje</i></b>	La comparación semántica permite detectar relaciones sutiles entre las preguntas y las respuestas más allá de la simple coincidencia de palabras lo cual es crucial para entender el significado real de las consultas [19].
<b><i>Comprensión del contexto</i></b>	Al considerar el contexto semántico, el sistema puede interpretar mejor el significado de las preguntas y proporcionar respuestas más contextualizadas [16].
<b><i>Experiencia del usuario</i></b>	Al ofrecer respuestas más relevantes y precisas, se mejora la satisfacción y la experiencia del usuario final que interactúa con el sistema de respuesta a preguntas [16].

*Tabla 1 Beneficios de usar comparación semántica*

La implementación de una interfaz web del proyecto tiene como objetivo principal optimizar la gestión de conocimiento en el soporte técnico mediante el uso de procesamiento de lenguaje natural (PLN). Esta plataforma permitirá a los usuarios realizar consultas a una base de conocimientos utilizando lenguaje natural, facilitando la búsqueda rápida de respuestas a preguntas comunes o incidentes técnicos. El sistema clasificará las consultas, proporcionando respuestas precisas basadas en la categoría correspondiente, y permitirá la interacción con una base de datos en PostgreSQL para garantizar una gestión eficiente y organizada de la información técnica.

### **1.5. Alcance del proyecto**

Con los estudios descriptivos se busca especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis [17]. Es decir, únicamente pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren, esto es, su objetivo no es indicar cómo se relacionan éstas. El alcance de este proyecto es de tipo descriptivo y tiene como propósito el desarrollo de una interfaz web para la gestión del conocimiento en entornos de soporte técnico, utilizando técnicas de procesamiento de lenguaje natural (NLP) para optimizar la recuperación de información y resolución de incidentes.

Las organizaciones enfrentan el desafío de gestionar de manera eficiente el conocimiento relacionado con incidentes de soporte técnico de nivel 1. La ausencia de herramientas automatizadas para centralizar y organizar este conocimiento resulta en tiempos de respuesta prolongados, duplicación de esfuerzos y una disminución en la calidad del servicio, lo que impacta tanto la satisfacción del usuario como la productividad operativa.

El desarrollo de esta interfaz permitirá a los usuarios acceder de forma rápida, precisa a soluciones documentadas a incidencias técnicas mejorando la eficiencia operativa y la calidad del servicio. El proyecto pretende satisfacer las necesidades de los usuarios al ofrecer información precisa, accesible mediante consultas en lenguaje natural. Las actividades claves incluyen la recopilación de dataset's de incidencias de soporte técnico y la implementación de tecnologías avanzadas de procesamiento de lenguaje natural para procesar los datos.

Además, se desarrollará una plataforma web que servirá como el canal principal de interacción entre los usuarios y el sistema de gestión del conocimiento, asegurando una experiencia de usuario fluida y eficiente. Estas actividades se estructurarán en diversas fases de desarrollo que permitirán evaluar y optimizar el rendimiento del sistema, garantizando su eficacia en la resolución de incidentes de manera automatizada y la mejora continua del servicio.

## **1.6. Metodología del Proyecto**

### **1.6.1. Metodología de la investigación**

“En el método experimental, el investigador actúa sobre algo y observa, comprueba o constata las consecuencias de esa actuación. En la investigación social esto equivaldría a decir que los investigadores toman un grupo de personas y ejercen un tipo de acción sobre ellas y registran los efectos que le ha provocado dicha actuación por parte del investigador” [18].

Dentro de este enfoque, se considerarán los siguientes pasos:

- **Diseño del experimento:** Seleccionar los algoritmos de machine learning que se desea evaluar.
- **Selección del dataset:** Utilizar un dataset de libre acceso que esté relacionado con la clasificación en un contexto similar al de un catálogo de servicios. El dataset servirá como base para entrenar y probar los modelos.
- **Implementación de los modelos:** Entrenar los algoritmos de machine learning seleccionados con el dataset, probar el modelo con distintos subconjuntos de dataset's para evaluar la consistencia de los resultados. Elección de modelos capaces de entender el idioma español.
- **Análisis de resultados:** Comparar los resultados obtenidos por cada algoritmo según las métricas definidas. Analizar cuál de ellos se ajusta mejor a las necesidades de clasificación de un catálogo de servicios.

### **1.6.2. Beneficiarios del proyecto**

El proyecto es de naturaleza descriptiva y se centra en la gestión del conocimiento en entornos de soporte técnico mediante el uso del procesamiento del lenguaje natural (PLN). Entre los posibles beneficiarios se encuentran:

- Los agentes de soporte técnico se beneficiarán directamente de tener acceso a una interfaz que les permita gestionar y recuperar el conocimiento.

- Los usuarios finales del sistema pueden obtener soluciones a problemas recurrentes sin intervención directa lo que reduce los tiempos de espera y aumenta la satisfacción.

### **1.6.2.1. Población**

En el artículo “Concepto de superpoblación en el diseño de experimentos geotécnicos de laboratorio” se define que, en estadística, el concepto de población conceptual se utiliza para describir una población imaginaria más amplia que permite a los investigadores suponer que se satisfacen ciertas propiedades como infinitud y probabilidad constante de selección [19]. La población conceptual no deberá tener elementos reales, sino que sus diferentes casos se pueden obtener por la acción repetitiva de experimentos [20].

#### **1.6.2.1.1. Población de algoritmos**

En este contexto, los algoritmos de *machine learning* de aprendizaje supervisado y semi-supervisado. Estos enfoques son cruciales para abordar problemas complejos, ya que automatizan procesos, mejoran la gestión de tareas y la eficiencia de la resolución, y facilitan la toma de decisiones basada en datos al extraer patrones y relaciones significativas de grandes cantidades de datos.

### **1.6.2.2. Muestra**

#### **1.6.2.2.1. Muestra de algoritmos**

Para este proyecto se hace uso de una muestra no probabilística por conveniencia. Se seleccionaron los modelos BERT, T5 y GPT 4mini como parte de una muestra no probabilística por conveniencia debido a su alto rendimiento en tareas de procesamiento del lenguaje natural, su disponibilidad preentrenada y la existencia de herramientas para realizar fine-tuning. Además, estos modelos presentan características complementarias: BERT es eficaz en tareas de comprensión, T5 en transformaciones de texto a texto, y GPT en generación de texto, lo que los hace ideales para abordar las necesidades del proyecto.

### **1.6.3. Variables**

**Tiempo de consulta:** es el tiempo que el usuario tarda al realizar una consulta de manera presencial o virtual información básica sobre los programas de maestrías que llegan a los coordinadores.

#### **1.6.4. Análisis de recolección de datos**

##### **1.6.4.1.Revisión de la literatura**

En el contexto de este proyecto, la recolección de datos se llevó a cabo a través de una revisión de documentos científicos y técnicos. Se analizaron estudios previos, artículos científicos y tesis que abordaron la gestión del conocimiento en entornos de soporte técnico, así como el uso de algoritmos de procesamiento de lenguaje natural (NLP) para la automatización de tareas repetitivas.

Durante la revisión se identificaron los principales desafíos en la gestión de incidentes técnicos tales como la pérdida de conocimiento crítico y la falta de herramientas automatizadas para gestionar soluciones recurrentes. Se examinaron estudios que detallaban los beneficios de implementar sistemas basados en inteligencia artificial para mejorar la eficiencia operativa y la calidad del servicio, la información recolectada proporcionó una base sólida para el desarrollo de la interfaz web propuesta en este proyecto, permitiendo identificar los módulos clave del sistema: procesamiento del lenguaje natural, generación de respuestas automáticas y la interacción a través de una plataforma web.

##### **1.6.4.2.Revisión de dataset´s en línea**

Para la recolección de datos en este proyecto, se utilizaron conjuntos de datos publicados en fuentes accesibles en línea, específicamente relacionados con incidentes de soporte técnico. Estos datasets contienen problemas recurrentes y sus soluciones en formato de respuestas estructuradas y no estructuradas. La selección de estos datos se basó en su relevancia para entrenar los modelos de procesamiento de lenguaje natural (NLP) empleados en la gestión de incidentes técnicos.

#### **1.7. Metodología de desarrollo**

El enfoque metodológico propuesto en el trabajo "Metodologías en Sistemas Basados en Multi-Agentes" [21], ha sido seleccionado como marco de referencia para el desarrollo de este proyecto debido a su estructura organizada y su aplicabilidad en el diseño de sistemas basados en agentes. Sin embargo, no todas las fases de dicho modelo han sido adoptadas, ya que algunas no son directamente aplicables al objetivo central de este

proyecto, que es la creación de un sistema de gestión de conocimiento sobre soluciones a incidentes de soporte técnico.

A continuación, se presentan las fases seleccionadas, justificando su elección en función de la relación con el modelo de gestión del conocimiento y su relevancia para el desarrollo del prototipo propuesto.

### **Revisión Bibliográfica y Análisis Documental:**

Esta fase es fundamental para obtener una base sólida en la que fundamentar las decisiones de diseño del sistema permitiendo identificar las mejores prácticas y enfoques actuales, elementos esenciales al desarrollar un sistema robusto de gestión de conocimiento.

#### **Actividades:**

- Búsqueda sistemática de literatura en bases de datos académicas.
- Análisis crítico y síntesis de la información recabada.
- Desarrollo de un marco teórico basado en los hallazgos.

### **Diseño Conceptual del Sistema:**

El diseño conceptual del sistema se ajusta al paso de Identificación de Agentes y Asociación de Roles y Metas en la metodología de sistemas multi-agente. En este paso, se identifican los elementos clave del sistema, se conceptualizan los agentes y sus interacciones, y se elabora un diseño que permita visualizar cómo estos agentes se interrelacionarán para gestionar el conocimiento de manera eficiente.

#### **Actividades:**

- Creación de esquemas y diagramas para la base de datos y la arquitectura del sistema.
- Diseño de mockups para la interfaz de usuario basados en los requisitos de usuario.

### **Desarrollo de Prototipo:**

Esta fase es comparable a la etapa de Selección de la Arquitectura de los Agentes y Producción del Código en el modelo multi-agente. Al igual que en esos enfoques, es

necesario elegir las herramientas y tecnologías apropiadas para implementar el sistema. La integración de datos y la codificación del prototipo aseguran que el sistema cumpla con los requisitos funcionales, optimizando el flujo de información y las interacciones entre los agentes del sistema.

**Actividades:**

- Desarrollo y codificación de la aplicación utilizando los lenguajes y frameworks seleccionados.
- Integración de Datos: Incorporación del dataset en la base de conocimientos para ser utilizada en pruebas y simulaciones del funcionamiento del sistema.

**Simulación y Evaluación:**

Esta fase se alinea con la etapa de Identificación de Interacciones y Selección de Protocolos de Comunicación en la metodología multi-agente, ya que es necesario evaluar cómo los agentes se comunican entre sí y verificar que todo el sistema funciona de manera eficiente en términos de tiempo de respuesta y experiencia de usuario. Estas pruebas aseguran que el sistema sea funcional, rápido y fácil de usar, lo cual es crucial para garantizar la efectividad del sistema en la gestión del conocimiento.

**Actividades:**

- Pruebas de funcionalidad: Se evaluará si el sistema cumple con los requisitos básicos de funcionamiento, comprobando si responde correctamente a las consultas.
- Pruebas de rendimiento: Se medirá el tiempo de respuesta y la eficiencia del sistema en términos de la velocidad para entregar resultados.
- Evaluación de la interfaz: Se comprobará la facilidad de uso de la interfaz y la calidad de la experiencia del usuario mediante pruebas sencillas de navegación

## **CAPITULO 2. PROPUESTA**

### **2.1. Marco Contextual**

El marco contextual de este proyecto está dado en función de dataset que presenta una amplia variedad de escenarios relacionados con preguntas frecuentes, resolución de problemas técnicos, gestión de pedidos y consultas generales, lo que lo hace ideal para proyectos que buscan abordar incidentes recurrentes y proporcionar respuestas automatizadas precisas. Su estructura y contenido se enfocan en resolver problemas relacionados con el soporte técnico de nivel 1, lo que lo hace relevante para empresas que ofrecen productos o servicios relacionados con el E-commerce y retail.

Una empresa E-commerce y retail se refiere a la comercialización de bienes y servicios a través de plataformas digitales, aprovechando la conectividad global para llegar a clientes de manera directa, eliminando barreras geográficas; Este modelo complementa al sector **retail**, que tradicionalmente se enfoca en la venta de productos al consumidor final en puntos físicos, y se transforma con estrategias omnicanal, integrando lo físico y lo digital para ofrecer experiencias de compra más personalizadas y eficientes [22].

### **2.2. Marco conceptual**

#### **2.2.1 Conocimiento táctico y explicativo**

El conocimiento táctico se refiere al conocimiento que reside en las personas, es difícil de articular y transmitir, y se adquiere a través de la experiencia; El conocimiento explicativo es aquel que puede ser codificado, documentado y transmitido de manera formal y sistemática [23].

#### **2.2.2 Gestión del conocimiento**

Para Ikujiro Nonaka la gestión del conocimiento es un sistema facilitador de la búsqueda, codificación, sistematización y difusión de las experiencias individuales y colectivas del talento humano de la organización, para convertirlas en conocimiento globalizado, de común entendimiento y útil en la realización de todas

las actividades de la misma, en la medida que permita generar ventajas sustentables y competitivas en un entorno dinámico [24].

Para los autores Juan Carlos Mijangos Noh, Karla Suguey Manzo Cabrera definen a la gestión del conocimiento como “un proceso estratégico, guiado y operado por la disciplina y las habilidades, que promueve ventajosamente la realización de los objetivos establecidos. Este proceso se utiliza para poner en práctica el conocimiento tácito y el conocimiento explícito y maximizar el valor de la organización” [25].

### **2.2.2.1 Modelos de gestión del conocimiento**

Los modelos de gestión del conocimiento son "descriptores, conceptualizaciones y enfoques" que permiten abordar los desafíos que enfrentan las organizaciones para transformar el conocimiento individual en conocimiento organizacional y crear una cultura colaborativa que favorezca este proceso, con el fin de incrementar el patrimonio intelectual de la empresa [26].

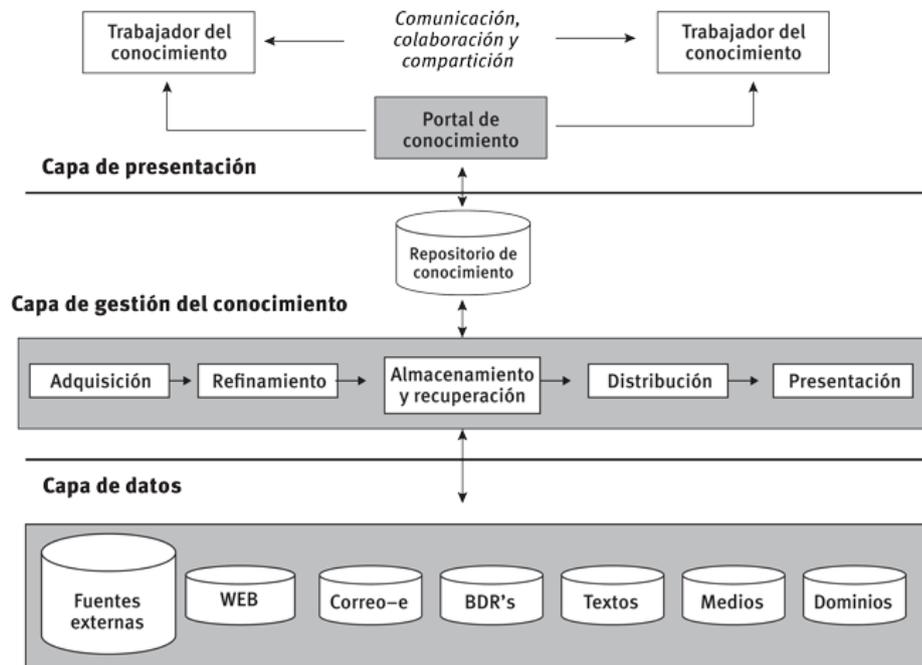
#### **2.2.2.1.1 Modelo Kerschberg**

El modelo de Kerschberg es una arquitectura integral diseñada para gestionar los activos de conocimiento en las empresas, enfocada en el uso de almacenes de datos (data warehouses). Este modelo abarca tres capas principales:

- **Capa de Presentación del Conocimiento:** Facilita la comunicación, colaboración y compartición del conocimiento a través de un portal de conocimiento personalizado para los trabajadores [27].
- **Capa de Gestión del Conocimiento:** Incluye un repositorio y los procesos clave para adquirir, refinar, almacenar, recuperar, distribuir y presentar el conocimiento [27].
- **Capa de Fuentes de Datos:** Abarca las fuentes internas y externas de información, como documentos, correos electrónicos, sitios web y repositorios multimedia [27].

Las actividades cubiertas por el modelo incluyen adquisición, refinamiento, almacenamiento/recuperación, distribución y presentación del conocimiento,

garantizando su calidad y facilitando el acceso y uso eficiente del mismo para mejorar los procesos empresariales.



*Figura 2 Estructura del modelo Kerschberg*

**Nota:** “Este modelo consiste en definir lo siguiente: 1) ayudar a asegurar la calidad de los datos y la información a los trabajadores de conocimiento; 2) contribuir con la transformación de los datos y la información en conocimiento; 3) permitir el almacenamiento eficiente y la recuperación de metadatos y de conocimiento; 4) promover la promoción a tiempo de la diseminación y la distribución del conocimiento; y, 5) apoyar la presentación adaptada de conocimiento” [28].

### 2.2.3 Machine Learning

“Machine Learning es una rama de la inteligencia artificial que se enfoca en la creación de sistemas que pueden aprender a partir de datos y mejorar su desempeño con el tiempo sin ser explícitamente programados para cada tarea específica” [29]. Este proceso de aprendizaje implica el uso de algoritmos que identifican patrones y hacen predicciones basadas en los datos de entrenamiento proporcionados. El aprendizaje supervisado, una forma común de machine

learning, consiste en aprender una función que mapea entradas (x) a salidas (y) usando ejemplos etiquetados (x, y) [29].

### **2.2.3.1 Algoritmos de aprendizaje automático**

“Un algoritmo de aprendizaje automático es un conjunto de reglas o procesos utilizados por un sistema de IA para realizar tareas, la mayoría de las veces para descubrir nuevos conocimientos y patrones de datos, o para predecir valores de salida a partir de un conjunto determinado de variables de entrada” [30]. Los algoritmos permiten que el aprendizaje automático (ML) aprenda.

### **2.2.3.2 Tipos de algoritmos de aprendizaje automático según su aprendizaje**

#### **2.2.3.2.1 Aprendizaje supervisado**

“El aprendizaje supervisado es un tipo de aprendizaje automático en el que el algoritmo aprende de los datos etiquetados, que incluyen tanto las etiquetas de los datos de entrada como las correspondientes de los datos de salida, con el objetivo de predecir o clasificar nuevos datos sobre la base de los patrones aprendidos a partir de los ejemplos etiquetados” [31].

#### **2.2.3.2.2 Aprendizaje no supervisado**

El aprendizaje no supervisado es un tipo de aprendizaje automático en el que el algoritmo aprende de los datos no etiquetados, explorando la estructura y los patrones internos de los datos, sin asesoramiento o comentarios explícitos, y tiene como objetivo descubrir información o agrupaciones ocultas [32].

La idea principal del aprendizaje no supervisado es exponer las máquinas a grandes volúmenes de datos variados y permitirles aprender e inferir a partir de ellos.

#### **2.2.3.2.3 Aprendizaje semisupervisado**

“El aprendizaje semisupervisado es un híbrido entre el machine learning supervisado y no supervisado, el algoritmo se entrena tanto con datos etiquetados como no etiquetados, primero aprende de un conjunto pequeño de datos etiquetados para hacer predicciones o tomar decisiones basadas en la información disponible, luego usa el set más grande de datos no etiquetados para refinar sus predicciones o decisiones buscando patrones y relaciones en los datos” [33].

#### **2.2.3.2.4 Aprendizaje por refuerzo.**

El aprendizaje reforzado se define como un método de aprendizaje automático que se ocupa de cómo los agentes de software deben realizar acciones en un entorno, forma parte del método de aprendizaje profundo que le ayuda a maximizar una parte de la recompensa acumulativa, esto ayuda a las redes neuronales a aprender cómo alcanzar un objetivo complejo o maximizar una dimensión específica a lo largo de muchos pasos [34].

#### **2.2.4 Procesamiento del Lenguaje Natural (NLP)**

EL NLP se define como un campo de la inteligencia artificial que se ocupa de la interacción entre las computadoras y el lenguaje humano, permite que las máquinas comprendan, interpreten y generen el lenguaje humano de una manera que sea tanto útil como significativa, incluyendo una variedad de tareas, como el análisis de texto, la extracción de información, el análisis de sentimientos y la traducción automática [35].

##### **2.2.4.1 Preprocesamiento de datos**

El preprocesamiento de datos es un paso crucial en la gestión de la cadena de suministro y en técnicas de aprendizaje automático. Incluye la limpieza de datos, transformación de datos y selección de características, con el objetivo de mejorar la calidad y utilidad de los datos antes de aplicarlos en modelos de predicción y análisis [36].

##### **2.2.4.2 Técnicas de preprocesamiento de datos**

Son un conjunto de métodos aplicados para preparar y transformar datos en un formato adecuado para su análisis, garantizando que sean válidos, confiables y relevantes para los modelos predictivos o las técnicas de minería de datos. Estas técnicas son fundamentales para mejorar la calidad de los datos, reducir el ruido y manejar problemas como valores faltantes, datos atípicos y redundancias [37].

###### **2.2.4.2.1 Tokenización**

Implica dividir una secuencia de texto en unidades más pequeñas llamadas tokens, estos tokens pueden ser palabras, subpalabras o caracteres individuales, según la

granularidad elegida para la tokenización; La importancia de la tokenización radica en su capacidad para convertir datos de texto sin procesar a un formato que pueda ser procesado de manera efectiva por redes neuronales [38].

#### **2.2.4.2.2 Lematización y stemming:**

La lematización es el proceso de reducir una palabra a su forma base o “lema”, que es su forma canónica en el diccionario, este proceso considera el contexto y las reglas gramaticales para convertir diferentes formas de una palabra en su forma raíz [39].

El stemming es una técnica más sencilla que corta los sufijos de las palabras para reducirlas a su raíz morfológica, a diferencia de la lematización, el stemming no necesariamente produce palabras con significado, sino simplemente la forma troncal de las palabras [39].

#### **2.2.4.2.3 Eliminación de palabras vacías (stop words):**

Las stop words son palabras que, aunque son esenciales para la construcción gramatical de las oraciones, no aportan un contenido significativo que ayude en tareas de análisis como la clasificación de texto; La eliminación de stop words es una técnica común en el Procesamiento de Lenguaje Natural (NLP) que implica la eliminación de palabras muy frecuentes en un idioma que no aportan un significado relevante al análisis del texto [39].

#### **2.2.4.2.4 Normalización del texto:**

“La normalización es una técnica de preprocesamiento de datos que se utiliza para escalar y estandarizar los valores de las características dentro de un conjunto de datos, el objetivo principal de la normalización es llevar todos los valores de las características a un rango similar sin distorsionar las diferencias en los rangos de valores” [40].

### **2.2.5 Clasificación de Métricas de Evaluación de Modelos de Clasificación**

#### **2.2.5.1 Precisión (Precision)**

La precisión es una métrica que evalúa la exactitud de las predicciones positivas de un modelo. Se define como:

$$Precisión = \frac{\text{Verdaderos positivos (TP)}}{\text{Verdaderos positivos (TP)} + \text{Falsos positivos (FP)}}$$

**Componentes:**

Verdaderos Positivos (TP): Cantidad de casos que fueron correctamente clasificados como pertenecientes a la clase positiva [41].

Falsos Positivos (FP): Cantidad de casos que fueron incorrectamente clasificados como positivos, pero que en realidad no pertenecen a la clase positiva [41].

**2.2.5.2 Recall (Sensibilidad o Exhaustividad)**

“El recall mide la capacidad del modelo para identificar correctamente todas las instancias positivas” [42]. Se define como:

$$Recall = \frac{\text{Verdaderos positivos (TP)}}{\text{Verdaderos positivos (TP)} + \text{Falsos negativos (FN)}}$$

**Componentes:**

Falsos Negativos (FN): Cantidad de casos que pertenecen a la clase positiva pero que el modelo clasificó incorrectamente como negativos [42].

**2.2.5.3 F1 Score**

El F1 Score es una métrica que combina precisión y recall en una sola medida. Se utiliza cuando deseas un balance entre ambas, especialmente en casos donde hay una distribución desigual de clases [42]. Se calcula de la siguiente manera:

$$F1 = 2 \times \frac{\text{Presición} \times \text{Recall}}{\text{Presición} + \text{Recall}}$$

**2.2.6 Base de datos**

“Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos se integran con una mínima cantidad de duplicidad” [43].

### **2.2.6.1 Base de datos relacionales**

Una base de datos relacional es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene su nombre: "Modelo Relacional" [44].

### **2.2.6.2 PostgreSQL**

PostgreSQL es un gestor de bases de datos orientadas a objetos (SGBDOO o ORDBMS en sus siglas en inglés) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales [45].

### **2.2.7 Base de conocimiento**

La base de conocimiento es un componente clave de los sistemas expertos. Contiene el conocimiento necesario para comprender, formular y resolver problemas, la base de conocimiento de un sistema experto contiene tanto conocimiento fáctico como heurístico. El conocimiento fáctico es el conocimiento del dominio de la tarea que se comparte ampliamente, mientras que el conocimiento heurístico es el conocimiento menos riguroso, más experiencial y más basado en el juicio, que rara vez se discute y es en gran medida individualista [46].

Una base de conocimientos es una colección estructurada de información que puede proporcionar soporte de back-end para tareas intensivas en conocimiento en aplicaciones del mundo real, como la comprensión del lenguaje, el razonamiento de sentido común y los sistemas de recomendación [47].

### **2.2.8 Modelos basados en Transformers:**

Los modelos basados en transformes han emergido como una arquitectura prominente en el aprendizaje automático, particularmente para tareas como visión por así como la generación de texto; Originalmente, los transformadores se introdujeron como una alternativa a las redes neuronales recurrentes (RNN) para

tareas de secuencia a secuencia, pero desde entonces han sido ampliamente adoptados y han permitido el desarrollo de modelos de lenguaje a gran escala [48].

#### **2.2.8.1 Bert**

BERT (Bidirectional Encoder Representations from Transformers) es un modelo de lenguaje basado en Transformers que ha logrado resultados de vanguardia en muchas tareas de procesamiento de lenguaje natural (NLP) [49].

#### **2.2.8.2 T5**

Este modelo utiliza una arquitectura basada en Transformers que ha sido entrenado en un corpus masivo, denominado Colossal Clean Crawled Corpus (C4), ha demostrado resultados sobresalientes en múltiples tareas de NLP, como traducción, resumen, respuesta a preguntas y clasificación de texto, aprovechando su capacidad para comprender y generar texto de manera flexible y eficiente [50].

#### **2.2.9 Embeddings**

Los emmdeddings de palabras son representaciones vectoriales de palabras de longitud fija, densas y distribuidas que se construyen utilizando estadísticas de coocurrencia de palabras de acuerdo con la hipótesis distributiva; Estos word embeddings han demostrado codificar información sintáctica y semántica sorprendentemente precisa, y han resultado útiles como características adicionales en muchas tareas de procesamiento de lenguaje natural (NLP) [51].

#### **2.2.10 SVM**

Las Máquinas de Vectores de Soporte (SVM) son un algoritmo de aprendizaje supervisado que permite generar clasificadores no lineales. SVM busca encontrar el hiperplano que mejor separa las clases, maximizando la distancia (margen) entre el hiperplano y los ejemplos más cercanos de cada clase, esto se logra resolviendo un problema de optimización que maximiza el margen geométrico, sujeto a que todos los ejemplos de entrenamiento tengan un margen geométrico mínimo [52].

#### **2.2.11 Django**

Django es un “software que puede utilizar para desarrollar aplicaciones web de forma rápida y eficiente. La mayoría de las aplicaciones web tienen varias funciones

comunes, como la autenticación, la recuperación de información de una base de datos y la administración de cookies. Django facilita su trabajo al agrupar las diferentes funciones en una gran colección de módulos reutilizables, llamada marco de aplicación web” [53].

### **2.2.12 Python**

“Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada, lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de plataformas” [54].

### **2.2.13 Java script**

JavaScript es un lenguaje de programación muy utilizado en el desarrollo web. Su principal cualidad es que permita agregar interactividad a la página web. Las principales características que posee este lenguaje son:

- Interactividad: JavaScript permite dinamismo, de manera que los elementos de la página web son capaces de reaccionar ante acciones del usuario, los botones cambian de estado, se procesa la entrada [55].
- Validación de formularios: Este lenguaje permite validar los datos de entrada de un formulario antes de enviarlo al servidor, lo que permite no enviar datos incorrectos [55].

## **2.3. Marco Teórico**

### **2.3.1 Gestión del conocimiento basado en el Modelo Nonaka y Takeuchi**

El artículo titulado "Managing Knowledge in Organizations: A Nonaka's SECI Model Operationalization" de Farnese, Barbieri, Chirumbolo y Patriotta ofrece una perspectiva valiosa sobre la aplicación práctica del modelo SECI de Nonaka en la gestión del conocimiento organizacional [56].

De acuerdo con el trabajo de investigación, en él se presenta dos estudios donde se evalúan la estructura y la validez del Cuestionario de Procesos de Gestión del

Conocimiento SECI (KMSP-Q), el cual centra su análisis en el modelo SECI. Estos estudios buscaban confirmar la robustez psicométrica del cuestionario y su capacidad para capturar las dimensiones del proceso de gestión del conocimiento en diversos contextos organizacionales.

**Estudio Uno:** En el primer estudio se probaron las estructuras de ocho dimensiones del Cuestionario de Procesos de Gestión del Conocimiento SECI (KMSP-Q), se lo realizó mediante el análisis factorial exploratorio y confirmatorio en una muestra de 372 empleados de diferentes sectores. Los resultados de las pruebas finales pudieron determinar la robustez psicométrica en términos de dimensionalidad y validez de constructo del K MSP-Q.

**Estudio Dos:** En el segundo estudio se examinó la validez de constructo y la fiabilidad del KMSP-Q en el contexto de trabajo intensivo en conocimiento, tomando como muestra 466 trabajadores de salud. Los resultados finales destacaron la validez de la estructura factorial y la contribución única de cada dimensión en los aspectos de rendimiento, innovación y eficacia colectiva.

Según los resultados de ambos estudios, estos determinan una sólida validación del Cuestionario de Procesos de Gestión del Conocimiento SECI (KMSP-Q), lo que confirma su capacidad para reflejar las dimensiones del modelo SECI en distintos contextos organizacionales. La robustez del cuestionario y la identificación de relaciones específicas entre las dimensiones del proceso de gestión del conocimiento organizacionales subrayan la relevancia del modelo SECI en la práctica de la gestión del conocimiento. Además, el enfoque multidimensional permite a las organizaciones comprender mejor los procesos de conversión de conocimiento y el impacto en el rendimiento y la innovación lo que a su vez, facilita la implementación de estrategias efectivas.

### **2.3.2 Gestión de servicios en contexto empresariales**

El estudio titulado "Gestión de servicio basada en programación neurolingüística en tiempos de pandemia" analiza los efectos de la Programación Neurolingüística (PNL) en la gestión de servicios de la empresa Electroperú durante la pandemia [57]. A continuación, se presentan los efectos que generados:

**Impacto en la Gestión de Servicios:** La implementación de la PNL en la gestión de servicios de Electroperú tuvo un efecto favorable, demostrando mejoras significativas en los indicadores de gestión de servicio antes y después del tratamiento. Las prácticas de PNL permitieron optimizar los procesos y mejorar la calidad del servicio al cliente [57].

**Mejora de Destrezas y Procesos de Servicio:** Las técnicas de PNL, tales como; la relajación, el anclaje, la sincronización y el reencuadre, fueron fundamentales para mejorar las destrezas de los empleados. Estas técnicas ayudaron a los empleados a manejar el estrés, a comunicarse de manera más efectiva con los clientes, y utilizar sus propios recursos de manera más eficiente, lo que dio como resultado una gestión de servicio efectiva [57].

**Evidencia de Resultados en la Gestión de Servicios:** Los resultados finales indicaron una optimización significativa en todos los procesos analizados de la gestión de servicio. La calidad de servicio y la satisfacción del cliente fueron las mejoras observadas para validar la hipótesis de que la PLN tiene un efecto positivo en la gestión del servicio durante la pandemia [57].

Finalmente, los diferentes resultados de este estudio sobre la gestión de servicio basada en PNL demuestran que no solo es viable, sino que también altamente beneficiosos para mejorar la eficiencia y la calidad del servicio al cliente. La combinación de técnicas de PNL y prácticas de gestión de servicios procuran ser una herramienta valiosa para enfrentar los desafíos planteados por situaciones de crisis, como la pandemia, además, de mejorar la capacidad de respuestas y la satisfacción del cliente en la empresa.

### **2.3.3 Machine Learning (ML) en servicios de T.I**

El artículo titulado "Leveraging AI and Machine Learning to Revolutionize IT Services: Practical Insights and Future Directions" ofrece una perspectiva valiosa sobre la implementación de tecnologías de aprendizaje automático (Machine Learning, ML) en la gestión de servicios de TI [61]. Consta de varios estudios en donde se evaluaron las mejoras junto con los beneficios obtenidos a través de la

aplicación de ML en diferentes áreas del servicio de TI destacando cómo estas tecnologías pueden transformar y optimizar las operaciones diarias.

#### Beneficios de Implementar ML en Servicios de TI

La implementación de Machine Learning en servicios de TI ofrece una amplia gama de beneficios que mejoran significativamente la eficiencia y la calidad del servicio.

Algunos de los beneficios más destacados incluyen:

- **Automatización y Eficiencia Operacional:** El ML permite la automatización de tareas repetitivas y rutinarias esto no solo reduce el tiempo de respuesta a incidentes, sino que también disminuye el margen de error humano [61].
- **Mejora en la Gestión de Incidentes y Problemas:** Mediante la aplicación de algoritmos de ML es posible predecir y detectar problemas antes de que ocurran, esto se traduce en una mayor disponibilidad del servicio [61].
- **Personalización y Mejora en la Experiencia del Usuario:** Las tecnologías de ML analizan grandes volúmenes de datos para ofrecer soluciones personalizadas a las necesidades de los usuarios mejorando la satisfacción del cliente y la percepción de calidad del servicio [61].
- **Análisis y Toma de Decisiones Basadas en Datos:** El Machine Learning simplifica el estudio de datos en tiempo real, brindando a los administradores de Tecnologías de la Información información útil para la toma de decisiones [61].
- **Optimización de Recursos y Costos:** La habilidad del Machine Learning para examinar patrones y tendencias facilita una mejor organización y uso de los recursos, lo que favorece una administración más eficaz y la disminución de los gastos operativos [61].

La tecnología de aprendizaje automático está cambiando la manera en que se gestionan los servicios de TI. Lo he visto en acción: puede hacer que las operaciones sean mucho más ágiles como efectivas. También ayuda a reducir costos y a mejorar la experiencia de los usuarios algo que realmente marca la diferencia. Además, usar datos como base para tomar decisiones importantes le da a cualquier organización un gran impulso. En un mundo tan competitivo como el de hoy creo que estas herramientas son indispensables para mantenerse al día y enfrentar los retos que vienen.

## 2.4 Requerimientos

### 2.4.1 Requerimientos Funcionales

#### 2.4.1.1 Módulo de Procesamiento de Lenguaje Natural

<i>Código</i>	<i>Especificación de requisitos</i>
<i>RF-01</i>	El sistema deberá interpretar variaciones en la formulación de preguntas para identificar sinónimos y ofrecer respuestas precisas.
<i>RF-02</i>	El sistema deberá buscar similitudes entre la consulta del usuario y las preguntas existentes en la base de datos para encontrar coincidencias relevantes para retornar una respuesta que debe guiar al usuario a resolver su incidente con instrucciones claras del paso a paso.
<i>RF-03</i>	El sistema deberá clasificar las consultas de los usuarios en categorías predefinidas para mejorar la precisión de las respuestas.
<i>RF-04</i>	El sistema deberá integrarse de manera efectiva con la base de conocimientos.
<i>RF-05</i>	El sistema deberá manejar consultas de diferente longitud y complejidad, proporcionando respuestas claras para preguntas tanto abiertas como cerradas.
<i>RF-06</i>	El sistema deberá identificar la intención de la consulta del usuario en base a las siguientes categorías y subcategorías: <ul style="list-style-type: none"><li>➤ Cuenta: crear cuenta, eliminar cuenta, editar cuenta, cambiar cuenta.</li><li>➤ Tasa de cancelación: verificar tasa de cancelación.</li><li>➤ Entrega: opciones de entrega.</li><li>➤ Factura: consultar factura, obtener factura.</li><li>➤ Pedido: cancelar pedido, cambiar pedido, hacer pedido.</li><li>➤ Pago: verificar métodos de pago, problema con pago.</li><li>➤ Reembolso: verificar política de reembolso, rastrear reembolso.</li></ul>

	<ul style="list-style-type: none"> <li>➤ Dirección de envío: cambiar dirección de envío, configurar dirección de envío.</li> </ul>
RF-07	El sistema deberá identifica errores ortográficos o gramaticales en las consultas para mejorar la precisión en el reconocimiento de intenciones.
RF-08	Los modelos de clasificación y búsqueda de similitudes deberán alcanzar un porcentaje mínimo de precisión al menos del 85% para garantizar la fiabilidad de las respuestas.

*Tabla 2 Requerimientos funcionales - Módulo NLP*

#### **2.4.1.2. Módulo de Atención al Usuario**

<i>Código</i>	<i>Especificación de requisitos</i>
RF-10	El sistema deberá permitir a los usuarios ingresar consultas a través de un campo de búsqueda accesible y visible en la interfaz.
RF-11	El sistema deberá mostrar gif de carga mientras se procesan las consultas para mejorar la experiencia del usuario.
RF-12	El sistema deberá clasificar la consulta ingresada.
RF-13	El sistema mostrará las 3 respuestas con mayor similitud en relación a la consulta ingresada en un formato de tarjetas, con una breve descripción.
RF-14	Al hacer clic en una tarjeta de respuesta, el sistema deberá actualizar la interfaz para mostrar la respuesta seleccionada en una columna principal.
RF-15	La interfaz deberá presentar las 3 mejores respuestas del lado izquierdo de la pantalla una vez se haya elegido una.

RF-16	La interfaz deberá incluir un botón de refresco que permita al usuario reiniciar la búsqueda, un botón para agregar una sugerencia a la respuesta seleccionada.
-------	---

*Tabla 3 Requerimientos funcionales - Módulo de atención al usuario*

### 2.4.1.3 Módulo de Registró de conocimiento

<i>Código</i>	<i>Especificación de requisitos</i>
<i>RF-17</i>	El sistema deberá permitir a los usuarios ingresar una consulta o incidente a través de un campo de texto en la interfaz.
<i>RF-18</i>	El sistema deberá incluir un menú accesible que permita la navegación entre diferentes secciones de la aplicación.
<i>RF-19</i>	El sistema deberá mostrar un formulario que permita al usuario seleccionar una categoría y una subcategoría de un conjunto predefinido.
<i>RF-20</i>	El sistema deberá permitir la generación de categorías e intenciones automáticamente al hacer clic en un botón específico ("Generar").
<i>RF-21</i>	El sistema deberá permitir al usuario ingresar una respuesta a la consulta o incidente en un área de texto.
<i>RF-22</i>	Al hacer clic en el botón "Validar datos" se hará una comparación entre la respuesta ingresada en el área de texto con las respuestas de la base de conocimientos.
<i>RF-23</i>	El sistema deberá mostrar un modal de confirmación cuando no se encuentren registros similares a la consulta ingresada, preguntando al usuario si desea guardar el nuevo registro.
<i>RF-24</i>	El sistema deberá mostrar un modal de error si no se puede guardar el registro, proporcionando información sobre el porcentaje de similitud

	con registros existentes, con 2 botones una para cerrar el modal y el otro para ver el registro similar.
RF-25	Al hacer clic sobre el botón “Ver registro” del modal que se muestra cuando hay un registro similar en la base de conocimientos, el sistema deberá mostrar una segunda columna donde se podrá visualizar el registro que se detectó como similar.
RF-26	Los modales de confirmación y error deberán incluir botones para cancelar o confirmar la acción (guardar registro o ver registro).
RF-27	El sistema deberá permitir al usuario cerrar la segunda columna de visualización y reiniciar el formulario mediante un botón.
RF-28	El sistema deberá mostrar mensajes de error claros y amigables en caso de que ocurran problemas durante la interacción con el formulario o la validación.

*Tabla 4 Requerimientos funcionales - Módulo de registro de conocimiento*

#### **2.4.1.4 Módulo de Almacenamiento**

<i>Código</i>	<i>Especificación de requisitos</i>
RF-29	El sistema deberá evitar la duplicación de registros, asegurándose de que no se almacenen consultas o respuestas idénticas.
RF-30	El sistema deberá permitir la recuperación de registros almacenados a través de consultas eficientes para facilitar el análisis.

*Tabla 5 Requerimientos funcionales - Módulo de almacenamiento*

#### **2.4.2 Requerimientos No Funcionales**

<i>Código</i>	<i>Especificación de requisitos</i>
RNF-01	El tiempo de respuesta del sistema a las consultas de los usuarios no deberá superar los 10 segundos para mantener una interacción fluida.

RNF-02	El sistema deberá manejar errores ortográficos comunes en las consultas de los usuarios, para facilitar la comunicación y comprensión.
RNF-03	Solo se permitirá la carga de archivos en formato CSV en el apartado de carga masiva, garantizando la compatibilidad con el sistema.
RNF-04	Los archivos CSV que se carguen deberán incluir una cabecera con el siguiente formato y orden: instruction, category, intent, response.
RNF-05	El sistema deberá ser capaz de generar automáticamente los campos category e intent utilizando los datos proporcionados en las columnas instruction y response.

*Tabla 6 Requerimientos no funcionales*

## **2.5. Componente de la Propuesta**

### **2.5.1 Revisión bibliográfica y análisis documental**

#### Modelos de gestión del conocimiento

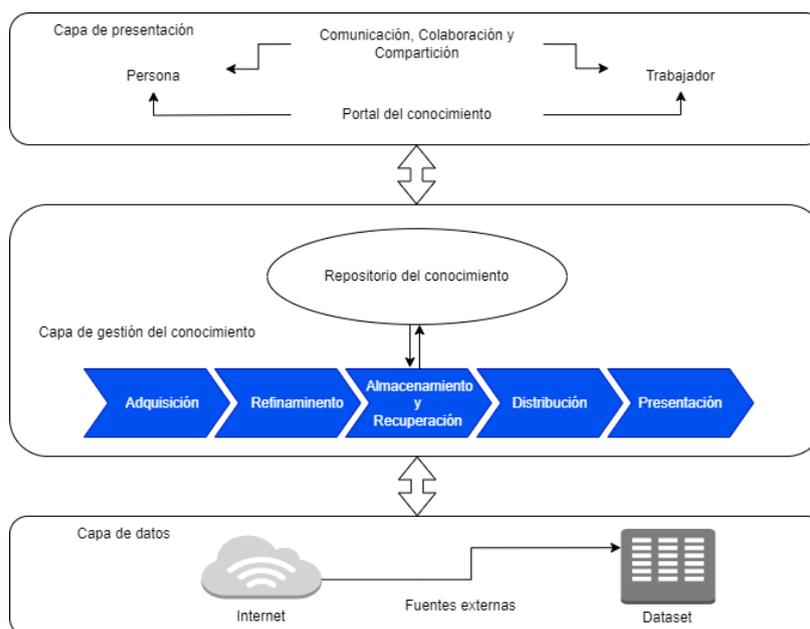
En el siguiente cuadro presenta una comparación detallada de modelos de gestión del conocimiento, considerando diversos aspectos clave como el motor de la organización, el uso de TIC, el contexto de aplicación, los actores involucrados, el tipo de conocimiento abordado y la influencia de la cultura organizacional [26].

En base a la revisión bibliográfica y el anterior cuadro comparativo se llega a la conclusión que para el proyecto se usara el modelo “Kerschberg” que es bien conocido por su orientación tecnológica al incorporar procesos como la recuperación, el almacenamiento y la presentación de información a través de sistemas tecnológicos e interfaces de usuario. Además, utiliza técnicas avanzadas como minería de datos y análisis para descubrir patrones y mejorar la organización del conocimiento. Este enfoque evidencia cómo el modelo integra la tecnología para optimizar la gestión del conocimiento, destacando el refinamiento como un valor agregado frente a otros enfoques más generales.

Descriptores						
Modelo	Motor de la gc	Uso de TIC	Contexto	Actores	Tipo de conocimiento	Cultura organizacional
Wiig (1993)	TIC	Uso de las TIC en el proceso de distribución de conocimiento	Organizaciones en general	Miembros de la organización Expertos	Factual, conceptual, explicativo y metodológico	Abierta al aprendizaje
Nonaka y Takeuchi (1995)	Procesos humanos	Un medio, sin inclusión indispensable	Empresarial	Individuos Equipos de trabajo	Tácito Explícito	Abierta al aprendizaje colaborativo
Sveiby (1997)	Procesos humanos	Sistemas de información, páginas web, intranet, bases de datos	Empresarial	Miembros de la organización Clientes Proveedores	Formal	Participativo
Bustelo y Amarilla (2001)	Procesos humanos TIC	Bases de datos corporativas Aplicaciones informáticas	Empresarial	Miembros de la organización Expertos en informática	Formal	Participativo
Kerschberg (2001)	TIC	Páginas web Correos - e Bases de datos Portal corporativo Dominios Mensajería electrónica Video conferencias Data mining	Empresarial	Ingenieros de conocimiento Expertos Usuarios de las TIC Grupos de discusión	Tácito Explícito	Comunicativo Colaborativo
Riesco (2004)	TIC Procesos humanos	Redes	Empresarial	Comunidades de práctica Equipos de gestión de conocimiento	Formal Experiencias	Colaborativo
Paniagua y López (2007)	TIC Procesos humanos	Apoyo de las TIC (entornos colaborativos o entornos de acceso y transferencia del conocimiento)	Empresarial	Miembros de la organización Expertos Líder de la organización	Tácito Explícito	Colaborativo
Angulo y Negrón (2008)	TIC Procesos humanos	Páginas web Software libre Internet	Académico	Individuos Grupos	Tácito Explícito	Comunicativo Colaborativo

Figura 3 Cuadro comparativo de modelos de gestión del conocimiento

Para este proyecto, la capa de datos de la arquitectura del modelo se adaptará para enfocarse solo en la adquisición de conocimiento desde fuentes externas, particularmente datasets obtenidos a través de la web. Este enfoque permitirá integrar información diversa y actualizada, garantizando que los datos procesados sean relevantes y respaldados por un origen confiable.



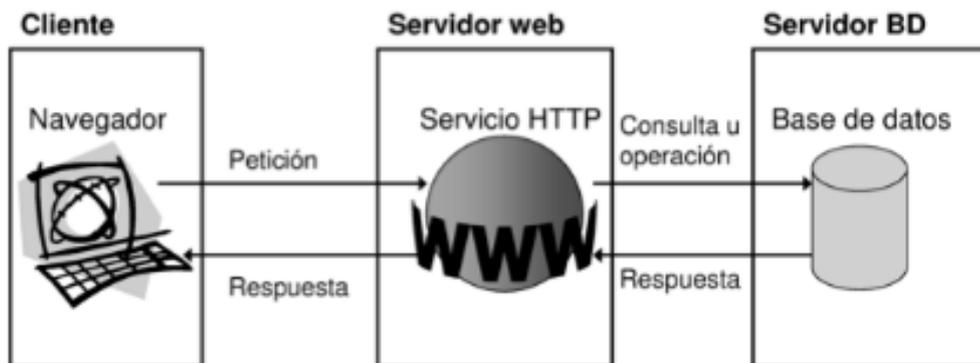
*Figura 4 Arquitectura del modelo Kerschberg adaptada para el Proyecto*

## **2.5.2 Diseño Conceptual del Sistema:**

### **2.5.2.1 Arquitectura del Sistema**

#### **Arquitectura Cliente - Servidor de tres capas**

“La arquitectura Cliente-Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras interconectadas entre ellas sea en el mismo nivel o en diferentes niveles, no tienen que estar igual” [58].



*Figura 5 Arquitectura cliente/servidor en tres capas*

#### **2.5.2.2 Diseño la base de datos**

La base de datos de este proyecto se diseña siguiendo la estructura conceptual de la capa 2 del modelo Kerschberg, que abarca los procesos clave de adquisición, refinamiento, almacenamiento y presentación del conocimiento. Esta organización asegura que cada proceso esté representado de manera clara, permitiendo un flujo estructurado y lógico para la gestión eficiente del conocimiento dentro del sistema.

El diseño incluye cinco tablas principales, cada una adaptada para cumplir con las necesidades específicas de estos procesos. Esto facilita tanto el acceso como la gestión del conocimiento, asegurando la consistencia y efectividad del sistema.

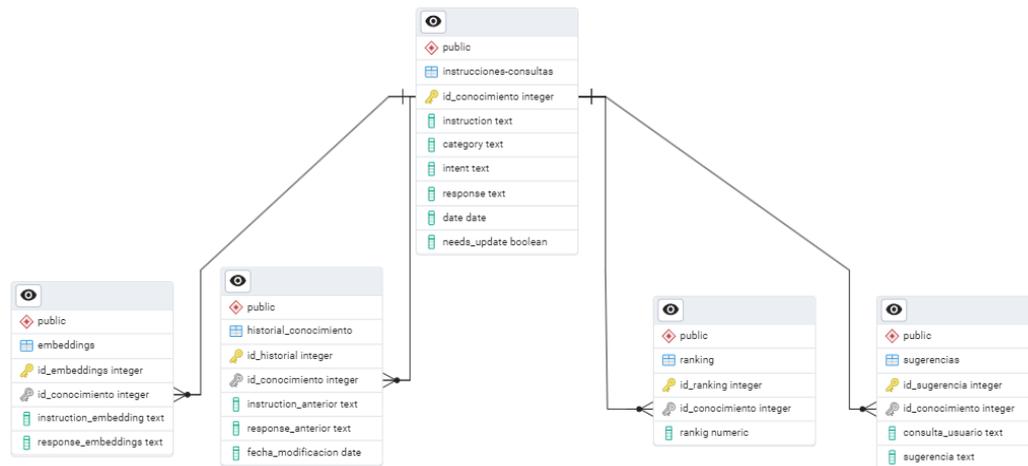


Figura 6 Arquitectura de la base de datos

### 2.5.2.3 Diccionario de datos

Un diccionario de datos se compone de información que incluye detalles sobre sus tablas, relaciones, tipos de datos y claves primarias, entre otros elementos, por lo que estos datos proporcionan a los desarrolladores una visión completa de cómo está configurada la base de datos [59].

#### Diccionario de datos

<i>Nombre de la tabla:</i>	Instrucciones-Consultas			
<i>Descripción:</i>	Captura y organiza la información, incluyendo la instrucción inicial, categoría, intención y respuesta.			
<i>Descripción de campos</i>				
<i>Descripción</i>	Tipo	Longitud	Nulo	Detalle
<i>ID_CONOCIMIENTO</i>	Integer	-	No	Clave primaria
<i>instruction</i>	Text	-	No	Incidente
<i>category</i>	Text	-	No	Categoría del incidente
<i>intent</i>	Text	-	No	Intención de la categoría
<i>response</i>	Text	-	No	Respuesta al incidente

<i>date</i>	Date	-	No	Fecha en la que se ingresó el conocimiento
<i>needs_update</i>	Boolean	-	No	Campo que cambiara en base se actualicé el registro

Tabla 7 Componentes de la tabla principal

Diccionario de datos

<i>Nombre de la tabla:</i>	Embeddings			
<i>Descripción:</i>	Almacena los embeddings de las columnas instruction y response de la tabla Instrucciones-Consultas			
<i>Descripción de campos</i>				
<i>Descripción</i>	Tipo	Longitud	Nulo	Detalle
<i>ID_EMBEDDINGS</i>	Integer	-	No	Clave primaria
<i>ID_CONOCIMIENTO</i>	Integer	-	No	Clave foranea
<i>instruction_embedding</i>	Text	-	No	Embedding generados por el modelo sentence-transformers/all-MiniLM-L6-v2
<i>response_embeddings</i>	Text	-	No	Embedding generados por el modelo sentence-transformers/distiluse-base-multilingual-cased-v2
<i>Relaciones</i>	Instrucciones-Consultas			

Tabla 8 Componentes de la tabla embeddings

Diccionario de datos

<i>Nombre de la tabla:</i>	Historial_conocimiento			
<i>Descripción:</i>	Almacena las versiones anteriores de los registros de conocimiento, permitiendo realizar un seguimiento de las modificaciones realizadas.			
<i>Descripción de campos</i>				
<i>Descripción</i>	Tipo	Longitud	Nulo	Detalle
<i>ID_HISTORIAL</i>	Integer	-	No	Clave primaria
<i>ID_CONOCIMIENTO</i>	Integer	-	No	Clave foranea

<i>instruction_anterior</i>	Text	-	No	Versión inicial de la instrucción
<i>response_anterior</i>	Text	-	No	Versión inicial de la respuesta
<i>fecha_modificacion</i>	Date	-	No	Fecha en la que se realizó la actualización
<i>Relaciones</i>	Instrucciones-Consultas			

Tabla 9 Componentes tabla *historial\_conocimiento*

*Diccionario de datos*

<i>Nombre de la tabla:</i>	Sugerencias			
<i>Descripción:</i>	Cada entrada en la tabla contiene una consulta específica y la sugerencia asociada, lo que facilita el análisis y la mejora continua de las respuestas proporcionadas.			
<i>Descripción de campos</i>				
<i>Descripción</i>	Tipo	Longitud	Nulo	Detalle
<i>ID_SUGERENCIA</i>	Integer	-	No	Clave primaria
<i>ID_CONOCIMIENTO</i>	Integer	-	No	Clave foranea
<i>consulta_usuario</i>	Text	-	No	Contiene la consulta a la que se le genero la similitud
<i>sugerencia</i>	Text	-	No	Contiene la sugerencia a la respuesta elegia por el usuario
<i>Relaciones</i>	Instucciones-Consultas			

Tabla 10 Componentes tabla *sugerencias*

*Diccionario de datos*

<i>Nombre de la tabla:</i>	Ranking			
<i>Descripción:</i>	La clasificación facilita el ordenamiento de los registros según su relevancia o calidad, optimizando la gestión del conocimiento al permitir un acceso rápido a los registros más pertinentes.			
<i>Descripción de campos</i>				
<i>Descripción</i>	Tipo	Longitud	Nulo	Detalle
<i>ID_RANKING</i>	Integer	-	No	Clave primaria
<i>ID_CONOCIMIENTO</i>	Integer	-	No	Clave foranea

<i>Ranking</i>	Number	-	No	Ranking del registro
<i>Relaciones</i>	<i>Instrucciones-Consultas</i>			

*Tabla 11 Componentes tabla ranking*

#### 2.5.2.4 Diseño de Interfaces

En base a la arquitectura presentada, se desarrollaron diversas interfaces con el objetivo de satisfacer las necesidades tanto de los usuarios generales como de los administradores. Estas interfaces se diseñaron de manera que cada una de ellas cumpla con una función específica dentro del flujo de trabajo del sistema. Para garantizar una correcta distribución y gestión de las funcionalidades, las interfaces se ubicaron estratégicamente en diferentes capas del modelo de gestión del conocimiento, específicamente en la **Capa 2 (Gestión del Conocimiento)** y la **Capa 3 (Presentación del Conocimiento)**.

#### **Interfaces capa 3 (Presentación del Conocimiento).**

Las interfaces están diseñadas para interactuar directamente con el usuario proporcionando una forma amigable para visualizar y acceder a la información. La interfaz de “consulta” es un ejemplo claro de esta capa, ya que permite al usuario realizar búsquedas obteniendo respuestas o soluciones a sus consultas sobre incidentes de soporte técnico. Además, el “Portal del Conocimiento” también forma parte de esta capa, actuando como una puerta de entrada al sistema, donde los usuarios pueden navegar entre las distintas funciones disponibles, como la consulta de información o el acceso a otras interfaces relacionadas.

#### **Fase Inicial: Portal del Conocimiento**



Figura 7 Interfaz inicial al acceder al portal del conocimiento

**Nota:** El usuario ingresa su pregunta en un campo de texto y la interfaz presenta opciones relacionadas con el incidente.

### Fase de Consulta

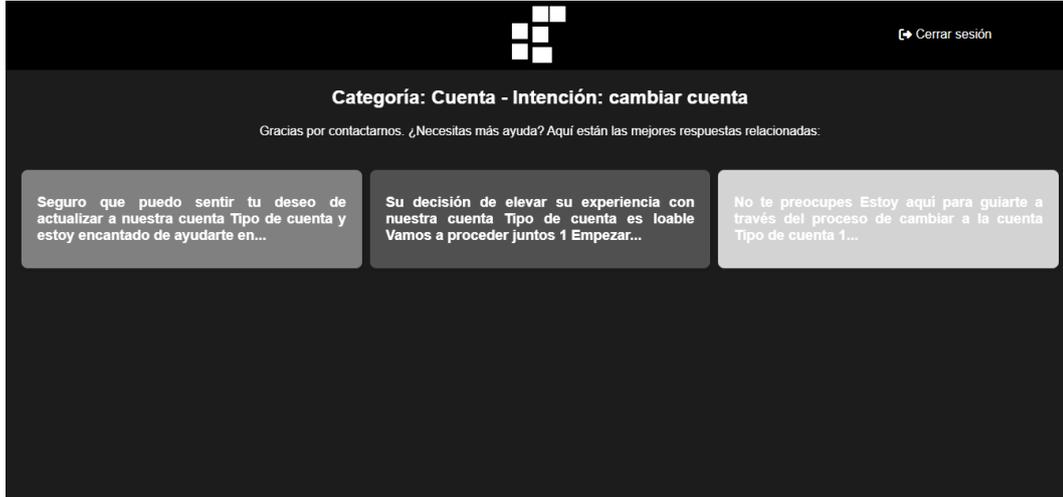


Figura 8 Segunda fase luego de clasificar y comparar la consulta

### Fase de Resultados

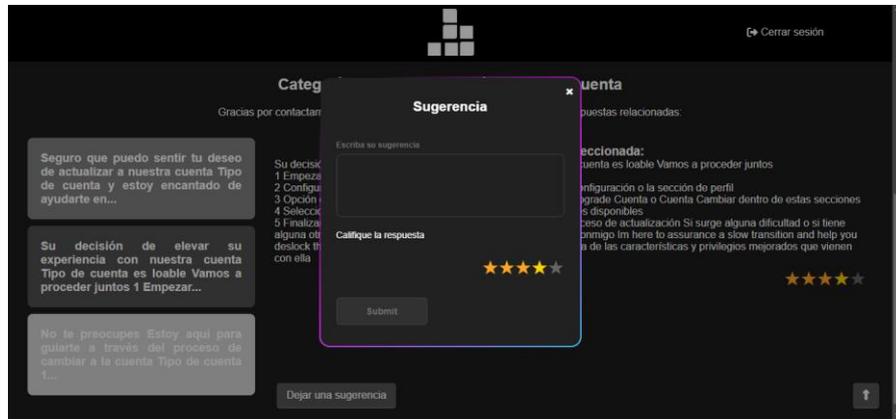


Figura 9 Tercera fase una vez elegida una respuesta

**Nota:** La interfaz permite al usuario cambiar de respuesta según sus necesidades, mostrando el ranking asignado a cada opción. Además, incluye un botón destacado que permite generar una nueva sugerencia al abrir un modal. En la parte inferior

derecha de la interfaz, los usuarios pueden interactuar con un sistema de estrellas para rankear las respuestas según su utilidad.

### Fase de sugerencias



*Figura 10 Ultima fase dedicada a generar una sugerencia para la respuesta actual*

**Nota:** En esta fase de la interfaz el usuario podrá dejar su sugerencia para la respuesta en la que se encuentre.

### Interfaces capa 2 (Gestión del Conocimiento)

Las interfaces están orientadas a gestionar el conocimiento de forma estructurada, permitiendo tareas como la carga individual y masiva de datos, la actualización de información existente, y la generación de embeddings para mejorar la recuperación de información. Estas interfaces están relacionadas con el almacenamiento, refinamiento y recuperación del conocimiento.

**Fase Inicial:** Portal de la gestión del conocimiento apartado carga individual

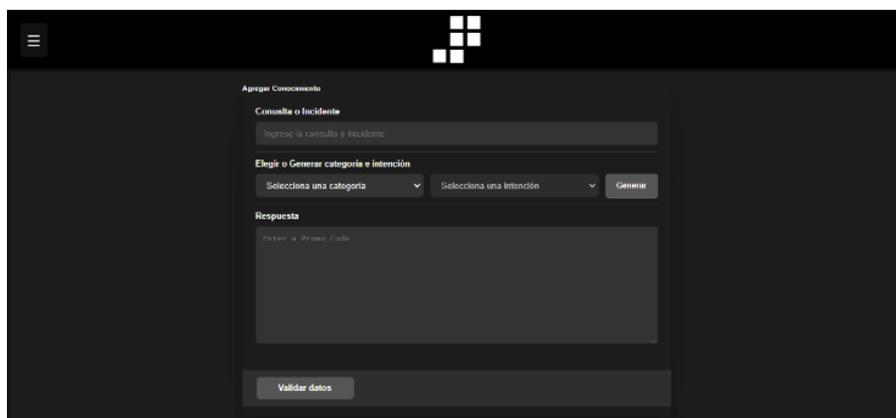


Figura 11 Fase inicial de la interfaz de carga masiva

**Nota:** En esta interfaz el usuario podrá cargar registros de forma individual, ingresando una consulta o incidente, generando su clasificación e ingresando su respuesta.

**Fase alternativa:** En el caso de existir un registro similar

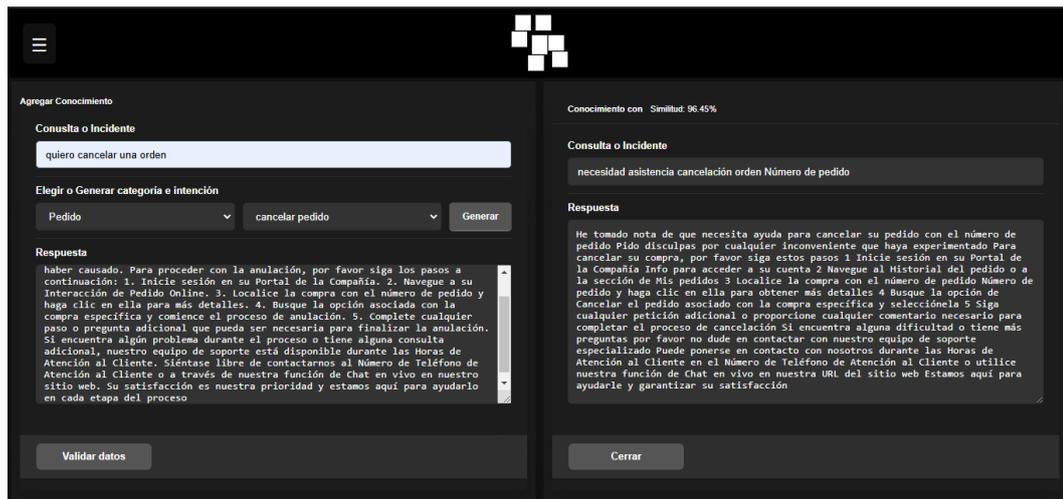


Figura 12 Fase final en el caso de existir un registro con similitud

**Nota:** En el caso de existir un registro con una similitud mayor al 95% se presenta un modal en la que se puede presentar mediante por pantalla el registro con similitud.

**Fase Inicial:** Portal de la gestión del conocimiento apartado carga masiva

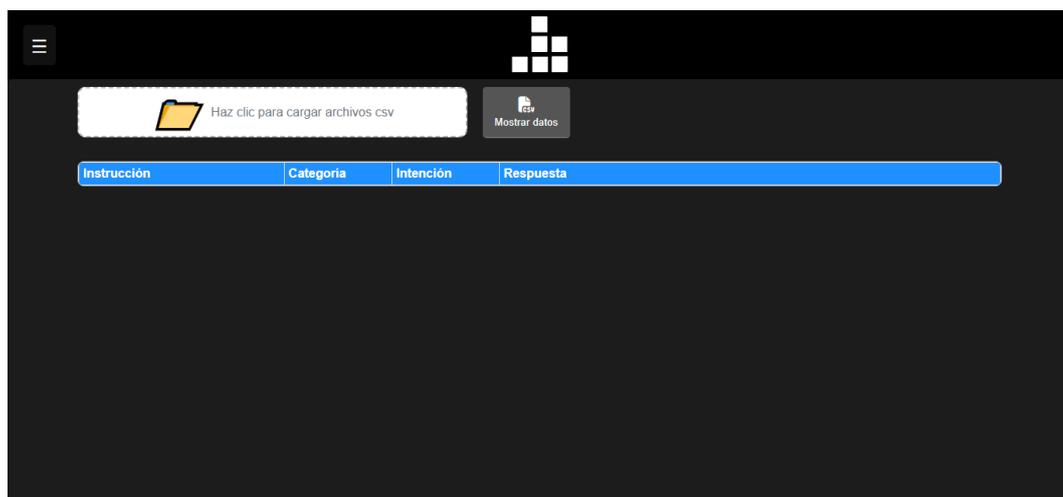


Figura 13 Fase inicial la interfaz de carga masiva

**Nota:** En la interfaz carga masiva se puede cargar archivos CSV con las cabeceras instruction, category, intent y response o con las cabeceras instruction y response con estos campos puede acondicionar el archivo para poder ingresar sus datos a la base datos.

### Fase validación

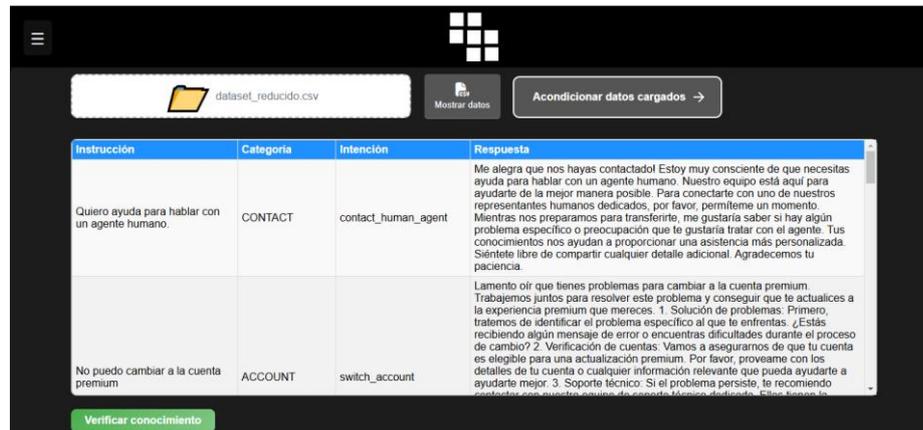


Figura 14 Fase luego del acondicionamiento de datos

**Nota:** La tabla es cargada con los registros luego de cargarlos y acondicionarlos, para evidenciar de forma visual los datos que se han cargado, con el botón validar conocimiento comienza el proceso de la validación de cada registro del archivo con los de la base de datos, mediante la comparación semántica de los campos en la columna response.

**Fase Inicial:** Portal de la gestión del conocimiento apartado actualizar conocimiento



Figura 15 Fase inicial de la interfaz de actualización

**Nota:** En esta parte de la interfaz, el usuario puede elegir el registro a modificar y realizar un filtrado de los datos por categoría, subcategoría, campo de instrucción o respuesta. Asimismo, se incluyen opciones adicionales para previsualizar los datos completos o su versión resumida.

### Fase de actualización

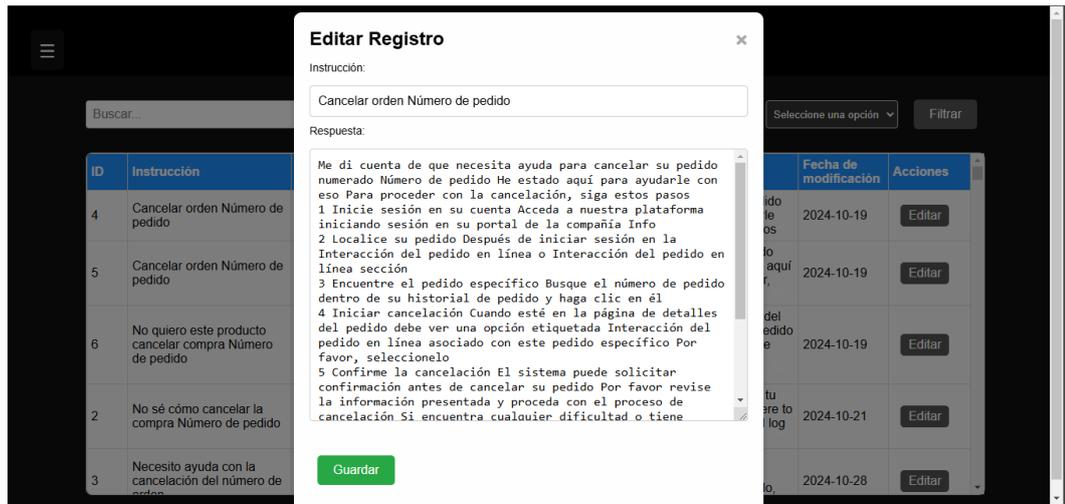
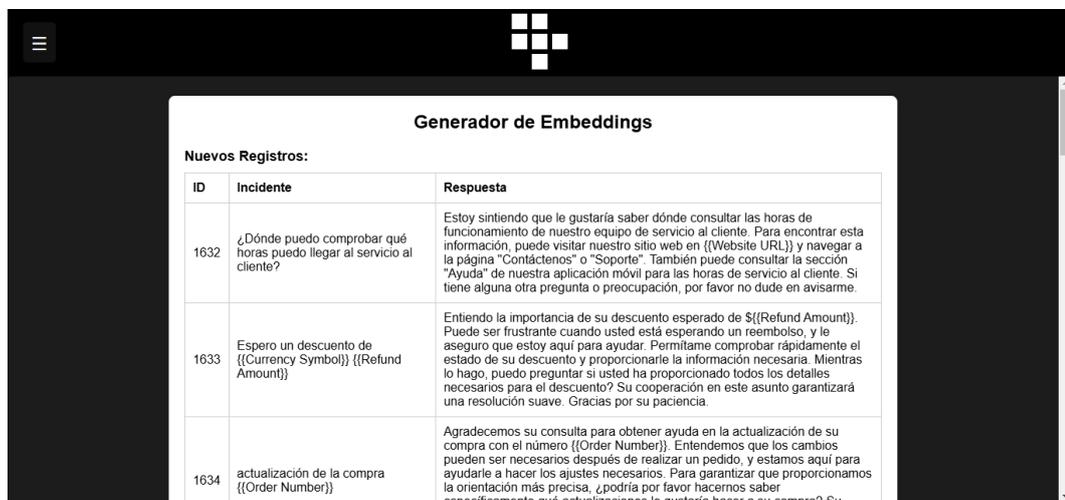


Figura 16 Modal para la actualización de registros

**Nota:** En esta parte luego de elegir el registro a editar se presenta el modal en el cual se puede modificar la instrucción, la respuesta o ambos.

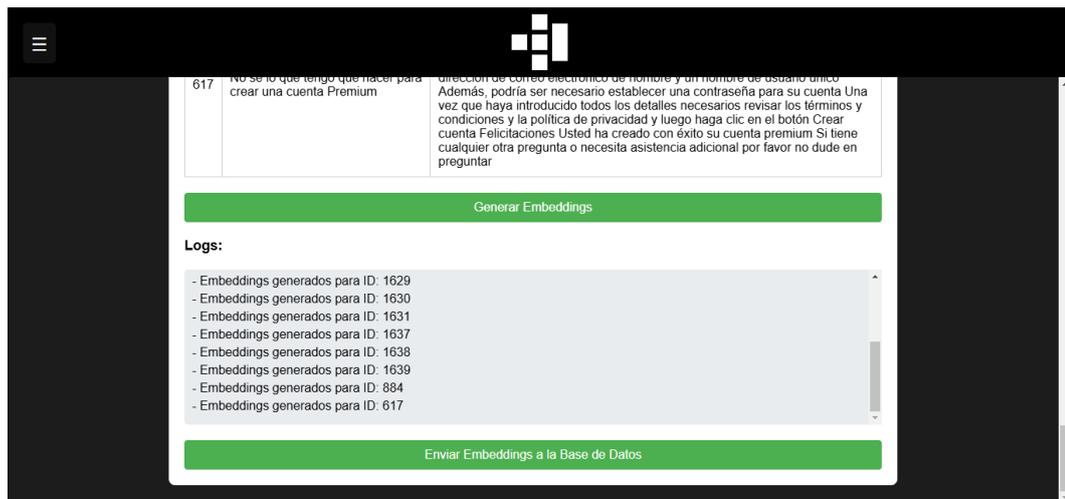
**Fase Inicial:** Portal de la gestión del conocimiento apartado generación de embeddigns.



*Figura 17 Fase inicial de la interfaz de generación de embeddings*

**Nota:** En esta interfaz al cargarla se carga de forma automática los registros que necesiten generar embeddings, ya sea el caso de ingresar nuevos embeddings o actualizarlos, estos embeddings son almacenados en la base de datos en formato hexadecimal.

### **Fase de visualización de embeddings generados**



*Figura 18 Fase de visualización de los registros a los que se le genero los embeddings*

**Nota:** En esta fase de la interfaz se muestran los ID de los registros a los cuales se generaron los embeddings tanto para los registros que necesitan actualización como para los nuevos registros.

### **2.5.3. Desarrollo del prototipo**

#### **Desarrollo y Codificación de la Aplicación**

En esta sección se presentan las principales funciones con los componentes claves para el desarrollo del sistema que permiten la implementación de los objetivos del proyecto. Cada funcionalidad fue diseñada para abordar una necesidad específica del sistema mediante la integración de tecnologías de procesamiento de lenguaje natural (PLN), bases de datos y una interfaz web para la gestión del conocimiento.

## **Modelo clasificador**

En este apartado se llevó a cabo una búsqueda de hiperparámetros utilizando GridSearchCV para encontrar los mejores valores de ciertos parámetros clave de un modelo de Máquinas de Soporte Vectorial (SVM). Este proceso de ajuste permite optimizar el rendimiento del modelo, probando diferentes combinaciones de parámetros como el valor de **C**, el **gamma** y el tipo de **kernel**, al finalizar la búsqueda, se obtuvo una precisión del 96.56% (Ver anexo 1).

## **Estrategias para optimizar la accesibilidad de la información**

Para abordar la comparación semántica en el sistema, se implementaron y evaluaron diversas estrategias. Inicialmente se probó con la similitud del coseno (cosine similarity) calculada sobre embeddings generados TF-IDF. Posteriormente, se probó un enfoque basado en el uso de clustering para agrupar las consultas mediante el uso del algoritmo K-means. Finalmente, se generó embeddings con modelos que soporten el idioma español para compararlos utilizando nuevamente la similitud del coseno, observando diferencias en el rendimiento según el enfoque utilizado (Ver anexo 2).

**Estrategia de combinar función cosen más embeddings TF-IDF:** Durante la evaluación de esta estrategia, se observó que el modelo basado en TF-IDF no maneja adecuadamente los sinónimos contextuales, como es el caso de los términos "activar" y "habilitar". Esto se debe a que TF-IDF se basa en la frecuencia de aparición de las palabras en un corpus, sin tener en cuenta el contexto semántico o la relación de sinónimos entre ellas.

**Estrategia clustering más K-means:** Durante la evaluación de esta estrategia se observó que para el funcionamiento correcto del algoritmo K-means es necesario definir previamente el número de clústeres que se desean generar. Este parámetro es crucial para que el algoritmo agrupe los datos de manera efectiva la dependencia de un número predefinido de clústeres plantea un desafío importante al intentar automatizar el proceso.

**Estrategia embeddings modelos en español más coseno:** En la evaluación de esta estrategia se utilizaron los modelos Flan-T5 y bert-base-spanish para generar

embeddings y calcular similitudes mediante la función de coseno. A través de los experimentos realizados, se logró evidenciar que ambos modelos presentan diferencias significativas en su comprensión semántica, especialmente en relación con las variaciones léxicas de las palabras dentro de un contexto dado.

Basado en las comparativas realizadas se concluyó que la combinación de la generación de embeddings con un modelo BERT que soporte el idioma español y la similitud del coseno resultó ser la opción más efectiva ya que ofreció una comparación semántica precisa y optimizada, lo que permitió seleccionarla como la más adecuada para las necesidades del sistema.

### **Función para generar la clasificación usando el modelo SVM (Ver anexo 3)**

Flujo de la función

1. Se deben cargar el modelo y el encoder previamente guardados.
2. Se recibe la consulta en formato JSON y se valida que esta no este vacía.
3. Se hace la predicción
  - 3.1. La variable modelo\_cargado hace la predicción sobre la consulta proporcionada.
  - 3.2. La predicción resultante es codificada (como un número).
  - 3.3. Se usa la variable encoder para convertir la etiqueta predicha de vuelta a su valor original (como una clase).
4. Se genera una predicción, esta se devuelve como una respuesta JSON.

### **Función para la comparación semántica tanto para incidentes como validación de registros (Ver anexo 4)**

Flujo para manejo de consulta del usuario

1. La consulta recibe un JSON con los parámetros query (consulta del usuario) y intent (intención asociada a la consulta).
2. Se valida que tanto el parámetro query e intent estén presentes en el JSON.
3. Se hace uso de la función “handle\_query\_optimized” para retornar un array “response\_data” que tiene los parámetros “intent” la intención de la consulta del usuario y “respuesta” los 3 registros con mayor similitud semántica.

- 3.1. Se genera un array que consta de los siguientes datos: id, instruction, response, ranking de los registros con mayor similitud.
- 3.2. El apartado de **ranking** se presenta como un sub-array dentro del array principal, donde cada registro contiene los valores de similitud asociados a dicho registro
4. Validación que el array “response\_data” no este vacío. Si está vacío, se retorna una respuesta con error 404 indicando que no se encontraron respuestas similares a la consulta.
5. Devolución de la respuesta en formato JSON con los parámetros intent y respuesta.

#### Flujo para manejo similitudes con nuevos registros

1. La consulta recibe un JSON con los parámetros query (consulta del usuario) y intent (intención asociada a la consulta).
2. Se valida que tanto el parámetro query e intent estén presentes en el JSON.
3. Se hace uso de la función “handle\_query\_optimized\_respuesta” para retornar un array que tiene los parámetros “respuesta”: response\_data, (contiene la respuesta que presento similitud con el registró a ingresar) y “similitud”: similitud\_data (contiene el porcentaje similitud que tiene con el registro a ingresar).
4. Validación que el array “response\_data” no este vacío. Si está vacío, se retorna una respuesta con error 404 indicando que no se encontraron respuestas similares a la consulta.
5. Devolución de la respuesta en formato JSON con los parámetros intent y respuesta.

#### **Funciones para la creación o actualización de embeddings (Ver anexo 5)**

##### Flujo de la función generar\_embeddings

1. Se reciben 2 parámetros “instruction” (la instrucción del registro) y “response” (la respuesta del registro) ambos como cadenas de texto.
2. Genera los embeddings para el parámetro instruction y response

- 2.1. Se utiliza la variable `modelo_similitd` para generar el embedding del parámetro “instruction”.
- 2.2. Se utiliza la variable `modelo_similitd` para generar el embedding del parámetro “instruction”.
- 2.3. Se convierte el embeddings a arreglos del tipo `numpy`.
3. Se retorna ambos embeddings en un diccionario con las claves `instruction_embedding` y `response_embedding`.

#### Flujo de la función `insertar_embedding`

1. Se reciben 4 parámetros “cursor” (objeto utilizado para interactuar con la base de datos), “consulta\_id” (id del registro), “instruction\_embedding” (embedding generado para la instrucción del registro) y “response\_embedding” (embedding generado para la respuesta del registro).
2. Se asegura que las variables “instruction\_embedding” y “response\_embedding” sean arreglos `numpy` de tipo `float32`.
3. Las variables “instruction\_embedding” y “response\_embedding” se convierten a formato binario usando el método `tobytes()`.
4. Se insertan los nuevos registros en la tabla `embeddings`.
5. Si la inserción es exitosa, se confirma con un mensaje indicando que la inserción se completó para el `consulta_id` especificado.

#### Flujo de la función `insertar_embedding`

1. Se reciben 4 parámetros “cursor” (objeto utilizado para interactuar con la base de datos), “consulta\_id” (id del registro), “instruction\_embedding” (embedding generado para la instrucción del registro) y “response\_embedding” (embedding generado para la respuesta del registro).
2. Se asegura que las variables “instruction\_embedding” y “response\_embedding” sean arreglos `numpy` de tipo `float32`.
3. Las variables “instruction\_embedding” y “response\_embedding” se convierten a formato binario usando el método `tobytes()`.

4. Se actualizan los registros la tabla embeddings mediante el uso del consulta\_id que es clave forane en la tabla embeddings y además se actualiza el campo “needs\_update” a False en la tabla “Instrucciones-Consultas”.
5. Si la inserción es exitosa, se confirma con un mensaje indicando que la inserción se completó para el consulta\_id especificado.

### **Integración de Datos**

En esta sección se realizó un análisis comparativo de cinco datasets obtenidos de distintos sitios web, cada uno con características únicas y enfoques variados en la organización de datos. El objetivo principal fue identificar un dataset que cumpliera con los criterios necesarios, es decir, que presentara consultas y respuestas bien estructuradas, organizadas de manera coherente y adecuadas para el propósito del proyecto.

Finalmente, se seleccionó el dataset más adecuado, cumpliendo con los requisitos definidos para el desarrollo del proyecto. Los detalles de los datasets analizados durante este proceso, junto con sus características principales, se presentan en una tabla incluida en el (Ver Anexo 6).

El dataset seleccionado fue “bitext/Bitext-customer-support-llm-chatbot-training-dataset”, ya que cumple con los requisitos de incluir una amplia variedad de respuestas estructuradas. Además, cuenta con una clasificación para las consultas, lo que facilita una recuperación de información más eficiente y precisa.

### **Acondicionamiento del dataset**

Este dataset consta de 26,872 registros que incluyen tanto respuestas planas como estructuradas. Para acondicionarlo se realizó un filtrado para separar respuestas planas de las estructuradas obteniendo dos conjuntos de datos diferenciados: uno con 6,177 registros que contienen respuestas estructuradas y otro con 20,695 registros de respuestas planas (Ver anexo 7). Después de obtener los dos datasets se seleccionó el dataset con respuestas estructuradas para traducirlo del inglés al español la traducción se realizó utilizando el modelo preentrenado “Helsinki-NLP/opus-mt-en-es”, diseñado específicamente para traducir entre estos idiomas.

Generación de Datos Sintéticos usando ChatGPT

Para aumentar la diversidad y representatividad de los datos en el sistema y validar la comparación semántica, se empleó una estrategia mixta. Inicialmente, se cargó un archivo CSV que contenía preguntas y respuestas frecuentes categorizadas. Posteriormente se utilizó la versión GPT-4<sup>o</sup> mini de ChatGPT para generar registros adicionales creando variaciones temáticas similares a los datos cargados. Estos nuevos registros se diseñaron específicamente para probar y evaluar la efectividad de las estrategias de comparación semántica implementadas en el sistema, garantizando resultados más precisos y representativos (Ver anexo 8).

Por ejemplo, una de las instrucciones proporcionadas fue:

Crea una nueva columna que haga referencia a una consulta ingresada por el usuario, los datos de esta columna los debes generar usando la columna "instruction" que tiene el dataset para que en base a esos registros generes registros similares.

Los datos generados incluyeron variaciones similares a los registros existentes pero reformulados manteniendo la intención inicial. Esto significa que las nuevas formulaciones no solo reflejaron el contenido semántico de las instrucciones originales, sino que también buscaron diversificar el lenguaje utilizado, preservando la claridad y coherencia del mensaje.

<b>Instrucción original</b>	<b>Instrucción reformulada por el modelo GPT</b>
Quiero saber cómo activar mi cuenta	Deseo entender de qué forma habilitar mi perfil
¿Qué hago si olvidé mi contraseña?	¿Qué pasos debo seguir si he perdido mi clave?
Ayuda para rastrear mi pedido	Necesito asistencia para localizar mi compra
Necesito anular la suscripción al servicio	Preciso cancelar la afiliación al sistema
No sé cómo devolver un artículo	No entiendo cómo retornar un producto

*Tabla 12 Ejemplo de datos generados con el modelo GPT*

### **Proceso de Integración**

Los datos generados sintéticamente se utilizaron específicamente para las pruebas de comparación semántica dentro del proyecto. En primer lugar, sirvieron como un

recurso clave para evaluar el desempeño del modelo clasificador al manejar variaciones temáticas y sinónimos, asegurando una capacidad de generalización robusta. Además, estos datos se emplearon en el proceso de selección de la mejor estrategia para optimizar consultas a la base de datos, proporcionando un entorno controlado para validar la eficacia de las métricas y técnicas implementadas, como la similitud semántica y el manejo de sinónimos.

Inicialmente, los datos se cargaron en la base de datos utilizando archivos CSV, los cuales se importaron directamente a la base mediante un proceso automatizado. Este enfoque permitió que la información se integrara de manera rápida y eficiente, minimizando la intervención manual y facilitando la carga de grandes volúmenes de datos.

#### 2.5.4 Simulación y Evaluación

El objetivo de esta etapa es identificar los requisitos funcionales (necesidades del cliente) que se implementarán en el software. Una de las estrategias para lograr este objetivo es el diseño de diagramas de casos de uso; El diagrama de casos de uso es un diagrama uml (lenguaje de modelado unificado) que, a través de una representación gráfica del sistema, permite modelar los requerimientos funcionales de una aplicación, mostrándolos desde el punto de vista del usuario [60].

#### Diagrama de caso de uso general del portal del conocimiento

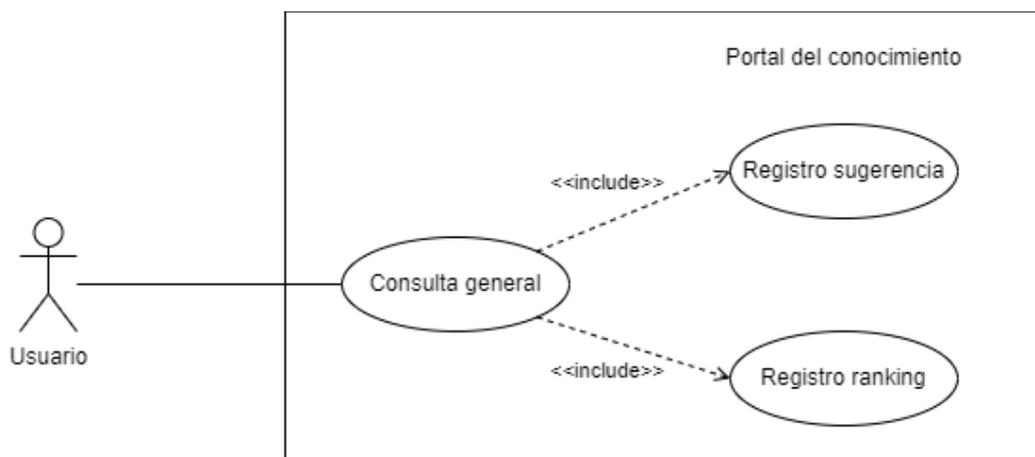
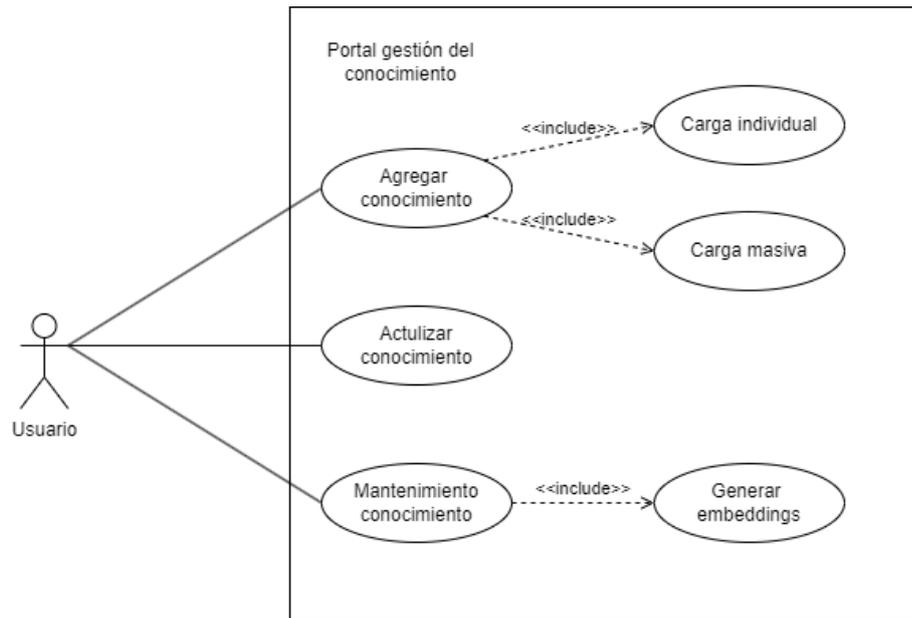


Figura 19 Caso de uso portal del conocimiento

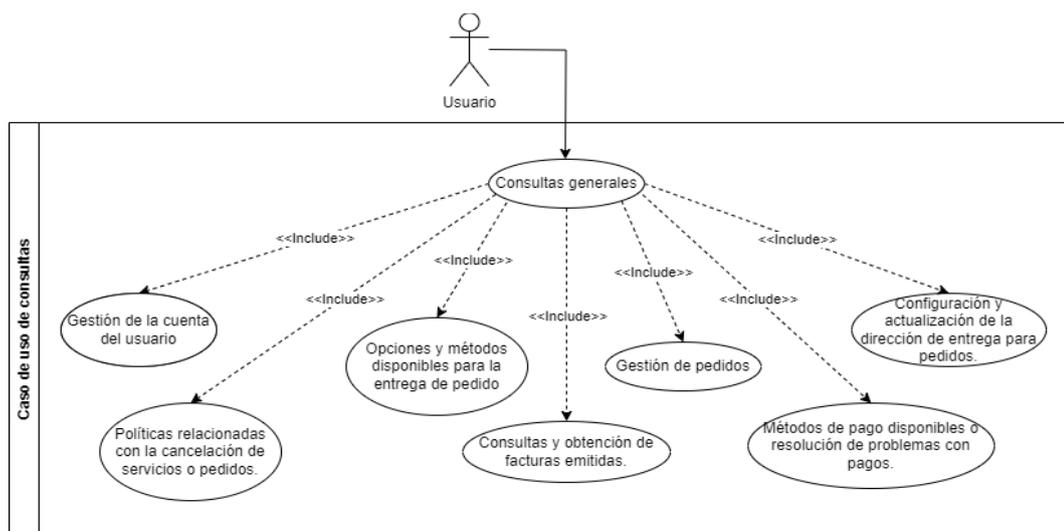
**Nota:** Tanto el proceso registros de sugerencias como los registros de ranking son procesados y gestionados de manera integral dentro del sistema, con el objetivo de optimizar la calidad y relevancia de las respuestas ofrecidas al usuario. Debido a que estos procesos son simples y automáticos, no requieren un caso de uso específico, ya que se manejan de manera eficiente sin intervención compleja.

**Diagrama de caso de uso general del portal gestión del conocimiento**



*Figura 20 Caso de uso portal de gestión del conocimiento*

**Diagrama de caso de uso: usuario consultante de la base de datos**

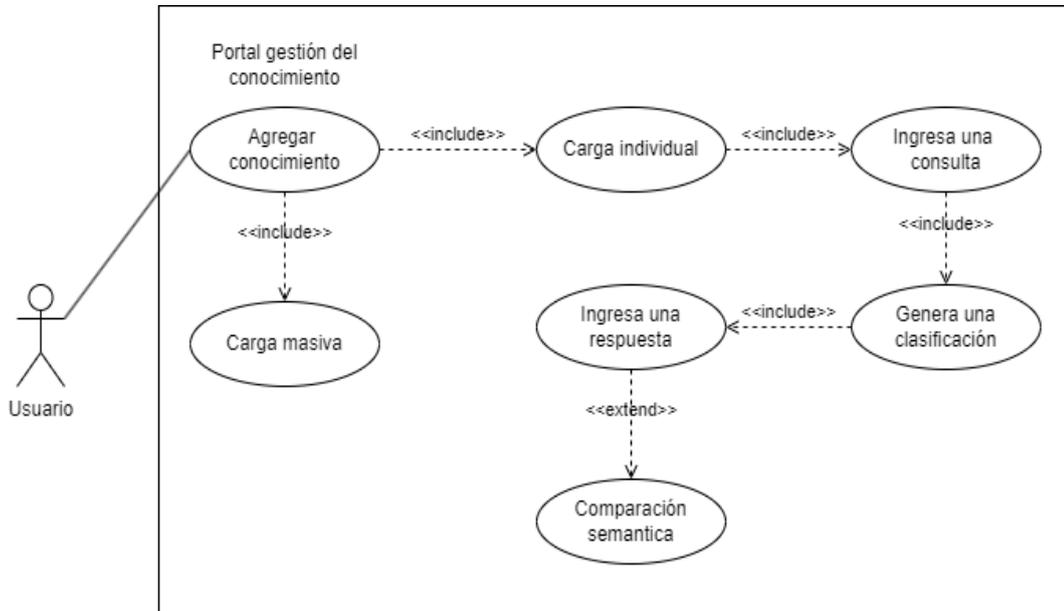


*Figura 21 Caso de uso general para las consultas del usuario*

<b>CASO DE USO DESCRIPCIÓN</b>	<b>01 – CONSULTAS GENERALES</b>
<b>ACTOR</b>	Usuario
<b>EVENTO DESENCADENADOR</b>	El usuario escribe su consulta relacionada con un incidente que está dentro de las categorías.
<b>PRECONDICIÓN</b>	Ninguna
<b>SECUENCIA PRINCIPAL</b>	<ol style="list-style-type: none"> <li>1. El usuario puede consultar sobre cómo gestionar su cuenta.</li> <li>2. El usuario puede consultar las políticas relacionadas con la cancelación de servicios o pedidos.</li> <li>3. El usuario puede consultar opciones y métodos de pago disponible.</li> <li>4. El usuario puede consultar como visualizar sus facturas en la plataforma.</li> <li>5. El usuario puede consultar como gestionar un pedido.</li> <li>6. El usuario puede consultar sobre como recibiría un reembolso.</li> <li>7. El usuario puede consultar sobre cómo gestionar su información de entrega para sus pedidos.</li> </ol>
<b>ERRORES / ALTERNATIVAS</b>	<p>ER – 01 El campo 'consulta' es obligatorio.</p> <p>ER – 02 Consulta no válida. Por favor, intente con una consulta más específica.</p> <p>ER – 03 Hubo un problema procesando la consulta. Intente más tarde.</p>
<b>POSTCONDICIÓN</b>	Se presentan 3 cards con las respuestas relevantes basadas en similitudes y ranking promedio.

*Tabla 13 Caso de uso consultas generales*

**Diagrama de caso de uso para ingresar un nuevo conocimiento de forma individual.**



*Figura 22 Caso de uso para ingresar un registro de forma individual*

**CASO DE USO DESCRIPCIÓN**

**02 – CARGA INDIVIDUAL**

	El usuario puede ingresar un registro que conste de una instrucción y una respuesta
ACTOR	Usuario
EVENTO DESENCADENADOR	El usuario escribe la instrucción del nuevo registro.
PRECONDICIÓN	La instrucción debe estar dentro de las categorías de la base de datos.
SECUENCIA PRINCIPAL	<ol style="list-style-type: none"> <li>1. El usuario ingresa la instrucción del nuevo registro.</li> <li>2. El usuario genera de forma automática o selecciona una clasificación para la instrucción.</li> <li>3. El usuario ingresa la respuesta a la instrucción.</li> <li>4. El usuario genera la comparación semántica.</li> <li>5. Respuesta:             <ol style="list-style-type: none"> <li>5.1. En el caso de existir registros se presenta un modal con el porcentaje de similitud y un botón para ver el registro con similitud.</li> </ol> </li> </ol>

	5.2. En el caso de no existir se presenta un modal para confirma el ingreso del nuevo registro.
ERRORES / ALTERNATIVAS	ER – 01 El campo “incidente” es obligatorio. ER – 02 Elija o genere una clasificación para el incidente. ER – 03 El campo “respuesta” es obligatorio.
POSTCONDICIÓN	Se presenta un modal dependiendo del resultado.

Tabla 14 Caso de uso para proceso de carga individual

**Diagrama de caso de uso para ingresar un nuevo conocimiento de forma masiva.**

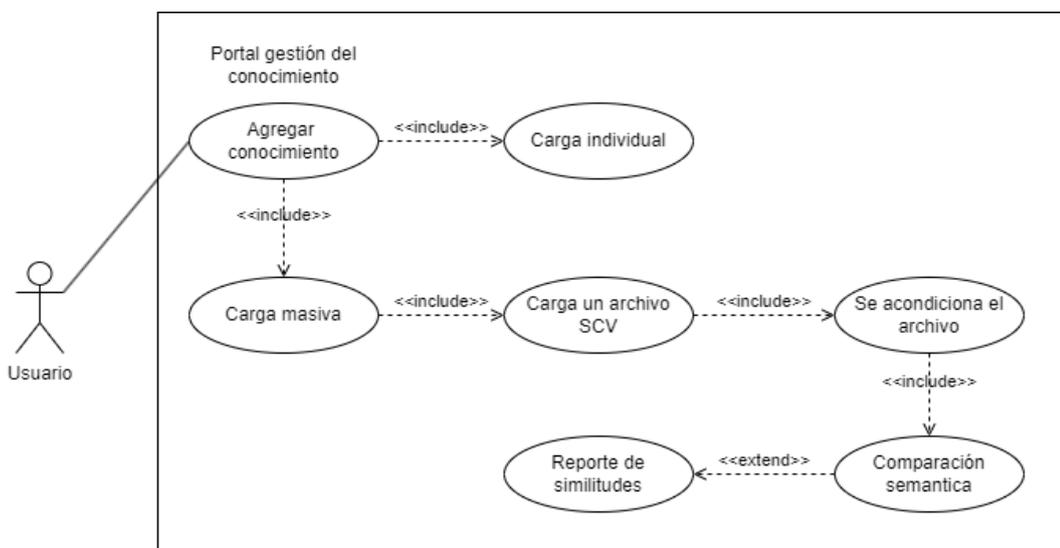


Figura 23 Caso de uso para ingresar varios registros

<b>CASO DE USO</b>	<b>03 – CARGA MASIVA</b>
<b>DESCRIPCIÓN</b>	El usuario puede ingresar un archivo con extensión .csv que conste con las columnas “instruction”, “category”, “intent” y “response” o solo con las columnas “instrucion” y “response”.
<b>ACTOR</b>	Usuario
<b>EVENTO DESENCADENADOR</b>	El usuario carga un archivo con extensión .csv.

PRECONDICIÓN	El archivo de constar con una cabecera con al menos las columnas “instruction” y “response”.
SECUENCIA PRINCIPAL	<ol style="list-style-type: none"> <li>1. El usuario carga un archivo csv.</li> <li>2. El usuario acondiciona el archivo cargado mediante un botón.</li> <li>3. El genera la comparación semántica de todos los registros del archivo.</li> <li>4. Se presenta un modal en el que se permite agregar los registros que no presentaron similitud y ver el reporte de los registros que presentaron similitud.</li> </ol>
ERRORES / ALTERNATIVAS	ER – 01 El archivo no consta con los requerimientos mínimos.
POSTCONDICIÓN	Se presenta un modal para realizar confirmar el envío de los registros a la base de datos.

*Tabla 15 Caso de uso para realizar una carga masiva*

## 2.6 Pruebas

Es de suma importancia verificar el correcto funcionamiento de un proyecto informático, especialmente cuando se trata de un sistema web al que accederá un número determinado de personas al día, buscando un rendimiento óptimo y de calidad del sistema. Por esta razón, se llevaron a cabo una serie de pruebas mencionadas a continuación para garantizar su eficacia:

- Verificar que el sistema pueda clasificar correctamente las consultas del usuario y hacer la comparación semántica adecuada con los registros de la base de datos.
- Validar que el sistema pueda manejar correctamente la carga de archivos CSV válidos y que los datos se procesen correctamente.
- Verificar que los registros existentes puedan ser actualizados de manera efectiva sin pérdida de datos.
- Asegurar que la generación de embeddings y actualización de los mismos funcionen correctamente tanto para registros nuevos como actualizados.

### INFORMACIÓN DEL CASO DE PRUEBA

<b>CASO DE PRUEBA N°</b>	<b>DE</b>	01
<b>CASO DE USO</b>		Realizar una consulta en el portal del conocimiento
<b>OBJETIVO DE LA PRUEBA</b>		Comprobar que el sistema pueda clasificar correctamente las consultas del usuario y hacer la comparación semántica adecuada.
<b>ROL</b>		Usuario general

#### PASOS DE LA PRUEBA

DATOS ENTRADA	Datos Salida
<ol style="list-style-type: none"> <li>1. El usuario ingresa su duda o incidente en la barra de la interfaz.</li> <li>2. El sistema genera su clasificación.</li> <li>3. El sistema con la clasificación hace uso de una función para obtener datos de esa clasificación.</li> <li>4. El sistema genera el embedding de la consulta ingresada por el usuario.</li> <li>5. El sistema realizara la comparación semántica.</li> </ol>	<ol style="list-style-type: none"> <li>1. La consulta a la base de datos debe recuperar los campos relevantes de las siguientes tablas para garantizar la integración y funcionalidad del sistema: T. instrucciones-respuesta: id, instruction, response, category, intent. T. embedding: instruction_embedding, response_embedding T. ranking: ranking</li> <li>2. El embedding generado para la consulta del usuario deberá ser de 384 dimensiones.</li> <li>3. La comparación semántica se hará con el embedding generado y los embeddings extraídos de la base de datos.</li> </ol>

#### RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Evaluación
El portal del conocimiento hace correctamente la clasificación y comparación semántica con los campos adecuados.	<b>Exitoso: ✓</b> <b>Fallido:</b>

*Tabla 16 Caso de prueba de comparación semántica consulta*

### INFORMACIÓN DEL CASO DE PRUEBA

<b>CASO DE PRUEBA N°</b>	<b>DE</b>	02
<b>CASO DE USO</b>		Realizar la carga masiva de registros.
<b>OBJETIVO DE LA PRUEBA</b>		Comprobar que el sistema pueda manejar correctamente la carga de archivos CSV válidos y que los datos se procesen correctamente.
<b>ROL</b>		Usuario gestor del conocimiento

PASOS DE LA PRUEBA	
DATOS ENTRADA	Datos salida
<ol style="list-style-type: none"> <li>1. El usuario carga un archivo csv</li> <li>2. El usuario acondiciona el archivo.</li> <li>3. El usuario valida los registros del archivo.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema reconoce si al archivo le hace falta las columnas “category” e “intent” para generarlas.</li> <li>2. El sistema toma cada registro en la posición de la columna “intent” para realizar la consulta a la base de datos.</li> <li>3. La consulta a la base de datos debe recuperar los campos relevantes de las siguientes tablas para garantizar la integración y funcionalidad del sistema: T. instrucciones-respuesta: id, instruction, response, category, intent. T. embedding: instruction_embedding, response_embedding</li> <li>4. El Sistema genera los embeddings para cada valor en la posición de la columna “response” de cada registro, estos embeddings deben ser de 768 Dimensiones.</li> <li>5. La comparación semántica se la realiza con el embedding generado para el valor de la columna “response” y el valor de “response_embedding” de cada registro obtenido de la consulta.</li> </ol>

#### RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Evaluación
El apartado de carga masiva del portal de gestión del conocimiento, realiza el preprocesamiento del archivo correctamente junto con la comparación semántica.	<b>Exitoso: ✓</b> <b>Fallido:</b>

*Tabla 17 Caso de prueba comparación semántica respuesta*

#### INFORMACIÓN DEL CASO DE PRUEBA

<b>CASO DE PRUEBA N°</b>	DE 03
<b>CASO DE USO</b>	Realizar actualización del conocimiento.
<b>OBJETIVO DE LA PRUEBA</b>	Comprobar que los registros existentes puedan ser actualizados de manera efectiva sin pérdida de datos.

<b>ROL</b>	Usuario gestor del conocimiento	
<b>PASOS DE LA PRUEBA</b>		
<b>DATOS ENTRADA</b>	<b>Datos salida</b>	
<ol style="list-style-type: none"> <li>1. El usuario elige el registro a modificar.</li> <li>2. El usuario decide si modificar la instrucción, la respuesta o ambas.</li> <li>3. El usuario realiza las modificaciones.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema solo modificara el campo “instruction” o “response” según sea el caso.</li> <li>2. El sistema al actualizar el registro cambia la columna “needs_update” a True.</li> </ol>	
<b>RESULTADOS DE LA PRUEBA</b>		
<b>RESULTADOS OBTENIDOS</b>	<b>Evaluación</b>	
El apartado de actualizar conocimiento del portal de gestión del conocimiento, realiza las actualizaciones de los campos correspondientes.	<b>Exitoso: ✓</b>	<b>Fallido:</b>

*Tabla 18 Caso de prueba actualización de registros*

<b>INFORMACIÓN DEL CASO DE PRUEBA</b>		
<b>CASO DE PRUEBA N°</b>	<b>DE</b>	04
<b>CASO DE USO</b>	Generación y almacenamiento de embeddigns	
<b>OBJETIVO DE LA PRUEBA</b>	Comprobar que la generación de embeddings y la actualización de los mismos funcionen correctamente tanto para registros nuevos como actualizados.	
<b>ROL</b>	Usuario gestor del conocimiento	
<b>PASOS DE LA PRUEBA</b>		
<b>DATOS ENTRADA</b>	<b>Datos salida</b>	
<ol style="list-style-type: none"> <li>1. El ingresa en la interfaz de generar embeddigns del portal de gestión del conocimiento.</li> <li>2. El usuario genera los embeddings.</li> <li>3. El usuario almacena o actualiza los embeddings.</li> </ol>	<ol style="list-style-type: none"> <li>1. El sistema al cargar hace una búsqueda de los registros que no se encuentren en la tabla “embeddings” y valida que los registros que estén en la tabla principal en el campo “needs_update” sean True para presentarlos en la interfaz.</li> <li>2. El sistema genera los embeddings correspondientes para el campo instruction con el modelo “sentence-transformers/all-MiniLM-L6-v2” y el campo response con el modelo “dccuchile/bert-base-spanish-wwm-uncased”.</li> </ol>	

	<p>3. El sistema valida que campo son de actualizar o de agregar con la función “existe_registro”.</p> <p>4. EL sistema actualiza el campo “needs_update” a False de los registros de los cuales sus embeddings fueron actualizados.</p>
--	--

### RESULTADOS DE LA PRUEBA

RESULTADOS OBTENIDOS	Evaluación
El apartado de generar embeddings del portal de gestión del conocimiento, realiza los registros y actualizaciones correspondientes.	<p><b>Exitoso: ✓</b></p> <p><b>Fallido:</b></p>

*Tabla 19 Caso de prueba de uso generar o actualizar embeddings*

## 2.7 Resultados

### 2.7.1 Análisis de resultados

Después de analizar los principales desafíos que se enfrentan al implementar la gestión de conocimientos en departamentos de TIC, se identificó que los tiempos de respuesta prolongados y la duplicación de esfuerzos son los problemas más comunes. Esta situación llevó al diseño de una interfaz web que integra algoritmos de procesamiento de lenguaje natural (NLP) para facilitar el acceso y la recuperación de información relevante en entornos de soporte técnico, optimizando así la resolución de incidentes recurrentes.

Posteriormente se llevaron a cabo pruebas de validación para evaluar el desempeño de la interfaz web comparando la precisión en la recuperación de información con los métodos manuales utilizados previamente. Los resultados se documentaron en un análisis comparativo que evidencia cómo la solución automatizada propuesta reduce significativamente los tiempos de tanto de búsqueda como de mejora de la eficiencia operativa en la gestión de conocimientos.

### ANÁLISIS DE PRUEBAS REALIZADAS

DIFICULTADES HALLADAS	Solución aplicada
La inexistencia de una plataforma para gestionar el conocimiento puede derivar en varios inconvenientes, como	El desarrollo de una interfaz web ajustada a un modelo de administración del conocimiento facilitará la

<p>el tiempo invertido por los colaboradores en la búsqueda o recuperación de conocimiento que ya existe, pero que es difícil de localizar.</p>	<p>concentración de la información en un solo portal accesible a través de consultas en lenguaje natural. Esta solución disminuirá considerablemente el tiempo que los empleados dedican a la búsqueda de información, proporcionando resultados exactos fundamentados en técnicas de procesamiento de lenguaje natural (NLP).</p>
<p>Al realizar búsquedas en amplias bases de datos, los hallazgos proporcionan cientos de informes lo que resulta en una labor que demanda mucho tiempo identificar los documentos de mayor relevancia.</p>	<p>La aplicación de comparación semántica analiza las solicitudes de los usuarios además de conectarlas con los registros de la base de datos facilitando así la adquisición de la información más relevante en menos de 10 segundos. Este procedimiento no solo acelera la búsqueda, sino que también garantiza resultados precisos y contextualizados, reduciendo el tiempo requerido para analizar cientos de informes.</p>

*Tabla 20 Análisis de pruebas*

### **2.7.2 Resultados finales**

En este proyecto de titulación se desarrolló un prototipo de interfaz web que implementa un modelo de gestión del conocimiento compuesto por tres capas: datos, gestión del conocimiento y presentación. Este enfoque permite la centralización de la información. A continuación, se presentan los resultados obtenidos.

Las estrategias de recolección de información se dirigieron a los técnicos de soporte y al personal encargado de la gestión del conocimiento, permitiendo así establecer y definir los requisitos funcionales y no funcionales del sistema web que facilitará la centralización y recuperación eficiente de la información.

El sistema web se desarrolló siguiendo una arquitectura cliente-servidor y la del modelo de gestión del conocimiento Kerschberg, se aplicó modelos bert que están basados en Transformers para la comparación semántica al momento de interactuar con la base de datos. Así mismo diseño 5 interfaces las cuales están ubicadas en cada capa de la arquitectura del modelo de gestión del conocimiento, Se

desarrollaron en HTML versión 5 y JavaScript para la parte del Front-End de la interfaz, mientras que la parte del Back-End está construida con el framework Django en Python 3.10.12, lo que permite manejar la lógica del servidor, la interacción con la base de datos y la implementación de las funcionalidades de la aplicación web.

Por último, se llevaron a cabo pruebas de funcionalidad para asegurar el correcto desempeño del sistema web, las cuales se fundamentaron en los casos de uso previamente establecidos, por lo que estas pruebas confirmaron un rendimiento óptimo y adecuado del sistema, reflejando resultados satisfactorios en su totalidad.

## CONCLUSIONES

- Se reviso varios modelos de gestión del conocimiento cuyos motores de la gestión del conocimiento eran procesos humanos y procesos en las TIC, de los cuales se centró la búsqueda en aquellos cuyo motor sean los procesos de TIC. Esto se debió a que dichos modelos presentan arquitecturas sencillas de automatizar, lo que facilita la implementación de sistemas automatizados para la gestión y recuperación del conocimiento.
- De los modelos de gestión del conocimiento basados en Tecnologías de la Información y Comunicación (TIC), se seleccionó el modelo Kerschberg debido a su arquitectura en capas, que facilita la integración de tecnologías de procesamiento de lenguaje natural (NLP). En particular, se aplicaron técnicas NLP en la capa de gestión del conocimiento, específicamente en los procesos de refinamiento, distribución y presentación de la información, así como en la capa de presentación, que abarca el portal de conocimiento. Esta estructura permitió optimizar la interacción con la base de datos y mejorar la eficiencia general del sistema, al automatizar la clasificación y recuperación de información relevante.
- Con el objetivo de mejorar la accesibilidad a la información mediante el uso de procesamiento de lenguaje natural, se entrenó un modelo de máquinas de soporte vectorial (SVM) que actúa como clasificador, alcanzando un 96% de precisión en las predicciones. Para la evaluación semántica de las consultas, se emplea una comparación semántica mediante embeddings generados por modelos preentrenados que comprenden el idioma español. Estos embeddings se obtienen a partir de modelos basados en BERT, que aprovechan la arquitectura de atención bidireccional para representar de manera más precisa el contexto de las palabras en las consultas. La similitud entre vectores de características se calcula mediante la función de similitud del coseno implementada en la librería scikit-learn
- El objetivo de generar un dataset sintético basado en incidentes de soporte técnico se alcanzó utilizando el modelo ChatGPT. Al cargar un conjunto de datos reales y aplicar prompts específicos, se generaron registros sintéticos que

simulan casos reales de soporte. Aunque se evaluaron otros modelos como LLaMA, GPT y Meta, la elección de ChatGPT se debió a su accesibilidad y capacidad para generar datos coherentes. Estos datos sintéticos fueron empleados para analizar cómo los modelos gestionan la similitud en las consultas y respuestas.

- Se creó una interfaz web capaz de utilizar procesamiento de lenguaje natural (NLP) centrada en facilitar consultas de texto a una base de conocimientos. Utilizando técnicas como análisis de similitud de texto y clasificación permitiendo al usuario obtener respuestas rápidas. Este enfoque optimiza la interacción con el sistema, mejorando la eficiencia en la recuperación de información relevante.

## RECOMENDACIONES

- Se recomienda realizar pruebas con otros modelos de gestión del conocimiento se los puede hacer con el modelo Wiig que presenta 3 capas principales: creación del conocimiento, codificación del conocimiento y aplicación del conocimiento, otra opción puede ser el modelo Bustelo y Amarilla que presenta 5 capas: captura del conocimiento, almacenamiento del conocimiento, procesamiento del conocimiento, distribución del conocimiento y aplicación del conocimiento. Estos modelos pueden ser automatizados haciendo uso de tecnologías como base de datos inteligentes, procesamiento de lenguaje natural, chatbots inteligentes.
- Para explorar otras estrategias que optimicen la accesibilidad a la información, se podría implementar una base de datos basada en ontologías, la cual estructuraría el conocimiento de manera semántica, permitiendo una recuperación más eficiente y precisa de información. Este enfoque semántico facilitaría la consulta de datos al establecer relaciones explícitas entre entidades y conceptos, lo que eliminaría la necesidad de utilizar un modelo clasificador, dado que la ontología proporcionaría un marco estructurado para la interpretación y búsqueda de información.
- Se sugiere que en el caso de modificación de la estrategia para la comparación semántica, una opción viable sería utilizar un modelo preentrenado que tenga una comprensión robusta del idioma español, y realizar un fine-tuning para especializar el modelo en una tarea específica de comparación semántica. Este proceso implicaría ajustar el modelo a las características particulares de la tarea, utilizando un conjunto de datos etiquetados con información sobre la similitud semántica de los pares de textos. Dicho ajuste permite que el modelo se adapte mejor a las necesidades del dominio específico, optimizando su capacidad para medir relaciones semánticas y mejorar la precisión en las tareas de comparación de textos.
- Este trabajo se puede complementar con la implementación de un sistema basado en chatbot o asistente virtual. Esta transición permitiría una interacción más dinámica y conversacional. Para ello, se podría entrenar un modelo de

lenguaje conversacional. Alternativamente, se podría construir un dataset anotado con placeholders (espacios reservados) para entrenar un modelo que sea capaz de identificar estos placeholders y generar preguntas de seguimiento contextualmente relevantes.

## REFERENCIAS

- [1] A. Canals, «DSAPCE,» Julio 2003. [En línea]. Available: <http://148.202.167.116:8080/xmlui/handle/123456789/3300>. [Último acceso: 07 Septiembre 2024].
- [2] V. H. M. García, D. G. Trujillo y L. M. M. Estrada, «ProQuest,» Febrero 2021. [En línea]. Available: <https://www.proquest.com/openview/bf09afd72a0cfbb921cc96447148a732/1?pq-origsite=gscholar&cbl=1006393>. [Último acceso: 23 Noviembre 2024].
- [3] J. J. V. Matos, M. d. P. L. Molina y G. A. P. Morillo, «Dialnet,» 20 Enero 2017. [En línea]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=6233640>. [Último acceso: 07 Septiembre 2024].
- [4] O. F. E. G. M. J. a. J. M. Ivar Örn Arnarsson, «Sage Journals,» Junio 2021. [En línea]. Available: <https://journals.sagepub.com/doi/full/10.1177/1063293X20982973>. [Último acceso: 21 Noviembre 2024].
- [5] A. Cuno y J. José, «REPOSITORIO ACADÉMICO UPC,» 26 Febrero 2020. [En línea]. Available: <https://repositorioacademico.upc.edu.pe/handle/10757/659990>. [Último acceso: 07 Septiembre 2024].
- [6] D. S. E. A. Trujillo, D. M. S. Gallegos y D. J. C. P. Merlos, «Revista de Desarrollo Sustentable, Negocios, Emprendimiento y Educación,» Mayo 2020. [En línea]. Available:

<https://www.eumed.net/rev/rilcoDS/07/industria-gestion-conocimiento.html>. [Último acceso: 23 Noviembre 2024].

- [7] A. Ramírez Martínez, «Repositorio Institucional BUAP,» 24 Mayo 2024. [En línea]. Available: <https://repositorioinstitucional.buap.mx/items/622ee1b0-e4e3-4108-82d1-04025f52bdb2>. [Último acceso: 21 Noviembre 2024].
- [8] A. Pascal, M. Olivera, P. S. Lapalma y I. Martinelli, «ResearchGate,» Noviembre 2024. [En línea]. Available: [https://www.researchgate.net/publication/385106301\\_Recuperacion\\_de\\_Informacion\\_de\\_Reglamentacion\\_Academica\\_en\\_Espanol\\_utilizando\\_Modelos\\_del\\_Lenguaje\\_Natural](https://www.researchgate.net/publication/385106301_Recuperacion_de_Informacion_de_Reglamentacion_Academica_en_Espanol_utilizando_Modelos_del_Lenguaje_Natural). [Último acceso: 23 Noviembre 2024].
- [9] A. S. Cabezuelo, «Universidad Oberta de Catalunya,» Mayo 2010. [En línea]. Available: <https://openaccess.uoc.edu/handle/10609/2681>. [Último acceso: 23 Noviembre 2024].
- [10] N. A. Vera Yépez, «DSpace de la Universidad Técnica de Babahoyo,» 2023. [En línea]. Available: <http://dspace.utb.edu.ec/handle/49000/14256>. [Último acceso: 24 Septiembre 2024].
- [11] I. Rodríguez Mora, «Repositorio Institucional CESA,» 07 Junio 2023. [En línea]. Available: <https://repository.cesa.edu.co/handle/10726/5205>. [Último acceso: 24 Septiembre 2024].
- [12] C. S. Yáñez1, «SciELO,» 26 Junio 2013. [En línea]. Available: [https://www.scielo.cl/scielo.php?pid=S0718-33052013000300015&script=sci\\_arttext](https://www.scielo.cl/scielo.php?pid=S0718-33052013000300015&script=sci_arttext). [Último acceso: 07 Septiembre 2024].

- [13] M. E. Inácio, «Repositório UEM Monografias,» 01 Agosto 2022. [En línea]. Available: <http://monografias.uem.mz/handle/123456789/3076>. [Último acceso: 11 Septiembre 2024].
- [14] R. D. P. Marroquín, « Business Innova Sciences Revista de Ciencias Empresariales,» 13 Agosto 2024. [En línea]. Available: <https://www.innovasciencesbusiness.org/index.php/ISB/article/view/173>. [Último acceso: 29 Septiembre 2024].
- [15] P. R. Fillottrani y C. M. Keet, «MIT Press Direct,» 01 Octubre 2020. [En línea]. Available: <https://direct.mit.edu/dint/article/2/4/487/94896/KnowID-An-Architecture-for-Efficient-Knowledge>. [Último acceso: 24 Noviembre 2024].
- [16] S. Agarwala, A. Anagawadi y R. M. R. Guddeti, «Scopus,» 2021. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050921011716>. [Último acceso: 24 Noviembre 2024].
- [17] D. R. H. Sampieri, D. C. F. Collado y D. M. d. P. B. Lucio, «Páginas Personales UNAM,» 2018. [En línea]. Available: [https://www.paginaspersonales.unam.mx/app/webroot/files/981/Investigacion\\_sampieri\\_6a\\_ED.pdf](https://www.paginaspersonales.unam.mx/app/webroot/files/981/Investigacion_sampieri_6a_ED.pdf). [Último acceso: Septiembre 13 2024].
- [18] A. R. Bueno, «Dipòsit Digital de la Universitat de Barcelona,» Noviembre 2015. [En línea]. Available: [https://diposit.ub.edu/dspace/bitstream/2445/67618/1/EL\\_m%C3%A9todo\\_Experimental\\_conceptualizaci%C3%B3n.pdf](https://diposit.ub.edu/dspace/bitstream/2445/67618/1/EL_m%C3%A9todo_Experimental_conceptualizaci%C3%B3n.pdf). [Último acceso: 11 Septiembre 2024].
- [19] C. Arbulú, «ResearchGate,» Junio 2023. [En línea]. Available: [https://www.researchgate.net/publication/371491131\\_Concepto\\_de\\_superp](https://www.researchgate.net/publication/371491131_Concepto_de_superp)

oblacion\_en\_el\_diseno\_de\_experimentos\_geotecnicos\_de\_laboratorio\_-  
\_Arbulu\_-\_2023. [Último acceso: 26 Septiembre 2024].

- [20] M. R. H. Montoya, «Repositorio del Sistema Bibliotecario Universidad de San Carlos de Guatemala,» 10 Agosto 2021. [En línea]. Available: <http://www.repositorio.usac.edu.gt/16005/>. [Último acceso: 27 Septiembre 2024].
- [21] T. J. Marchetti y A. J. García, «REPOSITORIO INSTITUCIONAL DE LA UNLP,» 2004. [En línea]. Available: <https://sedici.unlp.edu.ar/handle/10915/22566>. [Último acceso: 27 Noviembre 2024].
- [22] C. Aparicio, R. Stephany, L. Arispe y J. Luis, «Universidad de San Martín de Porres,» 2022. [En línea]. Available: <https://repositorio.usmp.edu.pe/handle/20.500.12727/11359>. [Último acceso: 25 Noviembre 2024].
- [23] «redalyc,» 03 Noviembre 2010. [En línea]. Available: <https://www.redalyc.org/pdf/3421/342130836007.pdf>. [Último acceso: 25 Noviembre 2024].
- [24] I. Nonaka y H. Takeuchi, «Editorial Oxford Press,» 1999. [En línea]. Available: [https://eva.fic.udelar.edu.uy/pluginfile.php/19695/mod\\_resource/content/2/Nonaka%20y%20Takeuchi\\_cap%203.pdf](https://eva.fic.udelar.edu.uy/pluginfile.php/19695/mod_resource/content/2/Nonaka%20y%20Takeuchi_cap%203.pdf). [Último acceso: 29 Septiembre 2024].
- [25] J. C. M. Noh y K. S. M. Cabrera, «Sistema de Información Científica Redalyc,» 2012. [En línea]. Available: <https://www.redalyc.org/pdf/998/99824765003.pdf>. [Último acceso: 29 Septiembre 2024].

- [26] V. A. Pérez y M. F. Urbáez, «Revistas UNAM,» 01 Junio 2017. [En línea]. Available: <https://www.revistas.unam.mx/index.php/entreciencias/article/view/62127/>. [Último acceso: 25 Noviembre 2024].
- [27] L. Kerschberg, «ResearchGate,» Noviembre 2001. [En línea]. Available: [https://www.researchgate.net/publication/2414511\\_Knowledge\\_Management\\_in\\_Heterogeneous\\_Data\\_Warehouse\\_Environments](https://www.researchgate.net/publication/2414511_Knowledge_Management_in_Heterogeneous_Data_Warehouse_Environments). [Último acceso: 30 Septiembre 2024].
- [28] V. A. Pérez y M. F. Urbáez, «Sistema de Información Científica Redalyc,» 05 Mayo 2016. [En línea]. Available: [https://www.redalyc.org/journal/4576/457646537004/html/#redalyc\\_457646537004\\_ref20](https://www.redalyc.org/journal/4576/457646537004/html/#redalyc_457646537004_ref20). [Último acceso: 30 Septiembre 2024].
- [29] A. Ng, «GitHub,» 2018. [En línea]. Available: <https://github.com/ajaymache/machine-learning-yearning.git>. [Último acceso: 28 Julio 2024].
- [30] IBM, «IBM,» IBM, 2019. [En línea]. Available: <https://www.ibm.com/mx-es/topics/machine-learning-algorithms>. [Último acceso: 30 Septiembre 2024].
- [31] M. Rouse, «Techopedia,» Techopedia, 03 Septiembre 2024. [En línea]. Available: <https://www.techopedia.com/es/definicion/aprendizaje-supervisado>. [Último acceso: 30 Septiembre 2024].
- [32] Hewlett Packard Enterprise, «Hewlett Packard Enterprise,» Hewlett Packard Enterprise, 2024. [En línea]. Available: <https://www.hpe.com/lamerica/es/what-is/supervised-machine-learning.html>. [Último acceso: 30 Septiembre 2024].

- [33] Elastic, « Elastic,» Elastic, 2024. [En línea]. Available: <https://www.elastic.co/es/what-is/machine-learning>. [Último acceso: 30 Septiembre 2024].
- [34] e. clarke, «GURU99,» GURU99, 12 Junio 2024. [En línea]. Available: <https://www.guru99.com/es/reinforcement-learning-tutorial.html>. [Último acceso: 30 Septiembre 2024].
- [35] E. K. a. E. L. Steven Bird, «NLTK,» [En línea]. Available: <https://www.nltk.org/book/>. [Último acceso: 28 Julio 2024].
- [36] M. S. H. S. J. S. I. A. R. M. F. M. Md Abrar Jahin, «arXiv,» 07 Febrero 2024. [En línea]. Available: [https://search.arxiv.org/paper.jsp?r=2307.12971&qid=1722214762870ler\\_nCnN\\_-341835186&qs=Text+Preprocessing+for+Machine+Learning](https://search.arxiv.org/paper.jsp?r=2307.12971&qid=1722214762870ler_nCnN_-341835186&qs=Text+Preprocessing+for+Machine+Learning). [Último acceso: 28 Julio 2024].
- [37] C. Fan, M. Chen, X. Wang, J. Wang y B. Huang, «frontiers,» 28 Marzo 2021. [En línea]. Available: <https://www.frontiersin.org/journals/energy-research/articles/10.3389/fenrg.2021.652801/full>. [Último acceso: 25 Noviembre 2024].
- [38] Academia EITCA, «Academia EITCA,» Academia EITCA, 05 Agosto 2023. [En línea]. Available: <https://es.eitca.org/categor%C3%ADa/inteligencia-artificial/eitc-ai-tff-tensorflow-fundamentos/>. [Último acceso: 30 Septiembre 2024].
- [39] P. Huet, «OpenWebinars,» OpenWebinars, 4 Julio 2024. [En línea]. Available: <https://openwebinars.net/blog/tecnicas-clave-para-procesamiento-texto-nlp/#lematizaci%C3%B3n-y-stemming>. [Último acceso: 30 Septiembre 2024].

- [40] Pure Storage, « Pure Storage,» Pure Storage, 2024. [En línea]. Available: <https://www.purestorage.com/es/knowledge/what-is-data-preprocessing.html#:~:text=La%20normalizaci%C3%B3n%20es%20una%20t%C3%A9cnica,de%20un%20conjunto%20de%20datos..> [Último acceso: 30 Septiembre 2024].
- [41] L. Derczynski, «ACL Anthology,» Mayo 2016. [En línea]. Available: <https://aclanthology.org/L16-1040/>. [Último acceso: 05 Noviembre 2024].
- [42] D. C. G. Jurman, «The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,» 02 Enero 2020. [En línea]. Available: <https://link.springer.com/article/10.1186/s12864-019-6413-7>. [Último acceso: 05 Noviembre 2024].
- [43] M. Marqués, «Biblioteca Digita UVHM,» 2011. [En línea]. Available: <https://bdigital.uvhm.edu.mx/wp-content/uploads/2020/05/Bases-de-Datos.pdf>. [Último acceso: 05 Noviembre 2024].
- [44] P. E. M. S. MS, «Google Academy,» Agosto 2008. [En línea]. Available: [https://d1wqtxts1xzle7.cloudfront.net/51623032/Base\\_de\\_datos\\_relacional-libre.pdf?1486147551=&response-content-disposition=inline%3B+filename%3DQue\\_es\\_una\\_base\\_de\\_datos\\_relacional.pdf&Expires=1730844244&Signature=GnbzJ9DVP3PVE2xRSm3-UxGPuUx~ohUAaf9qo8n~SW](https://d1wqtxts1xzle7.cloudfront.net/51623032/Base_de_datos_relacional-libre.pdf?1486147551=&response-content-disposition=inline%3B+filename%3DQue_es_una_base_de_datos_relacional.pdf&Expires=1730844244&Signature=GnbzJ9DVP3PVE2xRSm3-UxGPuUx~ohUAaf9qo8n~SW). [Último acceso: 05 Noviembre 2024].
- [45] M. G. Ginestà y O. P. Mora, «Google Academy,» 2012. [En línea]. Available: <https://artpatcusco.com/sis/pdf/20140527172742en.pdf>. [Último acceso: 05 Noviembre 2024].
- [46] K. P. Tripathi, «CiteSerx,» 2011. [En línea]. Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=19f89f1>

b19b558389e9c70b75403063376cb8e3d. [Último acceso: 05 Noviembre 2024].

- [47] N. Zhang, X. Xu, L. Tao, H. Yu, H. Ye, S. Qiao, X. Xie, X. Chen y H. Chen, «arxiv,» 18 Septiembre 2023. [En línea]. Available: <https://arxiv.org/abs/2201.03335>. [Último acceso: 05 Noviembre 2024].
- [48] A. R. Sajun, I. Zualkernan y D. Sankalpa, «MDPI,» 20 MAyo 2024. [En línea]. Available: <https://www.mdpi.com/2076-3417/14/10/4316>. [Último acceso: 05 Noviembre 2024].
- [49] A. Rogers, O. Kovaleva y A. Rumshisky, «arxiv,» 09 Noviembre 2020. [En línea]. Available: <https://arxiv.org/abs/2002.12327>. [Último acceso: 05 Noviembre 2024].
- [50] Google Research, «Google Research,» Google Research, [En línea]. Available: <https://research.google/blog/exploring-transfer-learning-with-t5-the-text-to-text-transfer-transformer/>. [Último acceso: 25 Noviembre 2024].
- [51] F. Almeida y G. Xexéo, «arXiv,» Enero 2019. [En línea]. Available: <https://arxiv.org/abs/1901.09069>. [Último acceso: 25 Noviembre 2024].
- [52] Stanford University, «Stanford Engineering Everywhere,» Stanford University, [En línea]. Available: <https://see.stanford.edu/Course/CS229>. [Último acceso: 25 Noviembre 2024].
- [53] Amazon AWS, «AWS,» Amazon, [En línea]. Available: <https://aws.amazon.com/es/what-is/django/>. [Último acceso: 05 Noviembre 2024].
- [54] Python, «Python,» Python, [En línea]. Available: <https://docs.python.org/es/3/tutorial/>. [Último acceso: 05 Noviembre 2024].

- [55] JavaScript, «JavaScript Info,» JavaScript Info, [En línea]. Available: <https://javascript.info/>. [Último acceso: 05 Noviembre 2024].
- [56] B. B. A. C. a. G. P. Maria Luisa Farnese, «Frontiers,» 10 Diciembre 2019. [En línea]. Available: <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2019.02730/full>. [Último acceso: 31 Julio 2024].
- [57] J. L. Romero Chacín, R. M. Romero Parra, L. A. Barboza Arenas y e. al., «Sistema de Información Científica Redalyc,» 2021. [En línea]. Available: <https://www.redalyc.org/journal/290/29069612007/>. [Último acceso: 01 Agosto 2024].
- [58] S. M. E. A. Solís, «Repositorio UNEMI,» Enero 2020. [En línea]. Available: <https://repositorio.unemi.edu.ec/handle/123456789/5560>. [Último acceso: 11 Noviembre 2024].
- [59] F. McCalla, I. González, I. Leguías, X. Medianero y V. López, «Universidad Tecnológica de Panamá,» 09 Agosto 2016. [En línea]. Available: <https://revistas.utp.ac.pa/index.php/prisma/article/view/548>. [Último acceso: 05 Noviembre 2024].
- [60] F. Y. P. Martínez, «Repositorio Institucional Universidad Cooperativa de Colombia,» 2022. [En línea]. Available: <https://repository.ucc.edu.co/server/api/core/bitstreams/a7ea175f-de13-49fd-afba-c021baf7a6d0/content>. [Último acceso: 05 Noviembre 2024].

# ANEXOS

## Anexo 1. Código y métricas de entrenamiento del modelo SVM para la clasificación

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
import evaluate

train_dataset =
pd.read_csv('dataset_no_traducido_completo_2700_1.csv')
X_train = train_dataset['instruction']
y_train = train_dataset['intent']
val_dataset = pd.read_csv('dataset_traducido_completo_2700_2.csv')
X_val = val_dataset['instruction']
y_val = val_dataset['intent']

# Codificar las etiquetas
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_val_encoded = label_encoder.transform(y_val)
# Crear un pipeline con vectorizador, escalador y modelo SVM
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer()), # Vectorizador de texto
    ('svm', SVC()) # Modelo SVM
])
# Definir los parámetros para GridSearchCV
param_grid = {
    'svm__C': [0.1, 1, 10],
    'svm__gamma': ['scale', 'auto'],
    'svm__kernel': ['linear', 'rbf']
}
# Configurar GridSearchCV
grid_search = GridSearchCV(pipeline, param_grid, cv=5,
scoring='accuracy', return_train_score=True)
grid_search.fit(X_train, y_train_encoded)
# Obtener el mejor modelo
best_model = grid_search.best_estimator_
# Evaluar el modelo en el conjunto de validación
y_pred_encoded = best_model.predict(X_val)
```

```

# Calcular métricas
metric = evaluate.load("accuracy")
accuracy = metric.compute(predictions=y_pred_encoded,
references=y_val_encoded)
print(f"Mejor modelo: {grid_search.best_params}")
print(f"Precisión en el conjunto de validación:
{accuracy['accuracy']:.4f}")
Mejor modelo: {'svm_C': 10, 'svm_gamma': 'scale', 'svm_kernel':
'linear'}
Precisión en el conjunto de validación: 0.9656

```

Figura 24 Código usado para entrenar el modelo de maquinas SVM

Con estos parámetros de entrenamiento se pudo obtener un porcentaje de acierto del 0.96% que se puede evidenciar en estos reportes.

Categoría	TP	FP	FN	Precisión	Recall	F1 Score
cancel_order	89	0	0	1	1	1
change_order	91	6	0	0,93814433	1	0,96808511
change_shipping	97	6	0	0,94174757	1	0,97
check_cancel	99	0	0	1	1	1
check_invoice	96	21	0	0,82051282	1	0,90140845
check_payment	98	0	0	1	1	1
check_refund	97	0	0	1	1	1
complaint	99	0	0	1	1	1
contact_customer	99	0	0	1	1	1
contact_human	100	0	0	1	1	1
create_account	93	10	0	0,90291262	1	0,94897959
delete_account	96	11	0	0,89719626	1	0,94581281
delivery_options	90	11	0	0,89108911	1	0,94240838
delivery_preferences	98	0	0	1	1	1
edit_account	93	9	0	0,91176471	1	0,95384615
get_invoice	77	2	0	0,97468354	1	0,98717949
get_refund	92	8	0	0,92	1	0,95833333
newsletter_subscription	99	0	0	1	1	1
payment_issues	97	2	0	0,97979798	1	0,98979592
place_order	93	17	0	0,84545455	1	0,91625616
recover_password	98	0	0	1	1	1
registration	94	3	0	0,96907217	1	0,98429319
review	98	1	0	0,98989899	1	0,99492386
set_up_shipping	93	1	0	0,9893617	1	0,99465241
switch_account	91	12	0	0,88349515	1	0,93814433
track_order	95	11	0	0,89622642	1	0,94527363

Tabla 21 Tabla de resultados F1 Score del modelo SVM

En general, el rendimiento del modelo es satisfactorio, obteniendo un porcentaje de acierto del 96%, pero se podrían realizar ajustes en categorías específicas para mejorar aún más su precisión y recall.

### Matriz de confusión generada

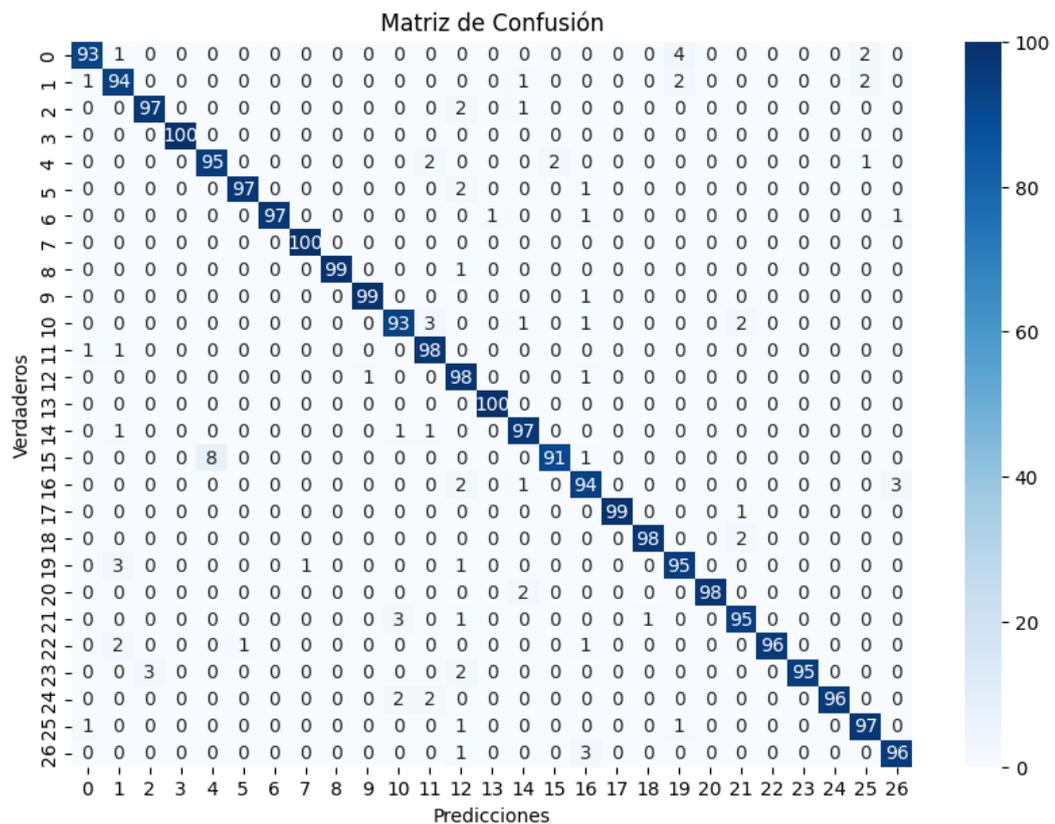


Figura 25 Matriz de confusión del modelo SVM

## Anexo 2. Comparativa de estrategias para optimizar el acceso a la información

Datos sintéticos generados

Instrucción original	Versión con cambios del 20%	Versión con cambios del 40%	Versión con cambios del 60%	Versión con cambios del 80%
¿Cómo puedo activar mi cuenta?	¿Cómo puedo habilitar mi cuenta?	Pasos para activar mi cuenta personal.	para mi Necesito activar una cuenta; ¿cómo hacerlo?	Configurar mi cuenta desde cero.
Quiero cambiar mi	Deseo cambiar	¿Cómo puedo la	He olvidado mi contraseña	Actualizar mis datos de acceso.

<b>contraseña olvidada.</b>	contraseña que olvidé.	contraseña olvidada?	y necesito cambiarla.	
<b>¿Cómo rastrear mi pedido?</b>	¿Cómo puedo rastrear mi paquete?	Pasos para encontrar el estado de mi pedido.	Quiero saber dónde está mi paquete actual.	Consultar el estado de una compra.
<b>Ayuda para eliminar mi cuenta principal.</b>	¿Cómo eliminar mi cuenta principal?	Necesito borrar mi cuenta principal.	Solicito ayuda para dar de baja mi cuenta principal.	Cerrar la cuenta asociada al sistema.
<b>Quiero configurar una nueva dirección de envío.</b>	¿Cómo puedo agregar una dirección de envío?	Pasos para añadir una nueva dirección para envíos.	Configurar una dirección alternativa para los envíos.	Actualizar información de envío en la cuenta.
<b>¿Qué métodos de pago puedo usar?</b>	¿Qué opciones de pago están disponibles?	Métodos de pago aceptados, ¿cuáles son?	Quiero conocer las formas de pago que permiten.	Consultar las opciones de transacciones disponibles.
<b>¿Cómo puedo actualizar mis datos personales?</b>	¿Cómo modificar mi información personal?	Actualizar mi perfil con datos nuevos, ¿cómo hacerlo?	Cambiar mi información personal en el sistema.	Editar los detalles registrados en mi cuenta.
<b>¿Cómo solicitar un reembolso?</b>	¿Qué pasos seguir para pedir un reembolso?	Instrucciones para solicitar la devolución del dinero.	Quiero saber cómo pedir un reembolso de mi compra.	Realizar una solicitud de devolución.
<b>¿Qué debo hacer para cambiar mi suscripción?</b>	¿Cómo modificar mi tipo de suscripción?	Pasos para ajustar mi suscripción actual.	Cambiar el plan de mi suscripción; ¿qué debo hacer?	Actualizar mi tipo de membresía.
<b>¿Cómo desactivar las notificaciones por correo?</b>	¿Cómo puedo deshabilitar los correos automáticos?	Pasos para dejar de recibir correos no deseados.	Quiero desactivar las notificaciones de mi correo electrónico.	Configurar preferencias para no recibir correos.

*Tabla 22 Datos sintéticos generados para la búsqueda de la mejor estrategia para la interacción NLP*

## Comparación semántica función coseno con TF-IDF

Instrucción original: ¿Cómo puedo activar mi cuenta?		
<b>Instrucción sintética</b>	Palabras diferentes de la oración original	Palabras diferentes de la oración sintética
¿Cómo puedo habilitar mi cuenta?	activar	habilitar
<b>Pasos para activar mi cuenta personal.</b>	puedo, cuenta?, ¿cómo	cuenta, para, pasos, personal.
<b>Necesito activar una cuenta; ¿cómo hacerlo?</b>	puedo, cuenta?, mi	una, necesito, hacerlo?, cuenta;
<b>Configurar mi cuenta desde cero.</b>	activar, puedo, cuenta?, ¿cómo	cero., desde, cuenta, configurar
Instrucción original: Quiero cambiar mi contraseña olvidada		
<b>Deseo cambiar la contraseña que olvidé.</b>	quiero, olvidada., mi	que, olvidé, la
¿Cómo puedo modificar la contraseña olvidada?	quiero, cambiar, olvidada., mi	modificar, puedo, ¿cómo, la, olvidada?
<b>He olvidado mi contraseña y necesito cambiarla.</b>	quiero, cambiar, olvidada.	y, he, olvidado, cambiarla., necesito
<b>Actualizar mis datos de acceso.</b>	contraseña, olvidada., mi, quiero, cambiar	datos, actualizar, acceso., mis, de

Tabla 23 Resultados del uso de TF-IDF más función coseno

**Conclusión:** El uso de TF-IDF para generar embeddings y la función coseno revela que la combinación de ambos es capaz de identificar variaciones de tipo de persona y cambios estructurales en las frases. Pero no es eficaz en manejar sinónimos contextuales y equivalencias semánticas más complejas, penaliza frases con cambios en el nivel de detalle, aunque el contexto general sea el mismo y no considera relaciones de dependencia semántica entre palabras.

## Comparación clustering + kmeans

Para esta prueba se le asignó a cada instrucción original un cluster, para que el algoritmo kmeans agrupe el resto de oraciones en base a las similitudes semánticas que encuentre con los cluster de las instrucciones originales.

Instrucción Original	Cluster
¿Cómo puedo activar mi cuenta?	8
Quiero cambiar mi contraseña olvidada.	1
¿Cómo rastrear mi pedido?	8
Ayuda para eliminar mi cuenta principal.	0

Quiero configurar una nueva dirección de envío.	5
¿Qué métodos de pago puedo usar?	2
¿Cómo puedo actualizar mis datos personales?	4
¿Cómo solicitar un reembolso?	7
¿Qué debo hacer para cambiar mi suscripción?	2
¿Qué debo hacer para cambiar mi suscripción?	3

Tabla 24 Tabla de cluster asignados a las instrucciones originales

<b>Instrucción original: ¿Cómo rastrear mi pedido?</b>	
<b>Instrucción sintética</b>	Cluster asignado por kmeans
¿Cómo puedo rastrear mi paquete?	8
Pasos para encontrar el estado de mi pedido.	5
Quiero saber dónde está mi paquete actual.	7
Consultar el estado de una compra.	7
<b>Instrucción original: Ayuda para eliminar mi cuenta principal.</b>	
<b>Instrucción sintética</b>	Cluster asignado por kmeans
¿Cómo eliminar mi cuenta principal?	6
Necesito borrar mi cuenta principal.	6
Solicito ayuda para dar de baja mi cuenta principal.	0
Cerrar la cuenta asociada al sistema.	1

Tabla 25 Asignación de cluster a instrucciones sintéticas por el algoritmo

#### *Kmeans*

**Conclusión:** El uso de agrupación por clustering en conjunto con el algoritmo kmeans tuvo problemas al logró agrupar correctamente todas las frases relacionadas en los clústeres esperados debido a las limitaciones en la similitud semántica y la falta de un anclaje explícito a las frases originales. Ya que KMeans se basa en medidas de distancia matemática (como la distancia euclidiana en el espacio TF-IDF) y no necesariamente en la similitud semántica profunda entre frases. Por ello, frases con términos diferentes, aunque relacionadas conceptualmente, pueden asignarse a clústeres distintos.

#### **Comparación modelo + coseno**

Para estas pruebas se probarán los modelos Flan-T5 y bert-base-spanish-wwm-uncased ambos modelos son capaces de comprender de forma óptima el idioma español. Esta prueba consiste en generar embeddings con los modelos y usar la

función coseno para ver la similitud, se usará el valor de 1 para indicar similitud y 0 para indicar que no existe similitud.

Instrucción sintetica	Similitud modelo Flan-T5	Similitud modelo bert-base-spanish
¿Cómo puedo agregar una dirección de envío?	0	1
Pasos para añadir una nueva dirección para envíos.	1	1
Configurar una dirección alternativa para los envíos.	1	1
Actualizar información de envío en la cuenta.	0	1
Instrucción original: ¿Qué métodos de pago puedo usar?		
¿Qué opciones de pago están disponibles?	1	1
Métodos de pago aceptados, ¿cuáles son?	0	1
Quiero conocer las formas de pago que permiten.	0	1
Consultar las opciones de transacciones disponibles.	0	1
Instrucción original: ¿Cómo puedo actualizar mis datos personales?		
¿Cómo modificar mi información personal?	1	1
Actualizar mi perfil con datos nuevos, ¿cómo hacerlo?	1	1
Cambiar mi información personal en el sistema.	0	1
Editar los detalles registrados en mi cuenta.	0	1

Tabla 26 Validación de similitud por el modelo T5 y Bert-Base-Spanish

**Conclusión:** El modelo Flan-T5 demostró ser menos sensible a las variaciones sintácticas cuando se usa para tareas de *embedding* o *similaridad* directas. Mientras que el modelo bert-base-spanish presenta un enfoque más centrado en la representación de la relación semántica entre palabras y frases, lo que lo hace más eficaz en tareas de similitud semántica.

### Anexo 3. Código de la función para generar la clasificación usando el modelo SVM

Función clasificadora

```
#Función para la clasificación
def clasificar_consulta(request):
    if request.method == 'POST':
        try:
            data = json.loads(request.body) # Cargar el cuerpo de la
solicitud
        except json.JSONDecodeError:
            return JsonResponse({'error': 'Invalid JSON'}, status=400)

        consulta = data.get('consulta') # Obtiene la consulta

        if not consulta: # Verifica si la consulta está vacía
            return JsonResponse({'error': 'Consulta es requerida'},
status=400)

        try:
            # Realizar la predicción
            prediccion_codificada = modelo_cargado.predict([consulta])
            prediccion =
encoder.inverse_transform(prediccion_codificada)

            # Retornar la predicción en formato JSON
            return JsonResponse({'prediccion': prediccion[0]},
status=200)

        except Exception as e:
            print(f"Error en la lógica de clasificación: {e}")
            return JsonResponse({'error': 'Error procesando la
consulta'}, status=500)

    return JsonResponse({'error': 'Invalid request'}, status=400)
```

Figura 26 Código usado para genrar la clasificación con el modelo SVM

**Anexo 4.** Código de las funciones para la comparación semántica tanto para incidentes como validación de registros.

Consulta de incidentes del usuario

```

Función clasificación semántica usuario consulta

def handle_query_optimized(intent, query):
    # Usa la función optimizada para obtener los datos consolidados
    con average_ranking
    results = query_database_with_average_ranking(intent)

    if not results:
        return JsonResponse({"intent": intent, "respuesta": []}) # No
hay resultados

    # Embedding de la consulta del usuario
    user_response_embedding = model_similitud.encode(query,
convert_to_tensor=False)
    respuestas_detalles = []
    result_embeddings = []

    for fila in results:
        instruction_text = fila["instruction"]
        response_text = fila["response"]
        average_ranking = fila["average_ranking"] # Promedio del
ranking
        instruction_embedding_str = fila["instruction_embedding"]
        consulta_id = fila["id"] # Recupera el ID del registro

        try:
            # Convertir el embedding desde el string hexadecimal
            embedding_bytes =
bytes.fromhex(instruction_embedding_str.replace('\x', ''))
            embedding_array = np.frombuffer(embedding_bytes,
dtype=np.float32)

            if embedding_array.ndim == 1: # Asegurar que sea 2D
                embedding_array = embedding_array.reshape(1, -1)

            result_embeddings.append(embedding_array)
            respuestas_detalles.append({
                "id": consulta_id, # ID del registro
                "instruction": instruction_text,
                "response": response_text,
                "average_ranking": average_ranking # Usar el promedio
            })
        except (ValueError, SyntaxError) as e:
            print(f"Error al procesar fila: {fila}, Error: {e}")
            continue

    if not result_embeddings:
        return JsonResponse({"intent": intent, "respuesta": []}) # No
hay embeddings válidos

    # Combinar embeddings en un array 2D
    result_embeddings = np.vstack(result_embeddings)
    user_response_embedding = np.expand_dims(user_response_embedding,
axis=0)

    # Calcular similitud coseno
    similitudes = cosine_similarity(user_response_embedding,
result_embeddings)[0]

    # Obtener las 3 respuestas con mayor similitud
    top_indices = np.argsort(similitudes)[-3:][::-1]

    top_respuestas_similitudes = [
        {
            "id": respuestas_detalles[index]["id"], # ID de la
respuesta
            "respuesta": respuestas_detalles[index]["response"],
            "average_ranking": respuestas_detalles[index]
["average_ranking"] # Usar average_ranking
        }
        for index in top_indices
    ]

    # Estructura final de la respuesta
    response_data = {
        "intent": intent,
        "respuesta": top_respuestas_similitudes
    }
    return JsonResponse(response_data)

```

Figura 27 Código para la comparación semántica de la consulta del usuario

## Validación de similitudes con respuestas

Función clasificación semántica usuario consulta

```
def handle_query_optimized_respuesta(user_response, intent,
threshold_cosine=0.95):

    results = query_database_3(intent)

    if not results:
        print("No se encontraron registros en la base de datos para el
intent especificado.")
        return

    user_response_embedding = modelo_respuesta.encode(user_response,
convert_to_tensor=False)
    respuestas_detalle = []
    result_embeddings = []

    for fila in results:
        response_text = fila[2]
        response_embedding_str = fila[6]

        try:
            # Convertir la cadena hexadecimal a bytes y luego a un
array de floats
            response_embedding_bytes =
bytes.fromhex(response_embedding_str.replace('\x', ''))
            response_embedding =
np.frombuffer(response_embedding_bytes, dtype=np.float32)

            if response_embedding.ndim == 1:
                response_embedding = response_embedding.reshape(1, -1)
            result_embeddings.append(response_embedding)
        except (SyntaxError, ValueError) as e:
            print(f"Error al procesar el embedding: {e} en fila:
{fila}") # Manejar el error según tu lógica
            continue # Omitir esta fila si hay un error

        respuestas_detalle.append({
            "id": fila[0],
            "instruction": fila[1],
            "categoria": fila[3],
            "intencion": fila[4],
            "response": response_text
        })

    # Apilar embeddings de respuestas
    if result_embeddings:
        result_embeddings = np.vstack(result_embeddings)
    else:
        print("No hay embeddings válidos para calcular la similitud.")
        return

    user_response_embedding = np.expand_dims(user_response_embedding,
axis=0)

    # Calcular similitudes
    similitudes = cosine_similarity(user_response_embedding,
result_embeddings)[0]

    # Filtrar las respuestas que cumplen con el umbral de similitud
    top_respuestas_similitudes = [(respuestas_detalle[i],
similitudes[i]) for i in range(len(similitudes)) if similitudes[i] >=
threshold_cosine]

    if top_respuestas_similitudes:
        mejor_respuesta, mejor_similitud =
max(top_respuestas_similitudes, key=lambda x: x[1])
        return mejor_respuesta, float(mejor_similitud)
    else:
        print("No se encontraron coincidencias con la similitud
coseno.")
        return [], [] # Retornar dos arreglos vacío
```

Figura 28 Código de la comparación semántica para respuestas

## Anexo 5. Código de las funciones para la creación o actualización de embeddings.

### Función generar\_embeddings

Función relacionado con los embeddings

```
def generar_embeddings(instruction, response):
    try:
        # Generar el embedding para la instrucción usando
        # model_similitud
        instruction_embedding = model_similitud.encode(instruction,
            convert_to_tensor=True).numpy()
        print("Size of instruction embedding:",
            instruction_embedding.shape)

        # Asegurarse de que instruction_embedding es un numpy array
        if isinstance(instruction_embedding, list):
            instruction_embedding = np.array(instruction_embedding)

        # Generar el embedding para la respuesta usando
        # modelo_respuesta
        response_embedding = modelo_respuesta.encode(response,
            convert_to_tensor=True).numpy()
        print("Size of response embedding:", response_embedding.shape)

        # Asegurarse de que response_embedding es un numpy array
        if isinstance(response_embedding, list):
            response_embedding = np.array(response_embedding)

        # Retornar los embeddings en un diccionario
        return {
            "instruction_embedding": instruction_embedding,
            "response_embedding": response_embedding
        }
    except Exception as e:
        print(f"Error al generar embeddings: {e}")
        return None
```

Figura 29 Código para generar embeddings

## Función insertar\_embedding

```
def insertar_embedding(cursor, consulta_id, instruction_embedding,
response_embedding):
    # Convertir a numpy array si no lo es y asegurarse de que sea
float32
    instruction_embedding = np.asarray(instruction_embedding,
dtype=np.float32)
    response_embedding = np.asarray(response_embedding,
dtype=np.float32)

    # Convertir los embeddings al formato binario usando tobytes()
instruction_embedding_bytes = instruction_embedding.tobytes()
response_embedding_bytes = response_embedding.tobytes()

    # Insertar el nuevo registro en formato binario (bytea)
cursor.execute("""
    INSERT INTO embeddings (consulta_id, instruction_embedding,
response_embedding)
    VALUES (%s, %s, %s)
    """, (consulta_id, instruction_embedding_bytes,
response_embedding_bytes))

    print(f"Inserción exitosa para consulta_id: {consulta_id}")
```

*Figura 30 Código para insertar embeddings*

## Función actualizar\_embedding

```
def actualizar_embedding(cursor, consulta_id, instruction_embedding,
response_embedding):
    # Convertir a numpy array si no lo es y asegurarse de que sea
float32
    instruction_embedding = np.asarray(instruction_embedding,
dtype=np.float32)
    response_embedding = np.asarray(response_embedding,
dtype=np.float32)

    print(f"Tamaño de response_embedding antes de tobytes:
{response_embedding.shape}")
    print(f"Tipo de datos de response_embedding:
{response_embedding.dtype}")

    # Convertir los embeddings al formato binario usando tobytes()
instruction_embedding_bytes = instruction_embedding.tobytes()
response_embedding_bytes = response_embedding.tobytes()

    print(f"Tamaño de response_embedding después de tobytes:
{len(response_embedding_bytes)} bytes")

    # Actualizar los registros en la tabla embeddings en formato
binario
    cursor.execute("""
        UPDATE embeddings
        SET instruction_embedding = %s, response_embedding = %s
        WHERE consulta_id = %s
    """, (instruction_embedding_bytes, response_embedding_bytes,
consulta_id))

    # Actualizar el campo needs_update a FALSE en la tabla soporte_n1
cursor.execute("""
        UPDATE soporte_n1
        SET needs_update = FALSE
        WHERE id = %s
    """, (consulta_id,))

    # Confirmar que la actualización se completó
    print(f"Actualización exitosa para consulta_id: {consulta_id}")
```

*Figura 31 Código para actualizar embeddings*

**Anexo 6.** Tabla comparativa de dataset encontrados

Dataset	Tamaño	Columnas	Idioma	Link
Bitext/bitext-customer-support-llm-chatbot-training-dataset	26,872 Registros	flags instruction category intent response	Ingles	<a href="#">Ver</a>
Nirmitsinghhh/raw_data_eng_helpdesk-llama2-1k	1,022 Registros	user_query intent solution	Ingles	<a href="#">Ver</a>
Nebulabyte/e-commerce_customer_support_conversations	1,000 Registros	issue_area issue_category issue_sub_category issue_category_sub_category customer_sentiment product_category product_sub_category issue_complexity agent_experience_level agent_experience_level_desc conversation	Ingles	<a href="#">Ver</a>
Bitext/bitext-mortgage-loans-llm-chatbot-training-dataset	36,778 Registros	system_prompt instruction intent category tags response	Ingles	<a href="#">Ver</a>

*Tabla 27 Dataset en línea analizados*

## Anexo 8. Acondicionamiento del dataset

### Separación de respuestas estructuradas y planas

```
import pandas as pd

# Cargar tu dataset
df =
pd.read_csv('Bitext_Sample_Customer_Support_Training_Dataset_27K_respo
nses-v11.csv', encoding='utf-8')

pattern = r".+:\s*\d+\.\s+.*"

# Filtrar los datos que cumplen con el patrón
filtered_data = df[df['response'].str.contains(pattern, regex=True)]

# Filtrar los datos que NO cumplen con el patrón
non_compliant_data = df[~df['response'].str.contains(pattern,
regex=True)]

# Número total de registros (filas) en el DataFrame
total_registros = len(filtered_data)
print("Total de registros:", total_registros)

total_registros = len(non_compliant_data)
print("Total de registros no:", total_registros)
```

*Figura 32 Código para separar registros con respuestas estructuradas y planas*

### Eliminación de placeholders de las columnas “instruction” y “response”

```
import pandas as pd

# Cargar el dataset original
df = pd.read_csv('resultados_filtrados_instrucciones.csv')

# Función para eliminar placeholders entre corchetes
def remove_placeholders(text):
    return re.sub(r'\{\{.*?\}\}', '', text)

# Aplicar la función a las columnas relevantes
df['instruction'] = df['instruction'].apply(remove_placeholders)
df['response'] = df['response'].apply(remove_placeholders)

# Guardar el dataset con los placeholders eliminados
df.to_csv('dataset_limpio.csv', index=False, encoding='utf-8-sig')
```

*Figura 33 Eliminación de placeholders*

Traducción de dataset usando el modelo “Helsinki-NLP/opus-mt-en-es”

Librerías, funciones e instalaciones

```
# Instalación de librerías
!pip install unidecode

import pandas as pd
from transformers import MarianMTModel, MarianTokenizer
from tqdm import tqdm
from unidecode import unidecode
import string
import torch
import re

# Cargar el modelo y el tokenizador MarianMT para inglés-español
model_name = 'Helsinki-NLP/opus-mt-en-es'
tokenizer = MarianTokenizer.from_pretrained(model_name)
model = MarianMTModel.from_pretrained(model_name)

# Función para traducir texto
def translate_text_2(text):
    inputs = tokenizer(text, return_tensors="pt", padding=True,
truncation=True).to(device)
    translated = model.generate(**inputs)
    return tokenizer.decode(translated[0], skip_special_tokens=True)

# Cargar el dataset
df = pd.read_csv('dataset_limpio_español_v1.csv', encoding='utf-8')

# Traducir las columnas: 'instruction' o 'response'
columns_to_translate = ['response']

# Mover el modelo a la GPU si está disponible
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model.to(device)

# Traducir solo la columna 'instruction' con una única barra de
progreso - GPU
with tqdm(total=len(df['response']), desc='Traduciendo instruction',
unit='item') as pbar:
    for i in range(len(df['response'])):
        df['response'].iat[i] =
translate_text_2(df['response'].iat[i]) # Traducir el texto
        pbar.update(1) # Actualizar la barra de progreso

# Traducir solo los primeros 2000 registros de la columna 'response'
con barra de progreso
with tqdm(total=1500, desc='Traduciendo response', unit='item') as
pbar:
    for i in range(min(2000, len(df['response']))): # Limitar a los
primeros 2000 registros
        df['response'].iat[i] =
translate_text_2(df['response'].iat[i]) # Traducir el texto
        pbar.update(1) # Actualizar la barra de progreso
```

```

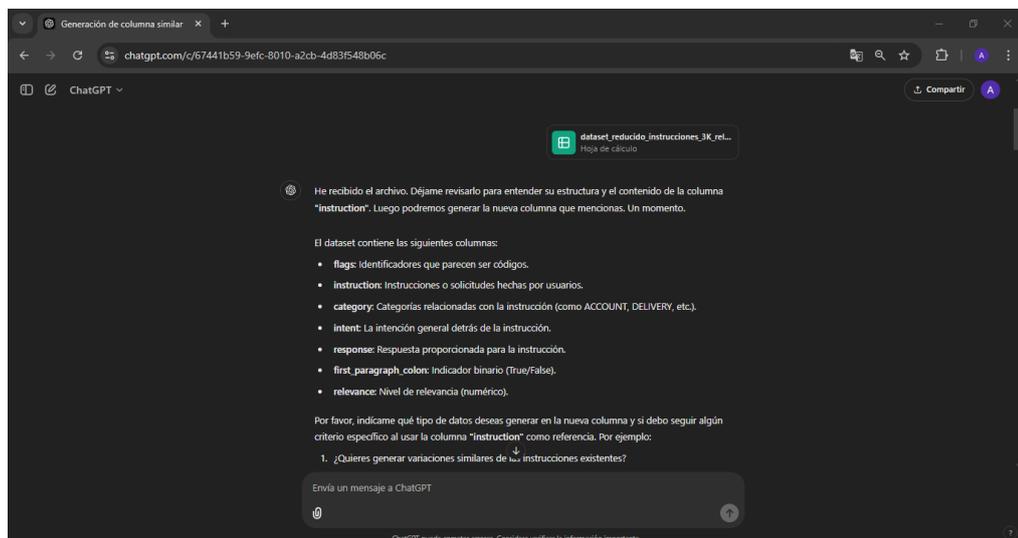
model = MarianMTModel.from_pretrained(model_name).to('cpu')

# Traducir solo la columna 'instruction' con una única barra de
# progreso
with tqdm(total=len(df['instruction']), desc='Traduciendo
instruction', unit='item') as pbar:
    for i in range(len(df['instruction'])):
        df['instruction'].iat[i] =
        translate_text(df['instruction'].iat[i]) # Traducir el texto
        pbar.update(1) # Actualizar la barra de progreso

```

*Figura 34 Código para hacer la traducción colocando el modelo usando GPU como CPU*

## Anexo 8. Generación de Datos Sintéticos usando ChatGPT

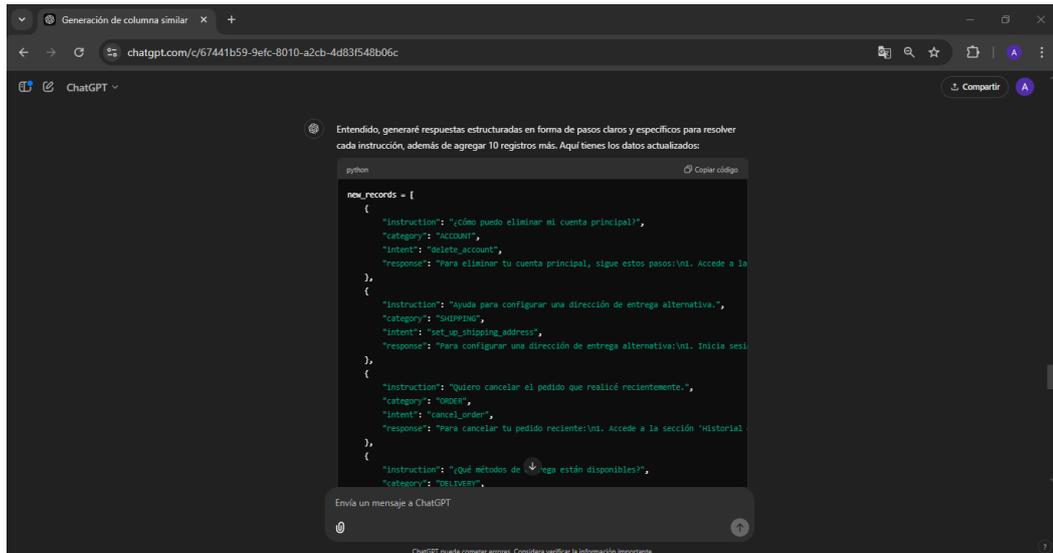


*Figura 35 Carga del dataset al modelo GPT 4min usando la interfaz de ChatGPT*

### Creación de datos para elegir la mejor estrategia de comparación semántica

**Pront:** Me puedes dar un arreglo con nuevos registros basados en los datos originales del dataset, quiero una instrucción, la categoría e intención en base al dataset inicial y una respuesta estructurada para la consulta que creaste. Maten el orden en el que crea los datos.

## Resultados:



Entendido, generaré respuestas estructuradas en forma de pasos claros y específicos para resolver cada instrucción, además de agregar 10 registros más. Aquí tienes los datos actualizados:

```
python
new_records = [
  {
    "instruction": "¿cómo puedo eliminar mi cuenta principal?",
    "category": "ACCOUNT",
    "intent": "delete_account",
    "response": "Para eliminar tu cuenta principal, sigue estos pasos:\n1. Accede a la
  },
  {
    "instruction": "Ayuda para configurar una dirección de entrega alternativa.",
    "category": "SHIPPING",
    "intent": "set_alt_shipping_address",
    "response": "Para configurar una dirección de entrega alternativa:\n1. Inicia sesi
  },
  {
    "instruction": "quiero cancelar el pedido que realicé recientemente.",
    "category": "ORDER",
    "intent": "cancel_order",
    "response": "Para cancelar tu pedido reciente:\n1. Accede a la sección 'Historial
  },
  {
    "instruction": "¿Qué métodos de entrega están disponibles?",
    "category": "DELIVERY",

```

Envía un mensaje a ChatGPT

ChatGPT puede cometer errores. Considera verificar la información importante.

Figura 36 Evidencia de generación de datos sintéticos