



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

TITULO DEL TRABAJO DE TITULACIÓN

**SISTEMA COGNITIVO DE GESTIÓN DEL CONOCIMIENTO PARA
LA ATENCIÓN DE INCIDENTES TECNOLÓGICOS EN EL PROCESO
DE NIVELACION UNIVERSITARIA, BASADA EN ONTOLÓGICAS Y
NLP/ML.**

AUTOR

ORRALA GONZALEZ, ANDREA CAROLINA

PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR

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

TUTOR

ING. CORONEL SUAREZ, MARJORIE. MSC.

Santa Elena, Ecuador

Año 2026



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

TRIBUNAL DE SUSTENTACIÓN



**Ing. José Sánchez Aquino Mgtr.
DIRECTOR DE LA CARRERA**



**Ing. Marjorie Coronel Suárez Mgtr.
TUTORA**



**Ing. Walter Orozco Iguasnia Mgtr.
DOCENTE ESPECIALISTA**



**Ing. Marjorie Coronel Suárez Mgtr.
DOCENTE GUÍA UIC**



UPSE

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

CERTIFICACIÓN

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

La Libertad, a los 27 días del mes de Julio del año 2026

TUTOR

Ing. Marjorie Coronel S. Msc.



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

DECLARACIÓN DE RESPONSABILIDAD

Yo, Orrala González Andrea Carolina

DECLARO QUE:

El trabajo de Titulación, **Sistema cognitivo de gestión del conocimiento para la atención de incidentes tecnológicos en el proceso de nivelación universitaria, basada en ontologías y NLP/MI**, desarrollado como requisito previo para la obtención del título de Ingeniera en Tecnologías de la Información, es resultado de mi trabajo académico y técnico, y ha sido elaborado respetando los derechos de propiedad intelectual de terceros, conforme a las citas y referencias bibliográficas incorporadas en el documento.

Asimismo, reconozco que la propiedad intelectual del presente trabajo no corresponde exclusivamente a mi autoría, debido a que se desarrolla en el marco del proyecto de investigación denominado “Aplicación de Inteligencia Artificial y Aprendizaje Automático para la Optimización de la Clasificación y Resolución de Incidentes Tecnológicos en la Gestión de Servicios de TI”, aprobado por la Universidad Estatal Península de Santa Elena, con código INCYT-PNF-2025-S73-199, bajo la dirección de la Ing. Marjorie Coronel S., Mgtr.

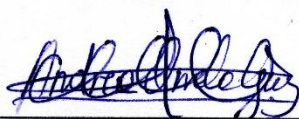
La idea original, el planteamiento general y el diseño conceptual de la propuesta fueron formulados por la directora del proyecto y tutora del trabajo de titulación. Mi

contribución se concentró principalmente en el diseño técnico, implementación, ejecución y documentación del sistema, de conformidad con los objetivos, lineamientos y alcance establecidos en el referido proyecto de investigación.

En virtud de lo expuesto, asumo la responsabilidad por el contenido, los resultados, las interpretaciones y las conclusiones presentadas en este trabajo de titulación, dentro del alcance de mi participación y contribución académica.

La Libertad, a los 27 días del mes de Julio del año 2025

EL AUTOR



Orrala González Andrea Carolina



UPSE

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

AUTORIZACIÓN

Yo, Orrala González Andrea Carolina

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 presente trabajo de titulación con fines de difusión pública, además apruebo la reproducción dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

Santa Elena, a los 27 días del mes de Julio del año 2025

EL AUTOR

Orrala González Andrea Carolina

AGRADECIMIENTO

Quiero agradecer en primer lugar a Dios por haber sido mi guía siempre, la oportunidad de culminar mis estudios gracias a la fortaleza, salud y sabiduría que me otorgó en cada momento para finalizar esta etapa.

Agradezco a mi familia, a mi mamá Sra. Elina González, mi papá Sr. Manuel Orrala, mi hermano Edwin Orrala por su comprensión, amor y apoyo incondicional en cada paso del camino, a mis abuelitos Sra. Azucena, Sr. Nelson y Sra. Betty por el cariño y sus sabios consejos en cada momento de mi vida. A todos mis familiares que me motivaron a seguir adelante.

A Miguel Carcelén por su apoyo, escucha y por siempre estar pendiente en cada momento. Tu compañía ha sido valiosa e importante.

Agradezco a mi docente guía Ing. Marjorie Coronel, por acompañarme en cada parte de este proceso, por el tiempo dedicado compartiendo sus conocimientos para culminar de manera satisfactoria este proyecto.

Andrea Carolina, Orrala González

DEDICATORIA

Le dedico este trabajo a Dios por no haberme dejado rendirme en los momentos más difíciles, A mis padres Sra.Elina González y Sr. Manuel Orrala por su amor, por su paciencia y su apoyo incondicional durante toda mi vida, a mi hermanito Edwin Orrala por ayudarme siempre y escucharme cuando más lo necesitaba, a mis queridos abuelitos Sra. Azucena, Sr. Nelson y Sra. Betty por el cariño y los consejos que nunca faltaron, ustedes siempre fueron mi motivación para seguir adelante.

Andrea Carolina, Orrala González

ÍNDICE GENERAL

TITULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XIII
ÍNDICE DE FIGURAS	XIV
RESUMEN	XVII
ABSTRACT	XVIII
INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN	2
1.1. Antecedentes	2
1.2. Descripción Del Proyecto	5
1.3. Objetivos Del Proyecto	9
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	13
1.6.3. Diseño y Alcance de la Investigación	13
1.6.4. Tipo y Métodos de Investigación	13
1.6.6. Muestra	14
1.6.7. Idea a Defender	14
1.6.8. Recolección de Datos	16
1.7. METODOLOGÍA DE DESARROLLO	17
1.7.1. Fase de Adquisición de Conocimiento:	17
1.7.2. Fase de Conceptualización:	17
1.7.3. Fase Formalización y Desarrollo:	18
1.7.4. Fase de Implementación:	18
1.7.5. Fase de Evaluación y Validación:	18
CAPÍTULO 2. PROPUESTA	19
2.1. MARCO CONTEXTUAL	19
2.2. MARCO CONCEPTUAL	19
2.2.1. Conocimiento Organizacional	19
2.2.2. Representación del Conocimiento	21
2.2.3. Contexto Institucional	24
2.2.4. IA y NLP	25
2.2.5. Herramientas Tecnológicas	28
2.3. MARCO TEÓRICO	30

2.3.1. Gestión del Conocimiento en el Soporte Técnico Institucional	30
2.3.2. Ingeniería Ontológica y Representación Formal del Conocimiento de Dominio	31
2.3.3. Modelos de Lenguaje y Recuperación Semántica del Conocimiento	32
2.3.4. Arquitecturas de Software Orientadas a Servicios para Sistemas Inteligentes Conversacionales	33
CAPITULO 3. COMPONENTES DE LA PROPUESTA	35
3.1 FASE DE ADQUISICIÓN DEL CONOCIMIENTO	35
3.1.1 Análisis de Recolección de Datos	35
3.1.2. Análisis Documental	36
3.1.3. Catálogo de Servicios	38
3.2. FASE DE CONCEPTUALIZACIÓN	42
3.2.1. Modelo Conceptual: Construcción de la Ontología	42
3.2.2. Datasets	43
3.3. FASE DE FORMALIZACIÓN Y DESARROLLO	46
3.3.1. Arquitectura del Sistema	46
3.3.2. Diseño y Especificaciones del Esquema de Base de Datos Relacional	47
3.3.3. Dataset en Formato Tabular	53
3.3.4. Reglas Lógicas para la Ontología	55
3.3.5. Desarrollo del Sistema Neuro-Simbólico de Clasificación	56
3.3.6. Desarrollo del Modal de Gestión del Conocimiento	59
3.4. FASE DE IMPLEMENTACIÓN	61

3.4.1. Uso del Sistema Neuro-Simbólico - Canal de Comunicación	61
3.4.2. Implementación del Gestor de Tickets Zammad	65
3.4.3. Implementación del Modal	69
3.5. FASE DE EVALUACION Y VALIDACION	71
3.5.1. Pruebas Experimentales	71
3.5.2. Evaluación Ontológica y del Sistema Cognitivo Desarrollado	75
CONCLUSIONES	82
RECOMENDACIONES	84
REFERENCIAS	86
ANEXOS	100

ÍNDICE DE TABLAS

Tabla 1. Indicadores de calidad ontológica	15
Tabla 2. Indicadores de desempeño del sistema	16
Tabla 3. Análisis documental: "Reglamento del Sistema Nacional de Nivelación y Admisión-2024"	37
Tabla 4. Catálogo de servicios	41
Tabla 5. Especificaciones de la tabla categorías	48
Tabla 6. Especificaciones de la tabla c_subcategoría	49
Tabla 7. Especificaciones de la tabla conocimiento	50
Tabla 8. Especificaciones de la tabla conocimiento_embedding	50
Tabla 9. Especificaciones de la tabla interaccion_chatbot	51
Tabla 10. Especificaciones de la tabla bert_clasificacion	51
Tabla 11. Especificaciones de la tabla referencia_ontologica	52
Tabla 12. Especificaciones de la tabla feedback	52
Tabla 13. Especificaciones de la tabla users (de zammad)	53
Tabla 14. Especificaciones de la tabla tickets (de zammad)	53
Tabla 15. Resumen de clasificaciones de Bert antes de la ontología - 264 incidencias	72
Tabla 16. Resumen de clasificaciones de la ontología - 264 incidencias	73
Tabla 17. Muestra representativa de las intervenciones de la ontología	74

ÍNDICE DE FIGURAS

Figura 1. Fases para el desarrollo del proyecto.	17
Figura 2. Prefijos definidos en la ontología para el uso sus herramientas [42] [43] [44] [45].	23
Figura 3. Estructura principal del catálogo de servicios	38
Figura 4. Diagrama ontológico del catálogo de servicios visualizado en PROTÉGÉ.	42
Figura 5. Registros de la evolución del dataset de incidentes a lo largo de sus versiones	43
Figura 6. Comparación de las 6 versiones del dataset de incidentes por categorías	43
Figura 7. Registro de la evolución del dataset de soluciones	44
Figura 8. Comparación de la evolución del dataset de soluciones por servicios	45
Figura 9. Arquitectura del Sistema	46
Figura 10. Esquema de base de datos	47
Figura 11. Vista parcial de los skos:altlabel definidos de 4 servicios seleccionados.	54
Figura 12. Dataset de soluciones en formato tabular (tabla conocimiento)	54
Figura 13. Representación vectorial (embedding) de cada conocimientos almacenado.	55
Figura 14. Inicialización de la fase ontológica (carga en memoria los terminos de la ontología)	57
Figura 15. Fase de Validación - ontología	58
Figura 16. Primera sección del modal: Gestión del conocimiento	59
Figura 17. Segunda sección del modal: Enriquecimiento de la ontología	60

Figura 18. Autenticación para acceder al servicio del asistente virtual	61
Figura 19. Ingreso de consulta al sistema para obtener una solución.	62
Figura 20. Sistema muestra el contenido de la solución elegida	62
Figura 21. Calificación del servicio brindado	63
Figura 22. Nuevo incidente, no obtuvo una respuesta válida para su solución.	64
Figura 23. Incidente no solucionado, escalado a ticket de soporte	64
Figura 24. Información almacenada en la base de datos de la consulta y ticket generado	65
Figura 25. Interfaz de acceso al gestor de tickets Zammad	65
Figura 26. Panel principal de Zammad - Datos estadísticos	66
Figura 27. Visión general del estado de los tickets	66
Figura 28. Vista general de todos los usuarios ingresados	67
Figura 29. Interfaz para registro de un nuevo usuario	67
Figura 30. Interfaz para registro masivo de usuarios	67
Figura 31. Interfaz del ticket abierto.	68
Figura 32. Solución del incidente, actualización del estado del ticket a cerrado.	68
Figura 33. Interfaz del modal	69
Figura 34. Modal - interfaz de top 3 de soluciones similares encontradas	70
Figura 35. Modal - interfaz de retroalimentación de la estructura ontológica	70
Figura 36. Modal - previene el ingreso de términos repetidos a la estructura ontológica.	71
Figura 37. Vista 1 – Resultados prueba 1 de inferencia lógica con el razonador HermiT	75

Figura 38. Vista 2 - Resultados prueba 1 de inferencia lógica con razonador HermiT	76
Figura 39. Resultados de prueba 2 premeditada con razonador HermiT	76
Figura 40. Cobertura ontológica	77
Figura 41. Completitud descriptiva en la estructura ontológica	77
Figura 42. Concisión de la estructura ontológica	78
Figura 43. Proporción de interacciones con intervención ontológica	78
Figura 44. Distribución de interacciones por tipo de intervención de la ontología	79
Figura 45. Tasa de aceptación del desempeño del sistema	79
Figura 46. Hit@3 - desempeño del sistema	80
Figura 47. Métricas de calidad del ranking de Top 3 de respuestas entregadas	81
Figura 48. Métrica de recuperación semántica	81

RESUMEN

La presente propuesta de implementación de un sistema cognitivo para la gestión del conocimiento institucional en el proceso de nivelación y admisión universitaria tiene como propósito capturar, estructurar y retroalimentar la base de conocimiento durante la atención de incidentes tecnológicos de nivel 1. La gestión formal del conocimiento se sustenta bajo la adaptación de la metodología para la creación de ontologías de Uschold y King implementando modelos de clasificación como Bert y ontología, recuperación semántica y además un modal interactivo para retroalimentación del conocimiento.

Se evaluó el sistema cognitivo obteniendo una tasa de aceptación del 75%, y se evidencia como la intervención ontológica en la gestión del conocimiento mejora la clasificación de los incidentes en un 19% de los casos evaluados para la correcta recuperación del conocimiento. Con los resultados obtenidos esta propuesta demuestra como la gestión del conocimiento puede pasar de un proceso informal a uno formal y estructurado.

Palabras claves: Gestión del conocimiento, sistema neuro-simbólico, tecnologías innovadoras.

ABSTRACT

The purpose of this proposal to implement a cognitive system for institutional knowledge management in the university placement and admissions process is to capture, structure, and provide feedback to the knowledge base while addressing Level 1 technical incidents. Formal knowledge management is based on the adaptation of Uschold and King's methodology for creating ontologies, implementing classification models such as Bert and ontology, semantic retrieval, and an interactive module for knowledge feedback.

The cognitive system was evaluated, yielding a 75% acceptance rate, and it was demonstrated that the ontological intervention in knowledge management improves incident classification in 19% of the cases evaluated, thereby facilitating accurate knowledge retrieval. Based on these results, this proposal demonstrates how knowledge management can transition from an informal process to a formal and structured one.

Keywords: Knowledge management, neuro-symbolic system, innovative technologies.

INTRODUCCIÓN

La gestión del conocimiento en las instituciones de educación superior constituye un desafío estratégico especialmente en procesos de soporte tecnológicos donde el conocimiento generado en la atención de incidentes es rara vez capturado y reutilizado de manera sistemática, por lo que por cada incidente resuelto que genera un conocimiento operativo valioso que ante la ausencia de mecanismos automatizados permanece como un conocimiento tácito individual. Esta inestabilidad en la gestión del conocimiento institucional puede generar a futuro ineficiencias directas en tiempos de respuestas elevados, en procesos recurrentes, en la calidad de atención y en la carga operativa del personal de atención.

La gestión formal del conocimiento emerge como una vía que transforma el conocimiento disperso en un activo institucional estructurado, reutilizable y enriquecido constantemente. Una ontología formal se constituye como el instrumento más adecuado para estructurar el conocimiento del dominio en sistemas cognitivos para permitir la definición de los conceptos, jerarquías y relaciones semánticas que caracterizan el dominio, siendo una herramienta fundamental para la interpretación humana y de los sistemas computacionales.

En el proceso de nivelación universitaria la gestión del conocimiento presenta desafíos particulares como el volumen de consultas elevados, el vocabulario informal de ciertos usuarios, y el conocimiento acumulado en cada solución que no es formalizado, siendo esto consecuencia directa en la repetición de incidencias por solucionar y que podría desencadenar tiempos de esperas largos o cortos dependiendo la gravedad del incidente que presenten los estudiantes del proceso de nivelación y admisión. A partir de esta brecha se propone un sistema cognitivo de gestión del conocimiento que integra una metodología adaptada con cinco fases completas para la gestión correcta del conocimiento y sus resultados evaluados mediante métricas e indicadores de desempeño del sistema y la estructura ontológica, todo lo anterior permite demostrar que el proyecto mejora significativamente la consistencia y escalabilidad de la atención de incidentes tecnológicos en el proceso de nivelación y admisión universitaria.

CAPÍTULO 1. FUNDAMENTACIÓN

1.1. Antecedentes

El desarrollo y gestión del conocimiento se identifica como un elemento fundamental para el progreso organizacional, sin embargo en ciertas instituciones educativas se observa una escasa prioridad en su ejecución correcta [1]. Existe un estudio de 121 artículos que revela un alto nivel de heterogeneidad y una carencia de marcos teóricos sólidos lo que muestra que muchas instituciones aún no adoptan practicas efectivas de gestión de conocimiento [2]. La ausencia de estrategias para la gestión de conocimiento en instituciones de educación superior se presenta como un obstáculo que limita la creación, transferencia y retención del conocimiento institucional [3].

En el marco de este estudio se consideró una Institución de Educación Superior ubicado en la provincia de Santa Elena- Ecuador con más de 28 años de trayectoria, dedicada a la formación de futuros profesionales que impulsen el progreso sostenible y aporten soluciones a la sociedad, además de promover el fortalecimiento del talento humano dirigido a la sociedad en general manteniendo de esta forma un compromiso firme con el alto nivel formativo y permitiendo establecerse como una institución destacada [4].

En lo que respecta al departamento de Nivelación y Admisión de la institución mencionada, tiene la responsabilidad de coordinar, conservar y difundir información correspondiente al sistema académico institucional rigiéndose a las normativas establecidas por el SENESCYT [5]. Esta Unidad gestiona procesos para una población universitaria de aproximadamente 3000 estudiantes por periodo académico, la misma que para efectuar sus actividades se apoya en diversas herramientas tecnológicas, recursos que contribuyen potencialmente a la estructura interna de esta unidad respaldando la ejecución correcta de las directrices establecidas [5].

Mediante una entrevista realizada a la dirección de Nivelación y Admisión (ANEXO 1) se obtuvo que a pesar de dirigir y garantizar el correcto desarrollo del proceso de admisión y nivelación, el bajo conocimiento que tiene los estudiantes ante el uso de las herramientas tecnológicas implementadas en el curso de Nivelación y Admisión

constituye un gran desafío que puede verse reflejado en la demora del cumplimiento de las actividades académicas por parte de los estudiantes, esta situación es una de las consecuencias a raíz de la ausencia de asistencias inmediatas ante los contratiempos tecnológicos que pueden presentar los aspirantes.

Además, debido al desconocimiento de los procesos a seguir por parte de aquellos aspirantes que buscan ingresar a la educación superior podría provocar la deserción de los mismos a causa de la poca familiarización con los medios digitales que deben manejar para poder seguir cada fase del proceso de admisión, esto afecta de tal forma el principal objetivo de la unidad de nivelación y admisión, incluso de la Institución que es garantizar la continuidad y culminación de los estudios de cada uno de sus aspirantes.

De igual manera los docentes suelen presentar ciertos inconvenientes con la operatividad física o lógica de las herramientas tecnológicas que utilizan durante el proceso de nivelación y admisión para impartir sus enseñanzas lo que podría afectar a las mismas, dando como resultado el retraso de sus actividades laborales por no poder encontrar una solución inmediata antes los incidentes sucitados, esta situación crítica repercute en la programación de actividades y a su vez en la efectividad de la tutoría que ofrecen a los estudiantes en el transcurso del periodo académico.

Aunque el departamento de Nivelación y Admisión se encarga de solucionar todos los incidentes presentados sea de forma presencial o reportados por correo, no cuenta con un medio de soluciones ante problemas repetitivos y no poseen registros que les permita analizar la cantidad precisa de casos (incidentes) resueltos diaria o semanalmente, además de carecer de procedimientos para medir o evaluar la eficacia de la atención brindada en el departamento de Nivelación y Admisión, lo que podría desatarse a futuro como una debilidad en la gestión académica de Nivelación y Admisión.

En el contexto internacional, un estudio relevante es el artículo “Modelo de gestión inteligente de incidentes de TI basado en ontologías” en la universidad Addis Ababa (Ethiopia), la investigación propone un sistema orientado a mejorar la administración de incidentes tecnológicos utilizando ontologías para estructurar la

información y técnicas de NLP para interpretar el texto de los reportes, incorporado un motor inteligente que categoriza dichas incidencias y recomienda soluciones automáticamente, se integra en procesos ITSM siguiendo lineamientos de ITIL para mejorar el flujo de soporte técnico [6]. A pesar de que el artículo no está enfocado en el ámbito educativo, se puede tomar una guía conceptual en cuanto a la implementación para gestionar incidentes en entornos educativos.

Una investigación en el contexto Latinoamericano en la Universidad UFPS (Colombia), un proyecto que se centra en el diseño de una plataforma fundamentada en ontologías para la gestión de conocimiento con el objetivo de administrar conocimiento semántico a partir de datos enlazados (Linked Data) integrando mecanismos de filtrado, limpieza y validación de contenido, así como el desarrollo de los servicios web que permite realizar consultas del conocimiento [7]. Aunque, si bien este enfoque se asemeja al de esta propuesta, se diferencia en que no realiza gestión de incidentes tecnológicos ni logra proporciona recomendaciones automáticas.

Como antecedente local, se destaca un proyecto denominado “Desarrollo de una interfaz basada en algoritmos de procesamiento de lenguaje natural (NLP) para la gestión de conocimiento.” que tuvo como propósito la gestión automatizada del conocimiento, agilizando el acceso a la información utilizando técnicas de NLP, adoptando una base de datos e interfaz y el desarrollo de dataset simulado por ChatGPT, todo esto permitió medir correspondencias entre preguntas y soluciones permitiendo una interacción más intuitiva dando soluciones inmediatas que reducen los tiempos de atención [8]. A partir de las recomendaciones expuestas en este trabajo, la presente iniciativa adopta la incorporación de ontologías a fin de mejorar de esta forma la eficiencia en la resolución de incidentes tecnológicos.

Ante esta realidad, nace la presente propuesta del desarrollo de un sistema de base de conocimiento cognitiva que permite gestionar oportunamente los incidentes tecnológicos de nivel 1 presentados durante el proceso de nivelación y admisión universitaria. La propuesta se fundamenta en la creación de un modelo ontológico y el uso de herramientas de NLP y algoritmos de recuperación semántica para organizar la información facilitando la interacción entre estudiantes, docentes y el sistema de soporte.

La implementación de este proyecto se compone de tecnologías emergente que brindan soluciones inmediatas a problemas recurrentes, por medio de un sistema neuro-simbólico que influye en la reducción del tiempo de espera para dar solución a un problema junto a un gestor de tickets que complementa esta solución tecnológica, de esta manera la gestión del conocimiento efectiva fortalecerá cada proceso académico.

1.2. Descripción Del Proyecto

Esta investigación se centró en el desarrollo de un sistema de base de conocimiento cognitiva que permitió gestionar incidentes tecnológicos de nivel 1 en el proceso de Nivelación Universitaria. Debido al desafío que representa la expansión de contratiempos tecnológicos que enfrentan estudiantes y docentes en entornos virtuales, se estableció esta propuesta basada en la jerarquización de incidentes tecnológicos de nivel 1 dentro de una estructura ontológica elaborada a partir de la del catálogo de servicios del dominio [6]. A su vez permitió la formalización del dataset de incidentes que entrena un algoritmo de clasificación automática implementado en una interfaz de ayuda inteligente, esta solución optimiza la identificación, categorización y resolución de incidencias de forma efectiva.

En el ámbito de la gestión del conocimiento, un sistema cognitivo se define como un marco estructurado orientado a la gestión y organización de la información dentro de una organización, incluyendo métodos que guían la creación y mantenimiento del conocimiento [9]. Por lo tanto, un sistema que gestiona el conocimiento tecnológico permite asistir de manera más efectiva a los incidentes tecnológicos de nivel 1 que se presentan en el proceso de nivelación y admisión, conformándose como un sistema fundamental que optimiza el soporte técnico que se brinda a los estudiantes lo cual contribuye a la calidad y eficiencia del proceso educativo.

El enfoque metodológico adopta el modelo de la composición de esquemas ontológicos propuesto por Uschold y King (1995), este método se ajusta en su mayoría a los requerimientos de este estudio, por lo cual se optó por adaptar las fases del mismo para los cada proceso del proyecto como son la adquisición, conceptualización, formalización, implementación, validación y retroalimentación

del conocimiento, formalizando así un marco estructurado que facilita el seguimiento durante todo el proceso y posibilita la medición del desempeño del modelo [10].

Posteriormente, se exponen las fases elegidas para llevar a cabo el desarrollo del presente proyecto:

Fase de adquisición de Conocimiento:

La recopilación la información se llevará a cabo mediante una entrevista a la dirección del Departamento de Nivelación y Admisión con el fin de identificar información relevante del proceso de Nivelación y Admisión. Esta entrevista permite la construcción del catálogo de servicios tecnológicos del Departamento de Nivelación y Admisión, siendo la base fundamental para el desarrollo de este proyecto.

- Entrevista dirigida a la dirección del curso de Nivelación y Admisión.
- Análisis documental de normativa vigente del proceso de Nivelación y Admisión del periodo en curso.
- Elaboración del catálogo de servicios organizado en categorías y subcategorías.

Fase de conceptualización:

En esta etapa se construye el esquema lógico que organiza el conocimiento recopilado anteriormente representándolo en un diagrama conceptual por medio del modelo ontológico. Esto permite determinar los conceptos esenciales, niveles jerárquicos y las relaciones que las vinculan, con el propósito de establecer un modelo estructurado que sustente su posterior formalización y la creación de los datasets iniciales.

- Elaboración del modelo conceptual a través de diagramas ontológicos para formalizar los conceptos claves del dominio antes definidos en el catálogo de servicios.
- Creación de datasets (incidentes y soluciones) con datos sintéticos a partir de datos reales identificados durante el proceso de recopilación de información.

Fase formalización y desarrollo:

Para esta fase se pretende transformar el conocimiento conceptual en un modelo computacional inteligente, lo cual implica la organización de los datasets en formato tabular, definición de reglas lógicas, desarrollo del sistema neuro-simbólico para la clasificación automática y la recuperación semántica de soluciones. Además, incorpora un sistema automático que impide duplicaciones y fomenta la mejora continua de la base de conocimiento.

- Elaboración del esquema de base de datos relacional para registro de clasificaciones, interacciones y retroalimentación del conocimiento.
- Preparación de los datasets en formato tabular.
- Establecer reglas lógicas para la ontología que regulan la representación del conocimiento.
- Desarrollo del sistema neuro-simbólico mediante el uso del motor de clasificación automática Bert y motor de inferencia semántica ontológica.
- Recuperación semántica de soluciones mediante el modelo Sentence-Bert (S-Bert) y similitud del coseno para identificar el conocimiento almacenado más relevante ante una solución.
- Desarrollo de un modal para gestión del conocimiento que permite: guardar, actualizar y evitar duplicados. Y gestión de ontología que permite: agregar nuevos términos alternativos según el servicio elegido y evitar duplicados.

Fase de implementación:

La incorporación de los componentes tecnológicos que consolidan el sistema en un entorno funcional que sea capaz de percibir, procesar y responder incidentes tecnológicos de nivel 1 de manera automática o escalar dichos incidentes que no logren ser resueltos hacia un gestor de soporte, además de aplicar mecanismos que faciliten la actualización y retroalimentación constante del conocimiento.

- Uso del sistema neuro-simbólico por medio de un canal de comunicación que permita recibir, procesar y responder incidentes tecnológicos nivel 1.
- Implementación del gestor de tickets Zammad para atender incidentes que escalaron a ticket para su solución.
- Implementación del modal que permite gestionar el conocimiento y la ontología de manera automática.

Fase de evaluación y validación

El propósito de esta fase es comprobar la capacidad del sistema cognitivo implementado para responder correctamente a los incidentes presentados. Para esto, se aplican un método mixto (cualitativo y cuantitativo), que lleva a cabo las pruebas prácticas para comprobar la pertinencia de las soluciones, contribuyendo a la retroalimentación de manera dinámica en la base de datos.

- Pruebas experimentales: pruebas controladas del sistema en situaciones simuladas, validando que el conocimiento se retroalimenta en la base de datos.
- Evaluación de la ontología y del sistema cognitivo desarrollado con indicadores cuantitativos y cualitativos:

Indicadores de calidad ontológica:

- Consistencia estructural: evaluación de la estructura lógica de la ontología sin contradicciones.
- Cobertura del catálogo: % del catálogo de servicio se encuentra representado en la ontología.
- Completitud Descriptiva: % de clases de la ontología completas en sus descripciones.
- Concisión: que tan libre de redundancias está la ontología.
- Relevancia ontológica: que tan seguido la ontología corrige, complementa o valida correctamente.

Indicadores de desempeño de sistema:

- Precisión semántica (%): porcentaje de las soluciones entregadas cuantas son las que tuvieron una respuesta correcta según el incidente presentado.
- Hit@3 (%): Porcentaje de utilidad del sistema
- Tasa de Aceptación: cuantos usuarios eligieron al menos 1 solución del top 3 mostrado
- MRR: El Rango recíproco medio indica en promedio la posición en la que aparece la solución útil
- MAP: Posición media promedia calcula la consistencia del ranking en general del orden entregado de las soluciones

- nDCG: Ganancia acumulada descontada normalizada evalúa la calificación puesta por el usuario (1 bajo al 5 excelente) de las soluciones presentadas,

1.3. Objetivos Del Proyecto

Objetivo General:

Implementar un sistema cognitivo de gestión de conocimiento que integre técnicas de NLP/ML, modelado ontológico y la personalización de un software libre con verificación de similitud semántica, que permita resolver incidentes tecnológicos de nivel 1 en el proceso de Nivelación y Admisión universitaria.

Objetivos Específicos:

- Estandarizar el conocimiento del soporte mediante catálogo de servicios y taxonomía de incidentes/soluciones, a partir de entrevista dirigida y análisis documental del proceso vigente.
- Modelar el conocimiento del dominio mediante ontología y reglas lógicas, asegurando correspondencia semántica entre taxonomía, ontología y dataset.
- Programar la personalización del software libre Zammad incorporando una interfaz de registro/actualización de soluciones con verificación de similitud semántica basado en el modelo S-Bert que permita evitar duplicidad.
- Integrar el sistema cognitivo incorporando ontología, clasificador NLP/ML, base de conocimiento con verificador semántico para automatizar la respuesta y mantener una actualización controlada de soluciones.
- Evaluar la eficacia del sistema mediante pruebas experimentales controladas, comparando su desempeño entre una línea base sin soporte ontológico y una variante con soporte ontológico, utilizando métricas de precisión semántica y cobertura para validar su capacidad de respuesta.

1.4. Justificación Del Proyecto

La Gestión y Creación del conocimiento se constituye como una táctica esencial que permitirá a los procesos organizativos favorecer a la producción, difusión y preservación del conocimiento fomentando la continuidad de los procesos

educativos que mejorará la calidad académica [11]. De tal forma, la gestión del conocimiento no solo permite anticipar barreras, sino que también se aprovecha los recursos tecnológicos innovadores haciendo posible contar con ambientes tecnológicos más eficientes y sostenibles [12].

Respecto al departamento de Nivelación y Admisión de una Institución de Educación Superior, la aplicación de un sistema de base de conocimiento cognitiva que permita gestionar los incidentes tecnológicos nivel 1 resulta ser una contribución importante ante las actividades profesionales y académicas. De este modo, la innovación de la gestión conocimiento cognitivo aplicando inteligencia artificial se presenta como una herramienta imprescindible que impulsa una mejora de los servicios tecnológicos en ámbito académico, los procesos educativos y la experiencia de los usuarios con las herramientas tecnológicas [13].

La creación del sistema cognitivo propuesto en este proyecto permitirá al Departamento de Nivelación y Admisión agilizar la resolución de incidentes tecnológicos frecuentes, otorgando un repositorio de soluciones accesible en tiempo real reduciendo la atención presencial o por correo. Esta solución gestiona el registro, la organización y constante actualización del conocimiento apoyando positivamente a la mejora continua, estudios previos ponen en evidencia que la implementación de ontologías en interfaces conversacionales fortalece la gestión del conocimiento respaldando la efectividad del proyecto [14].

El acceso inmediato a una herramienta tecnológica con Inteligencia artificial capaz de dar guía para la resolución de incidentes tecnológicos nivel 1 contribuirá en la disminución los retrasos en las actividades académicas lo a su vez mejora el desempeño y continuidad de las mismas potenciando la eficiencia laboral y la experiencia de aprendizaje al minimizar las barreras tecnológicas tanto para estudiantes y docentes [15]. Estas tecnologías innovadoras implementadas en entornos universitarios muestra una minoría en la tasa de deserción gracias al acompañamiento tecnológico constante y accesible [16]. De esta manera, se asegura que el proceso de nivelación cumpla su propósito principal: garantizar la permanencia y culminación de la formación profesional.

La implementación de esta solución en el departamento de Nivelación y Admisión la Institución de Educación Superior, constituye un avance importante hacia la innovación tecnológica y la modernización de sus procesos formativos y administrativos [17]. La adopción de las tecnologías innovadoras que influyen en la de manera efectiva en la gestión de conocimiento refuerza la capacidad institucional para conservar y difundir conocimientos, generando a su vez un valor estratégico mediante la construcción de una cultura de mejora continua y sostenibilidad tecnológica [18]. Lo que dará como resultado una mejora en la atención de incidentes tecnológicos repetitivos del proceso de Nivelación y Admisión gracias al uso de las nuevas tecnologías.

La presente propuesta se orienta hacia el Plan de Desarrollo Ecuador No Se Detiene 2025- 2029 específicamente en los siguientes puntos:

- Eje Social

En concordancia con el objetivo 2: Potenciar las capacidades de la ciudadanía con acceso universal a una educación inclusiva de calidad, acceso a espacios de intercambio cultural y una vida activa [19]. Correspondiendo a la política 2.1, y objetivo 2.2.

- Eje Institucional

Relacionado al objetivo 8 Fortalecer la institucionalidad pública de forma eficiente, transparente y participativa; en cumplimiento de la política 8.3 [19].

1.5. Alcance Del Proyecto

Esta investigación se centra en el desarrollo de un sistema cognitivo que parte de una base de conocimiento integrada en taxonomía de servicios del dominio, estructurada mediante el uso de métodos que permiten la mejora continua del conocimiento institucional almacenado [20]. Este sistema neuro-simbólico permite el ingreso de incidentes tecnológicos de nivel 1 facilitando su clasificación y recuperación de sus respectivas soluciones aplicando técnicas de ML, ontologías y algoritmo de recuperación semántica [21]. La integración de estas tecnologías innovadoras mejora la precisión del sistema para la atención de incidentes y perfeccionando progresivamente el conocimiento institucional almacenado.

El alcance abarca la aplicación de un proceso metodológico estructurado en cinco fases iniciando por la fase de adquisición que recopila información relevante mediante entrevista y análisis documental realizado el 19 de junio del 2025. La fase de conceptualización permite definir los elementos claves del dominio, establecer relaciones entre ellos y elaborar datasets de incidentes tecnológicos de nivel 1 y sus respectivas soluciones. Posteriormente, en la formalización se implementa el algoritmo de clasificación automática junto a un filtro semántico lógico, además de la incorporación de un mecanismo de similitud semántica que contribuye a la calidad del conocimiento. Sistema que será implementado en un entorno de prueba que permitirá gestionar los incidentes tecnológicos de manera automatizada.

La evaluación se llevará a cabo mediante indicadores cuantitativos como la precisión semántica y la cobertura, así mismo, por medio de pruebas experimentales se evaluará el correcto funcionamiento de la atención automatizada de los incidentes tecnológicos de nivel 1 suscitados durante el proceso de nivelación y admisión universitaria.

1.6. Metodología Del Proyecto

1.6.1. Metodología de la Investigación

La iniciativa se aplicará en una Institución de Educación de Educación superior, concretamente en el departamento de Nivelación y Admisión que tiene como propósito gestionar el proceso académico de los aspirantes y el acceso a la vida universitaria [5]. Su gestión se alinea a las normativas establecidas por el SENESCYT, brindando servicios a una comunidad estudiantil aproximada a 3 mil estudiantes por proceso académico y convirtiéndose en un eje clave para la etapa pre universitaria [5].

El departamento dispone de una infraestructura digital elemental compuesta por plataformas oficiales, no obstante, el manejo de incidentes tecnológicos carece de procesos automáticos lo que impide la documentación, organización y resolución de casos repetidos, a partir del cual nace la presente propuesta con el fin de gestionar el conocimiento tecnológico del proceso de nivelación universitaria y brindar una atención inmediata a los incidentes tecnológicos repetitivos de nivel 1 que se presenten, este proyecto será evaluado mediante pruebas experimentales para comprobar la eficacia de su aplicación.

1.6.2. Beneficiarios del Proyecto

Los beneficiarios del presente proyecto son estudiantes y docentes que participan en el proceso de Nivelación Universitaria, quienes podrán recibir un soporte tecnológico más rápido y preciso ante incidentes de nivel 1 en entornos virtuales. De igual forma, el soporte institucional se verá favorecido mediante la automatización de procesos repetitivos, la institución obtendrá beneficios estratégicos al contar con una base de conocimiento estructurada que facilita la toma de decisiones, la mejora continua del servicio y el fortalecimiento de la eficiencia en la gestión académica.

1.6.3. Diseño y Alcance de la Investigación

Este estudio se basó en un diseño **cuasi-experimental** al incorporar una solución tecnológica en un entorno de prueba, aunque no se realiza una distribución aleatoria se pretende evaluar el cambio producido por la solución propuesta [22]. Con respecto al alcance del proyecto, se trata de un enfoque **exploratorio** que se orienta a descubrir necesidades en la gestión del conocimiento institucional permitiendo establecer marcos conceptuales y diseñar soluciones innovadoras para la gestión académica por medio de las tecnologías [22].

1.6.4. Tipo y Métodos de Investigación

La modalidad investigativa que se adopta está orientada hacia un análisis cuantitativo que pretende interpretar interacciones operativas dentro del proceso de nivelación institucional [23]. Esta estrategia cuantitativa permite estudiar situaciones en los ambientes académicos habituales explorando las perspectivas de los miembros de la comunidad del proceso de nivelación y admisión.

Considerando el método de investigación se aplica el método **sintético**, basado en la unión de los resultados del sistema para determinar si cumple con los criterios de calidad y desempeño esperados para solucionar incidencias tecnológicas [24], este método se pondrá a prueba utilizando técnicas de medición y seguimiento.

1.6.5. Población

Se entiende por población la totalidad de unidades de análisis que son objeto de estudio o intervención [25], para este caso la unidades de análisis corresponde a datos sintéticos de más de 4 mil datos que reflejan los incidentes tecnológicos

presentados, constituyendo una base fundamental para el entrenamiento y construcción del sistema neuro-simbólico.

1.6.6. Muestra

Para esta propuesta se opta por un muestreo no probabilístico por conveniencia como método que permite seleccionar una fracción de la población para según el propósito investigativo, debido a los requerimientos de este estudio para trabajar con registros pre procesados técnicamente viables [26]. La muestra representa al 20% del dataset generado, dicho porcentaje será destinado a la validación funcional del sistema, esta división técnica corresponde al desarrollo del sistema de aprendizaje supervisado donde un 80% de los datos se destina para el entrenamiento y el 20% a la validación.

1.6.7. Idea a Defender

La incorporación de soporte ontológico mejora la capacidad de respuesta del sistema para la correcta solución de los incidentes tecnológicos de nivel 1.

- **Variable:** Evaluar la eficacia de la ontología y del sistema cognitivo, a través de indicadores de calidad y desempeño:

Indicadores de calidad ontológica:

- Consistencia estructural
- Cobertura del catálogo
- Completitud Descriptiva
- Concisión
- Relevancia ontológica

Indicadores de desempeño del sistema:

- Precisión semántica (%)
- Hit@3 (%).
- Tasa de Aceptación
- MRR
- MAP
- nDCG

Indicadores de calidad ontológica	
Consistencia estructural:	Evaluación de la estructura ontológica por medio del razonador HermiT
Cobertura:	
Cobertura de categorías=	$\frac{\text{Categorías del catálogo en la ontología}}{\text{Total de categoría en el catálogo (BD)}} \times 100$
Cobertura de servicios=	$\frac{\text{Servicios del catálogo en la ontología}}{\text{Total de servicios en el catálogo (BD)}} \times 100$
Cobertura global=	$\frac{\text{Cobertura Categorías} + \text{Cobertura Servicios}}{\text{Total de servicios en el catálogo (BD)}} \times 100$
Complejidad Descriptiva:	
• Complejidad de nombres (CN)=	$\frac{\text{Clases con rdfs:label} \geq 1}{\text{Total de clases en la ontología}} \times 100$
• Complejidad de definiciones (CD)=	$\frac{\text{Clases con skos:definition} \geq 1}{\text{Total de clases en la ontología}} \times 100$
• Complejidad de Terminos (CT)= alternativos	$\frac{\text{Clases con skos:altLabel} \geq 1}{\text{Total de clases en la ontología}} \times 100$
• Complejidad Descriptiva = Global	$\frac{(\text{CN} + \text{CD} + \text{CT})}{\text{Total de clases}}$
Concisión:	$\frac{(0,1 - (\text{etiquetas duplicadas}) + \text{clases huérfanas})}{\text{Total de etiquetas (rdfs:label + skos:altLabel)}} \times 100$
Relevancia Ontológica:	$\frac{\text{Interacción donde la ontología actuó (corrigió/ complementó/ validó)}}{\text{Total de interacciones evaluadas en BD}} \times 100$

Tabla 1. Indicadores de calidad ontológica

Indicadores de desempeño del sistema:	
Precisión semántica=	$\frac{\text{Soluciones relevantes entregadas}}{\text{Total de soluciones presentadas (Top 3)}} \times 100$
Hit@3 =	$\frac{\text{Interacciones donde el usuario marcó fue útil}}{\text{Total de interacciones con feedback}} \times 100$

MRR-Mean Reciprocal Rank:	
•	$\text{Rango Recíproco} = \frac{1}{\text{Posición donde apareció la solución útil}}$
•	$\text{MRR} = \frac{\text{Suma de todos los rangos recíprocos por interacción}}{\text{Total de interacciones con feedback}}$
MAP-Mean Average Precision:	
•	$\text{Precisión por interacción} = \frac{1}{\text{Posición donde apareció la solución útil}}$
•	$\text{MAP} = \frac{\text{Suma de todas las precisiones por interacción}}{\text{Total de interacciones con feedback}}$
Ndcg@3:	
•	$\text{DCG@3} = \frac{\text{Calificación pos.1}}{\log_2 (1 + 1)} + \frac{\text{Calificación pos.2}}{\log_2 (2 + 1)} + \frac{\text{Calificación pos.3}}{\log_2 (3 + 1)}$
•	$\text{nDCG@3} = \frac{\text{Norzamlizacion dividida}}{\text{Ganancia acumulada del ranking perfecto (IDCG@3)}}$ Ganancia acumulada Ganancia acumulada obtenida (DCG@3)

Tabla 2. Indicadores de desempeño del sistema

1.6.8. Recolección de Datos

Técnicas e Instrumentos de Recolección de Datos

Para el desarrollo de este proyecto se consideraron métodos de recolección de datos empezando por una entrevista técnica que tiene la finalidad de explorar y receptar información importante en cuanto a las necesidades y dificultades tecnológicas que enfrentan los docentes y estudiantes durante el proceso de nivelación académica universitaria (ANEXO 1).

Se aplica un análisis documental del Reglamento del Sistema Nacional de Nivelación y Admisión, para conocer el proceso del curso de Nivelación y Admisión, junto al análisis de la entrevista realizada poder identificar el contenido del catálogo de servicios del departamento de Nivelación y Admisión que muestran los servicios tecnológicos disponibles (Tabla 3).

Además, se realizan pruebas de usuario que permite validar el desempeño del sistema cognitivo implementado en situaciones simuladas reales durante el proceso de nivelación académica universitaria, estas pruebas están orientadas a la

comprobación de la eficiencia de las respuestas generadas desde la base de conocimiento y la retroalimentación de la misma (Ver Anexo3).

1.7. METODOLOGÍA DE DESARROLLO

Para la elaboración de este proyecto se utilizará la metodología de desarrollo de ontologías propuesta por Uschold y King (1995) [27], la cual se ajustó a las necesidades del proyecto guiando cada proceso desde la adquisición y organización del conocimiento hasta la implementación práctica de todo lo desarrollado en cada fase. A continuación, se describen las 5 fases adaptadas (ver Figura 1):

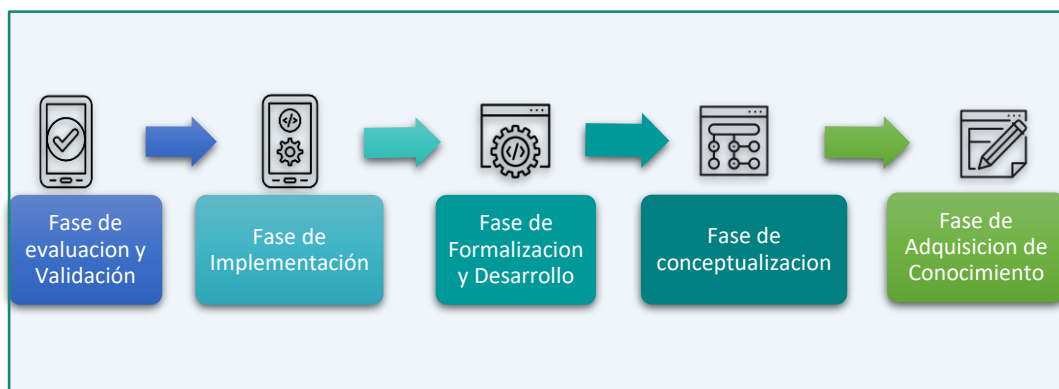


Figura 1. Fases para el desarrollo del proyecto.

1.7.1. Fase de Adquisición de Conocimiento:

En esta fase se llevará a cabo la recopilación de información inicialmente por medio de una entrevista realizada a la dirección del departamento de Nivelación y Admisión permitiendo identificar los incidentes tecnológicos más comunes dentro del proceso de académico, dando paso a la creación del catálogo de servicios que es la base fundamental para la ejecución del proyecto.

1.7.2. Fase de Conceptualización:

Para esta fase se realiza la representación lógica del catálogo de servicios (categorías y servicios) del dominio en diagramas ontológicos, identificando y definiendo conceptos claves que lo describen, junto a la definición reglas ontológicas entre sus jerarquías que establecen las relaciones entre los conceptos definidos anteriormente, esto posibilita la creación de los datasets de incidentes y soluciones de manera coherente de tal forma se pueda sustentar la formalización del mismo.

1.7.3. Fase Formalización y Desarrollo:

En este tercer paso se pretende lograr transformar el conocimiento conceptual a un modelo computacional inteligente, desde la preparación en formato tabular los dataset creados, el uso del algoritmo de clasificación automática “*Bert*” junto a la estructura semántica “*ontológica*” que fueron desarrollados en base al catálogo de servicios y al dataset de incidentes, además del desarrollo del modal de validación del conocimiento para determinar el destino de una nueva solución y a su vez permita alimentar la estructura ontológica de tal manera se logre mantener la base de conocimiento y la estructura ontológica del dominio constantemente actualizada y que la recuperación semántica se fortalezca.

1.7.4. Fase de Implementación:

La implementación de cada uno de los componentes técnicos que se desarrollaron hasta esta fase consolidan un sistema funcional permitiendo percibir, procesar y responder ante los incidentes tecnológicos de nivel 1 de manera inmediata y automatizada por medio de una herramienta virtual aplicando mecanismos de similitud semántica para llevar a cabo la retroalimentación constante del conocimiento por medio de un modal interactivo desarrollado e implementado dentro de una aplicación gestora de tickets Zammad.

1.7.5. Fase de Evaluación y Validación:

Para esta fase se lleva a cabo la comprobación y validación de la capacidad del sistema cognitivo creada e implementada para dar asistencia más efectiva ante incidentes tecnológicos. Para lo cual se aplican métricas para evaluar el comportamiento de la ontología como la consistencia estructural, cobertura del catálogo, completitud descriptiva, concisión y relevancia ontológica. De igual manera se implementaron los indicadores de desempeño para evaluar el sistema cognitivo desarrollado, acompañado de pruebas experimentales lo que permitió medir la eficiencia de las soluciones entregadas.

CAPÍTULO 2. PROPUESTA

2.1. MARCO CONTEXTUAL

Dentro del marco contextual de este proyecto se estableció en función de la gestión del conocimiento para una Institución de Educación Superior, tomando en cuenta como eje principal el conocimiento generado durante los procesos de nivelación y admisión. Se construyó un conjunto de información compuesta por incidencias y consultas recurrentes además de las soluciones para las mismas, que fueron modelados mediante ontología para poder representar el conocimiento del dominio. Mediante la estructura creada se facilitó la relación entre conceptos y la organización del conocimiento, asimismo contribuyó en la mejora de la precisión en la atención de requerimientos solicitados por los usuarios

Como entorno de aplicación se enmarcó dentro del contexto de gestión académica universitaria en asistencias técnicas durante los procesos de nivelación estudiantil, entorno caracterizado por la alta demanda de atención a usuarios y de consultas repetidas que sin una adecuada gestión podría afectar a la eficiencia operativa y de las actividades académicas. En este escenario la implementación de un sistema cognitivo basado en inteligencia artificial fortalece los mecanismos de atención mediante la automatización de respuestas a los usuarios, consolidándose como una estrategia esencial para el desarrollo continuo de las instituciones educativas en donde la correcta gestión del conocimiento destaca como un enfoque maestro para la mejora de los servicios ofrecidos.

2.2. MARCO CONCEPTUAL

2.2.1. Conocimiento Organizacional

Información

Es la resultante del proceso, organización y estructuración de datos que adquieren significado dentro del dominio perteneciente [28]. En el contexto tecnológico, la información se genera a partir de reglas o modelos que transformen datos en contenido comprensible, los datos adquieren valor haciéndolos útiles para la toma la toma de decisiones y la generación de conocimiento organizacional [28].

Conocimiento

El conocimiento se constituye de información, contexto, procesos mentales y experiencias que permiten la comprensión de una realidad [29]. En un enfoque organizacional el conocimiento es un recurso importante para la innovación y competitividad, la correcta gestión del conocimiento facilita la resolución de problemas y la mejora continua de procesos [29].

Conocimiento Tácito

Basado en experiencia personal, habilidades y competencias adquiridas a lo largo del tiempo lo que lo convierte en un conocimiento difícil de formalizar ya que se relaciona directamente a la práctica y contexto individual [30]. El conocimiento tácito es esencial para la innovación y resolución de problemas a pesar del desafío que representa su captura y transferencia [31].

Conocimiento Explícito

Es aquel que puede ser documentado y almacenado como textos manuales o en bases de datos lo que permite compartirlo fácilmente entre individuos y organizaciones lo que lo convierte en una base fundamental para la creación de sistemas informáticos ya que posibilita la normalización de tareas y la transmisión del conocimiento del contexto corporativo [32]

Gestión del Conocimiento

Es un conjunto de procesos que se rigen a capturar, organizar, almacenar, difundir y la actualización del conocimiento con el fin de procesar datos sueltos para generar conocimiento valioso, esta formalización del conocimiento tácito en explícito facilita la resolución de problemas dentro de las instituciones [33]. La literatura señala que la gestión del conocimiento permite optimizar el desempeño organizacional promoviendo la innovación y respalda las decisiones estratégicas gracias a la integración de las Tic [34].

Base de Conocimiento Cognitiva

Se trata de un sistema avanzado que gestiona información de forma sistemática implementando motores de inferencias y razonamiento automatizado permitiendo extraer conocimiento implícito y explícito, lo que demuestra que en el presente

proyecto las bases de conocimiento cognitivas soportan procesamiento inteligente y mejora continua lo que las diferencias de repositorios estáticos tradicionales [35].

Ciclo de Vida del Conocimiento

Dentro del contexto se describen etapas por las cuales el conocimiento es creado, almacenado, compartido y aplicado dentro de una institución permitiendo comprender como puede ser gestionado de manera eficaz [36]. Cada una de sus fases es clave para que el conocimiento permanezca actualizado, gestión que ayuda a maximizar el uso del conocimiento institucional [36].

2.2.2. Representación del Conocimiento

Ontologías

Las ontologías son modelos conceptuales que definen elementos y sus interconexiones de un área temática lo que establece un vocabulario común entre humano y máquinas, con la función de integrar sistemas, procesamiento lógico y la búsqueda conceptual [37]. Algunos estudios demuestran que las ontologías son primordiales en los procesos de estructuración de un contexto con múltiples relaciones y pueden ser combinadas con embeddings para el enriquecimiento de la representación vectorial del conocimiento [37].

Ontología de Dominio

Estructura conceptual que representa de manera formal el conocimiento donde se establecen conceptos, relaciones y restricciones focalizada en un área específica, con el propósito de capturar y representar de manera íntegra los elementos esenciales que componen la estructura del dominio [38]. Facilita la reutilización y automatización del conocimiento mejorando la calidad de los sistemas informáticos proporcionando una estructura conceptual coherente [39].

Ontología de Aplicación

La ontología de aplicación es la ramificación de una ontología de dominio siendo un modelo semántico especializado para la resolución de problemas dentro de un contexto específico [40]. Moldeada por las variables de entorno de una aplicación

dada incluyendo reglas y restricción que responden a requerimientos puntuales contribuyendo a mejorar la eficiencia del procesamiento de información [41].

RDF (Resource Description Framework)

RDF (Figura 2) se define técnicamente como una estructura basada en grafos de la web semántica para representar información en la web mediante tripletas sujeto-predicado-objeto que permiten describir los recursos y las relaciones entre estos para que las máquinas puedan procesarlas, siendo útil para una matriz organizativa de donde parte la construcción de lenguajes como RDFS y OWL [42].

RDFS (RDF Schema)

RDFs (Figura 2) una extensión de RDF que permite un vocabulario para describir clases, subclases, propiedades y restricciones de dominio permitiendo establecer jerarquías taxonómicas entre conceptos (mediante `rdfs: subClassOf`) y designar etiquetas a los recursos (mediante `rdfs: label`) estableciendo el modelado de esquemas previo a OWL [43].

OWL (Web Ontology Language)

Lenguaje de ontología web (Figura 2) es una herramienta formal para el desarrollo y representación ontológica en la web semántica estandarizado por el Consorcio World Wide Web (W3C), permitiendo modelar la estructura del conocimiento y las relaciones entre sus conceptos en un contexto definido, además de utilizarse para construir ontologías posibilitando la comunicación y compatibilidad entre tecnologías, su aplicación es ampliamente utilizada en sistemas de inteligencia artificial y gestión del conocimiento [44].

SKOS

SKOS (Figura 2) es una recomendación del W3C que permite un vocabulario estándar construido sobre RDF, permite la representación de esquemas del conocimiento organizacional tales como: esquemas de clasificación, sistemas de encabezamientos de materias, taxonomías y tesauros en un formato compatible con la web semántica, su uso de manera independiente así como la posibilidad implementar su combinación con otros lenguajes formales de representación semántica como la ontología (OWL) [45].

Ontology prefixes:	
Prefixes	+
:	http://www.semanticweb.org/ontologies/2025/nivelacion#
owl:	http://www.w3.org/2002/07/owl#
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
skos:	http://www.w3.org/2004/02/skos/core#

Figura 2. Prefijos definidos en la ontología para el uso sus herramientas [42] [43] [44] [45].

Taxonomía

Modelo estructural que permite clasificar jerárquicamente en categorías y subcategorías el conocimiento, utilizadas en diversos campos para organizar contenidos de manera coherente ayudando a reducir la ambigüedad y mejorar la consistencia en la clasificación de datos, lo que la lleva a optimizar procesos de búsqueda y análisis [46].

Dataset

Un dataset se constituye como una colección de datos organizados listos para su análisis y procesamiento o incluso entrenamiento de modelos tecnológicos inteligentes, se presentan en formatos tabulares que permiten su manipulación por medio de herramientas computacionales y su correcta preparación influye directamente en los resultados obtenidos [47].

Dataset Categorizado

Es una colección de datos estructurados por tipos o etiquetas previamente establecidas, esto permite facilitar las tareas como minería de datos aprendizaje de automático y exploración semántica, un dataset bien estructurado facilita la automatización el entrenamiento de modelos y la recuperación eficiente de la información [48].

Sistema Conversacional Basado en Recuperación

Se lo define como un sistema de diálogo automatizado que responde a las consultas del usuario seleccionando entre un conjunto de respuestas almacenadas aquella que resulta más relevante según criterios de similitud semántica o léxica, su enfoque resulta muy útil en contextos de soporte técnico donde las respuestas deben ser confiables y previamente verificadas [49].

Modelo computacional inteligente

Sistema basado en técnicas de inteligencia artificial capaz de lograr procesar información, aprender patrones y tomar decisiones de forma automática para la solución de problemas, mismos que podrían integrar algoritmos de NLP, aprendizaje automático y representaciones del conocimiento, combina componentes capaces de interpretar solicitudes recuperar soluciones pertinentes y adaptación continua [50].

Métricas de calidad

Criterio cuantitativo y cualitativo que implica la evaluación de un sistema, su aplicación se convierte en un método fundamental para garantizar que la arquitectura sobre la que opera el sistema sea confiable coherente y representativa del dominio que pretende automatizar [51].

2.2.3. Contexto Institucional

Proceso de Nivelación y Admisión Universitaria

Actividades institucionales orientadas a diagnosticar y asignar niveles adecuados a los estudiantes entrantes, comprende etapas claves como análisis de conocimientos previos, reparación académica verificación de competencias y admisión oficial, todas estas etapas se apoyan en componentes tecnológicos para los estudiantes que enfrentan barreras técnicas [52].

Incidentes Tecnológicos de Nivel 1

Se trata de inconvenientes básicos que afectan la interacción de los usuarios con las herramientas tecnológicas que utilizan, pueden manifestarse como bloqueos de cuenta configuraciones incorrectas, mal uso de funciones y entre otras formas, ante los cuales se lograr una solución mediante guías básicas sin necesidad soporte avanzado [53].

ITIL V4

Marco de buenas prácticas para la gestión de servicios de tecnología de la información, con el fin de mantener y mejorar el conocimiento organizacional con el objetivo de garantizar que la información esté disponible, confiable y se actualizándose constantemente para brindar apoyo en la toma de decisiones,

posicionándose como un marco adaptable en entornos digitales complejos gracias a la incorporación de conceptos metodológicos ágiles [54].

COBIT 5

Se define como un marco de gobernanza y gestión de TI empresarial que proporciona un conjunto estructurado de objetivos de control y métricas que les permite a las instituciones gestionar los riesgos tecnológicos que se presenten [55]. Este estándar también facilita la definición de roles, responsabilidades y flujos de información que son necesarios para que las tecnologías cumplan efectivamente los objetivos estratégicos del negocio y misión de la institución [55].

Catálogo de servicios

El catálogo de servicios se define como un instrumento de gestión organizacional que formaliza la estructura organizativa con información precisa de los servicios de una institución que los clasifica en categorías y subcategorías(servicios), incluye detalles importantes como el estado activo o inactivo, encargado, el nivel y las dependencias de los servicios, lo que permite convertir el conocimiento tácito de la institución en información explícita . [56]

2.2.4. IA y NLP

Tecnologías innovadoras

Se define como tecnología innovadora a aquellas tecnologías que al ser aplicadas en combinación o en contextos no explorados antes pueden con el potencial de transformar o generar cambios de manera beneficiosa ante problemas específicos, lo que optimiza resultados y supera las limitaciones de los enfoques tradicionales previamente utilizados generando impacto en los sectores aplicados [57].

Machine Learning

El aprendizaje automático (ML) es una rama de la inteligencia artificial que permite a los sistemas reconocer irregularidades y activar en función de la información disponible, estos modelos se entrenan con datos históricos ya que se basa en el aprendizaje por experiencia, como por ejemplo cuanto a la educación y soporte técnico se realiza clasificación de incidentes, predicción de necesidades y recomendación automática [58].

Natural Language Processing

El procesamiento de lenguaje natural (NLP) consiste en técnicas que permiten a los algoritmos interpretar y construir lenguaje humano de manera organizada incluyendo varias actividades como tokenización, análisis de su estructura, creación de embeddings y generación textual [59]. En sistemas de asistencia técnica el NLP podría interpretar preguntas de los usuarios en conceptos ontológicos y proporcionar soluciones adecuadas [59].

Limpieza de Texto

La limpieza del texto es una etapa muy importante para cualquier proceso de análisis de datos, este método consiste en el proceso que detecta y elimina el ruido, es decir corrige los textos aplicando técnicas como transformar el texto con letras minúsculas, la eliminación de tildes, números, caracteres no alfabéticos, así como excluir palabras vacías o los denominados “stopwords” con el fin de mejorar la validez del texto [60].

Bert

Bert es un modelo de procesamiento de lenguaje natural basado en arquitectura transformer aplicando un método de atención bidireccional profunda lo que permite que cada palabra sea representada en función de todo su contexto [61]. Lo que permite tomar en cuenta los aportes de modelo Bert adaptado al idioma castellano variante a la que denominaron “Beto”, demostrando mejoras significativas respecto al modelo original en tareas de comprensión de este lenguaje [62].

Embedding Semántico

El embedding semántico es la representación vectorial de un texto en un formato de múltiples números en diversas dimensiones donde la posición y dirección del vector capturan el significado del contenido de un texto, dichos contenidos con significados similares podrían quedar representados por vectores muy cercanos entre sí incluso si ciertas palabras son diferentes [63].

Sentence-bert (S-Bert)

S-Bert es la adaptación del modelo Bert, entrenada puntualmente para la generación de embeddings en representación del significado completo de una oración en vez

de hacerlo solo para palabras individuales con el fin de conservar sus significados semánticos para que de esta forma sea útil en la comparación directa de significados entre oraciones [64].

Similitud del Coseno

Métrica utilizada para cuantificar el grado de similitud entre vectores calculando el coseno del ángulo formado entre ellos en un espacio multidimensional, siendo de mucha utilidad en sistemas de recuperación de información y procesamiento de lenguaje natural para la comparación de representaciones en embeddings de textos y definir qué tan parecidos son semánticamente dos fragmentos [65].

Fuzzy Matching

Fuzzy matching o también denominado coincidencia difusa permite comparar cadenas de texto que presenten discrepancias menores como faltas ortográficas o variaciones como errores en la escritura, y su funcionamiento se apoya en algoritmos de distancia de edición como el de Levenshtein que permite asignar un grado de similitud entre dos textos para determinar si corresponden al mismo concepto aun si no son exactamente iguales [66].

Sistema Neuro-simbólico

Se entiende como sistema neuro-simbólico aquella disciplina con un enfoque híbrido de inteligencia artificial con capacidad de razonamiento más similares a las humanas, dicho sistema integra razonamiento simbólico que se basa en reglas o representación explícita del conocimiento que se une al aprendizaje automático, lo que propone una metodología de principios que superan las limitaciones individuales de los modelos puramente conexionistas o puramente simbólicos [67].

Algoritmo de Similitud Semántica

Proceso automatizado que analiza representaciones semánticas para estimar la afinidad de contenidos, entre estos procesos se destacan medidas como la similitud del coseno o la distancia euclidiana entre otros que cuantifican la proximidad semántica entre 2 vectores [68]. Los algoritmos son imprescindibles para la detección de redundancias, la similitud semántica y la organización conceptual, lo que resulta factible mantener una base de datos actualizada [69].

Métricas de Desempeño

Indicadores cuantitativos utilizados para medir la eficiencia y efectividad de los componentes de un sistema frente a situaciones de pruebas de uso, se centran en los resultados observables que produce el sistema, mide aspectos como precisión, nivel de satisfacción que califican los usuarios permitiendo validar si el sistema cumple su propósito [70].

2.2.5. Herramientas Tecnológicas

Api Rest

Conjunto de reglas y convenciones que permiten la comunicación entre sistemas mediante solicitudes http estándar como get, post, put, y delete, representan operaciones sobre recursos identificados mediante urls mientras que su adopción se extiende en sistemas distribuidos debido a que no requiere mantener estado entre solicitudes lo cual simplifica la integración de servicios independientes [71].

FastApi

FastApi se define como un framework web moderno de alto rendimiento para la construcción de Apis en Python basado en el estándar ASGI (Asynchronous Server Gateway Interface) que le permiten brindar un alto rendimiento en velocidad de desarrollo y reducción de errores, así como la disposición nativa para la gestión de peticiones simultaneas [72].

Sistema de Tickets

Solución tecnológica que permite gestionar incidentes desde la creación de una solicitud de soporte hasta su solución permitiendo escalar el incidente reportado y asignarlo a un agente para su atención [73]. Su propósito prioriza los casos urgentes y mejorar la calidad del servicio de soporte convirtiéndose en un sistema clave para la gestión de grandes volúmenes de atención técnica de manera ordenada [74].

Zammad

Plataforma de gestión de tickets de código libre, diseñada con el fin de centralizar las operaciones del área de soporte técnico y atención al cliente, ofrece funciones de asignación de tickets, priorización, automatización de tareas repetitivas y puede

integrarse a herramientas externas como APIs y permite autenticación con proveedores externos como google, Microsoft [75].

PostgreSQL

Sistema de Gestion de base de datos relacional orientado a objetos, reconocido por su robustez, escalabilidad y funciones avanzadas lo que le permite almacenar, organizar y recuperar datos estructurados, se caracteriza por su destacada extensibilidad y el cumplimiento con los estándares del lenguaje SQL garantizando la integridad de la información almacenada [76].

Pgvector

Módulo adicional para PostgreSQL que habilita el almacenamiento nativo de vectores numéricos de alta dimensión y la ejecución de consultas de vecinos más cercanos sobre dichos vectores [77], lo cual hace que esta extensión resulte imprescindible en sistemas que combinan datos relacionales tradicionales con representaciones semánticas generadas por modelos de lenguaje ya que evita la necesidad de infraestructura adicional para la búsqueda vectorial [78]

Python

Un lenguaje de programación de alto nivel, interpretado y de tipado dinámico, además soporta la programación orientada a objetos imperativa y funcional cuyo propósito es facilitar su aprendizaje gracias a su sintaxis sencilla por lo que la convierte en una de las tecnologías ampliamente utilizadas en el ámbito académico como herramienta introductoria a la programación [79].

JavaScript

Lenguaje de programación interpretado, dinámico y multiparadigma, se define como orientado a objetos mediante prototipos y programación funcional, es implementado principalmente por ser imperativo, dinámico y lógico para las páginas, aplicaciones web o incluso para servidores. En cuando al desarrollo web JavaScript complementa 3 pilares fundamentales junto a HTML y CSS [80].

Protégé

Software de código abierto desarrollado por la universidad de Stanford, sirve para la construcción de bases de conocimiento semántico “*ontologías*”, específicamente

permite la definición de clases, subclases, propiedades y etiquetas, además cuenta con plugins como razonadores, visualizadores, consultores, importadores, exportadores y editores de reglas lógicas [81]

HermiT

Razonador ontológico que actúa como un motor lógico para calcular inferencias dentro de una estructura ontológica, lo que le permite verificar la consistencia de la ontología, es decir verifica que no haya clases con definiciones lógicas contradictorias, la coherencia en la jerarquía de sus clases y que las relaciones entre sus conceptos sean lógicamente válidas [82].

2.3. MARCO TEÓRICO

2.3.1. Gestión del Conocimiento en el Soporte Técnico Institucional

La teoría de gestión del conocimiento consiste en administrar el conocimiento institucional para que sea útil y accesible dentro de la organización, el conocimiento debe ser tratado como un recurso estratégico para ser gestionado como un activo organizacional más [83]. Este enfoque destaca lo importante que es estructurar y formalizar el conocimiento para su acceso y aplicación para la toma de decisiones considerando que dicho conocimiento se encuentra tanto en documentos como en percepciones individuales [84].

La gestión del conocimiento se basa en la creación de un entorno organizacional que fomenta el aprendizaje continuo y la innovación desarrollándose a través de un conjunto de procesos que transforman el conocimiento individual en colectivo permitiendo la transferencia del conocimiento siendo un aspecto clave para compartir experiencias y mejores prácticas entre los usuarios de la organización [85]. En el ámbito educativo superior se traduce en gestionar los saberes académicos y científicos que fortalezcan cada proceso educativo para mejorar la enseñanza el aprendizaje y la innovación, aplicando esta teoría de gestión del conocimiento de educación impulsa mejora el rendimiento académico, la innovación pedagógica y la excelencia institucional [86].

Hislop, Bosua y Helms argumentan que los sistemas de gestión del conocimiento que incorporan mecanismos de retroalimentación sobre la calidad del conocimiento

almacenado superan en efectividad a los que tratan el conocimiento como un activo estático, porque la evaluación continua de la utilidad de cada pieza de conocimiento permite depurar el repositorio, priorizar actualizaciones y detectar vacíos de conocimiento que el sistema aún no cubre [87]. Esto sustenta la incorporación del método de calificación de satisfacción del usuario como mecanismo de mejora continua en sistemas de soporte técnico basados en gestión del conocimiento.

2.3.2. Ingeniería Ontológica y Representación Formal del Conocimiento de Dominio

Una ontología es una especificación explícita de una conceptualización argumentando que para que los sistemas computacionales puedan razonar sobre el conocimiento de un dominio es necesario formalizar sus conceptos, propiedades y relaciones en un lenguaje con semántica precisa y sin ambigüedades interpretativas [88]. Las ontologías permiten representar jerarquías taxonómicas, restricciones lógicas entre conceptos y relaciones de especialización que posibilitan la inferencia automática de nueva información a partir del conocimiento declarado superando ampliamente las capacidades de estructuras más simples como listas de términos o taxonomías planas [89].

La metodología de ingeniería ontológica propuesta por Uschold y King establece que la construcción de ontologías debe seguir un proceso sistemático que comprende la identificación del propósito y alcance, la captura de conceptos mediante consulta a expertos del dominio, la codificación formal en un lenguaje ontológico estándar y la evaluación de consistencia lógica y adecuación al dominio modelado [27]. Se ha demostrado que el lenguaje owl es el estándar más adecuado para este proceso porque permite expresar restricciones lógicas complejas que los motores de inferencia pueden procesar automáticamente y porque su interoperabilidad con otros estándares de la Web Semántica facilita la integración con sistemas externos [90].

Ontologías a escala industrial en empresas como Google y Microsoft que demuestran que las organizaciones que formalizan su conocimiento de dominio en ontologías desarrollan infraestructuras de conocimiento extensibles y que se pueden mantener de forma independiente al resto del sistema, lo que les permite actualizar

la representación del conocimiento sin necesidad de modificar los modelos de inteligencia artificial que la utilizan [91]. Esta evidencia valida teóricamente el uso de ontologías de dominio como capa de conocimiento institucional que puede evolucionar de forma autónoma ante cambios en los servicios de la organización.

Un estudio que combina el razonamiento de ontologías con modelos de aprendizaje automático demuestra que los sistemas híbridos que combinan razonamiento con estas 2 estrategias de clasificación alcanzan una precisión superior a la de cualquiera de los dos enfoques por separado en tareas de clasificación sobre cualquier dominio lo que permite sobrellevar los desafíos en datos con alta demanda predictiva [92]. El razonamiento ontológico aporta el conocimiento explícito del dominio que los modelos estadísticos no pueden adquirir del lenguaje natural, mientras que los modelos estadísticos aportan la capacidad de generalizar sobre lenguaje libre que las ontologías por sí solas no poseen generando una sinergia lo que teóricamente justifica el uso combinado de estas tecnologías en sistemas de clasificación de consultas de soporte técnico en un ambiente académico [93].

2.3.3. Modelos de Lenguaje y Recuperación Semántica del Conocimiento

Arquitectura Transformer basada en el mecanismo de auto atención que pondera simultáneamente la relevancia de todas las palabras de una secuencia entre sí, supera significativamente a las arquitecturas recurrentes previas en tareas de comprensión del lenguaje natural, estableciendo un nuevo paradigma en el procesamiento de lenguaje natural conecta palabras distantes sin perder información en el camino [94].

El preentrenamiento bidireccional de modelos Transformer sobre grandes corpus de texto no etiquetado produce representaciones del lenguaje tan ricas que pueden adaptarse a tareas específicas de dominio con una cantidad mínima de ejemplos etiquetados, estableciendo el prototipo de modelos de lenguaje preentrenados y ajustables que ha definido desde su publicación [95]. Bert desarrolla representaciones semánticas profundas del contexto lingüístico que le permiten distinguir el significado de palabras ambiguas según el contexto que las rodea validando su uso para clasificación de texto en dominios especializados [96].

Existe el estudio de realización de pruebas de redes siamesas basadas en Bert sobre pares de oraciones con etiquetas de similitud produce embeddings de oraciones completas que son directamente comparables entre sí mediante similitud del coseno, resolviendo el problema de eficiencia que impedía usar Bert original para búsquedas semánticas en tiempo real [97]. Mediante la destilación del conocimiento es posible comprimir estos modelos manteniendo más del 99% de su capacidad semántica con una fracción del costo computacional, haciendo viable su despliegue en servidores con recursos limitados [97].

Malkov y Yashunin (2020) probaron matemáticamente que el algoritmo HNSW permite realizar búsqueda aproximada del vecino más cercano en espacios vectoriales de alta dimensión con complejidad logarítmica y precisión superior al 95% respecto a la búsqueda exacta, fundamentando teóricamente la viabilidad de la búsqueda semántica en tiempo real sobre grandes bases de conocimiento [98]. El aprendizaje automático es una rama de la inteligencia artificial que capacita los sistemas para evolucionar a partir de datos sin programación directa, en la gestión del conocimiento se facilita el análisis inteligente de información y la generación de predicciones útiles, contribuyendo a que los sistemas se ajusten dinámicamente y ofrezcan respuestas personalizadas según el contexto de la organización [99].

2.3.4. Arquitecturas de Software Orientadas a Servicios para Sistemas Inteligentes Conversacionales

La arquitectura de micro servicios son la evolución de las arquitecturas orientadas a servicios, se garantiza que los sistemas complejos sean más flexibles, mantenibles y escalables para componentes individuales según su carga específica y facilita la evolución tecnológica progresiva sin necesidad de reescribir el sistema completo [100]. Bass, Clements y Kazman (2021) fundamentan este ejemplar durante décadas de investigación en arquitectura de software, demostrando que los sistemas con bajo acoplamiento entre componentes presentan tiempos de mantenimiento significativamente menores y mayor capacidad de adaptación ante cambios de requerimientos [100].

Modelos de aprendizaje automático según sus teorías consolidan con el auge del despliegue de sistemas de inteligencia artificial en producción, establece que los

modelos entrenados deben exponerse como servicios de inferencia independientes accesibles mediante APIs REST, separando la lógica de predicción del resto de las aplicaciones consumidoras [101]. Esta separación combinada con el patrón Singleton para mantener el modelo cargado en memoria durante todo el tiempo de vida del servidor, resuelve la tensión entre el alto costo de inicialización de los modelos de lenguaje y la baja latencia requerida en aplicaciones interactivas convirtiéndola en una solución arquitectónica que hace viable el uso de modelos de lenguaje profundos en sistemas de soporte en tiempo real [102].

La integración de modelos de lenguaje con bases de conocimiento estructuradas representa el prototipo más efectivo para sistemas conversacionales de soporte institucional ya que combina la flexibilidad del lenguaje natural con la precisión del conocimiento validado evitando las suposiciones y respuestas incorrectas que producen los modelos generativos sin acceso a fuentes de conocimiento verificadas [103]. Esto confirma la eficacia de la arquitectura de sistemas que separan el modelo de comprensión lingüística del componente de recuperación de conocimiento y estructura semántica, conectándolos mediante interfaces bien definidas que les permitan actualizarse de forma independiente sin comprometer la calidad del sistema completo [104].

CAPITULO 3. COMPONENTES DE LA PROPUESTA

3.1 FASE DE ADQUISICIÓN DEL CONOCIMIENTO

3.1.1 Análisis de Recolección de Datos

Mediante la entrevista realizada a la dirección del departamento de Nivelación y Admisión (ANEXO 1) se logró comprender el proceso del curso de Nivelación y Admisión de manera técnica, destacamos sus herramientas tecnológicas que sirven de medio para clases y comunicación. No obstante, el uso de estas herramientas tecnológicas evidencia los incidentes se presentan durante el proceso de Nivelación y Admisión lo que podría afectar a futuro ante el desempeño del mismo.

Uno de los puntos más relevantes identificado durante esta entrevista es que los estudiantes pueden enfrentar complicaciones en el proceso de adaptación del uso de las herramientas tecnológicas del curso de Nivelación Universitaria, dicho inconveniente tecnológico limitaría la participación de los estudiantes durante las actividades del curso, provocando retrasos y a futuro se puede ver reflejado en la dificultad para la continuidad de sus estudios.

Aunque el departamento de Nivelación y admisión se encarga de solucionar todos los incidentes presentados sea de forma presencial o reportados por correo, no cuenta con un medio de soluciones ante problemas repetitivos. Esto dificulta analizar la cantidad precisa de casos (incidentes) resueltos diaria o semanalmente, además carece de procedimientos para medir o evaluar la eficacia de la atención brindada en el departamento de Nivelación y Admisión, lo que podría desatarse a futuro como una deficiencia en la gestión académica de Nivelación y Admisión.

A partir de este análisis se puede concluir que los retos tecnológicos dentro del curso de Nivelación y Admisión podrían afectar a futuro la efectividad del proceso educativo debido a la falta de mecanismos automáticos que faciliten la atención y solución inmediata a los incidentes tecnológicos repetitivos, además de la poca formación digital con la que ingresan los estudiantes. Ante esta realidad nace la iniciativa de incorporar tecnologías innovadoras que fortalezcan el desempeño académico y la gestión del conocimiento institucional.

3.1.2. Análisis Documental

Fecha: 19/06/2025

- **Ficha de Análisis Documental**

Campo	Descripción
N°	01
Documento Analizado	Reglamento del Sistema Nacional de Nivelación y Admisión
Tipo de Documento	Codificación/ Normativo Institucional
Autor / Institución	SENESCYT
Año	2024
Fuente	https://www.registrounicoedusup.gob.ec/
Objetivo del Documento	El presente reglamento tiene por objetivo gestionar el proceso para el acceso a la educación superior pública de los aspirantes, según el marco del Sistema Nacional de Nivelación y Admisión. Así mismo se detalla como determinar el procedimiento para el registro de los cupos que se otorguen en instituciones de educación superior a nivel nacional de conformidad con lo establecido en la Ley Orgánica de Educación Superior.
Contenido Relevante	Estructura del proceso de nivelación académica y sus fases.

Interpretación o Análisis	El reglamento permite comprender las normativas del proceso de nivelación y admisión que deben seguir las instituciones de educación superior y las directrices que orientan su aplicación.
----------------------------------	---

Tabla 3. Análisis documental: "Reglamento del Sistema Nacional de Nivelación y Admisión-2024"

• **Conclusión del Análisis Documental**

El Acuerdo Nro. SENESCYT-SENESCYT-2024-0055-AC establece el Reglamento del Sistema Nacional de Nivelación y Admisión (SNNA) instrumento jurídico que regula el acceso a la educación superior pública en el Ecuador articulando los procesos entre la SENESCYT y las Instituciones de Educación Superior (IES) públicas bajo los principios de mérito, equidad e igualdad de oportunidades.

El proceso se organiza en nueve etapas secuenciales de obligatorio cumplimiento desde el Registro Nacional hasta la Matriculación, e incorporando medidas de acción afirmativa que reservan entre el 5% y 10% de los cupos por carrera para grupos históricamente excluidos, lo que amplía la diversidad de perfiles de aspirantes que interactúan con las plataformas digitales institucionales durante el proceso.

La implementación del SNNA se apoya en una infraestructura tecnológica compuesta por múltiples plataformas informáticas que convierten el acceso digital en un requisito más para participar en cada etapa del proceso, se convierte en una dependencia tecnológica que genera una alta demanda simultánea de soporte técnico por parte de aspirantes que enfrentan problemas de acceso, inconsistencias en el estado de sus cupos o inconvenientes tecnológicos durante el proceso de nivelación validando así la pertinencia de una solución automatizada de soporte técnico para el contexto institucional universitario.

3.1.3. Catálogo de Servicios

Con base a la información recopilada mediante la entrevista y análisis documental, se elaboró un catálogo de servicios institucional que organiza de manera estructurada los incidentes tecnológicos identificados en cuatro categorías principales (Figura 3): Gestión de Accesos, Gestión de Comunicación, Gestión de Soporte y Gestión de Sistemas Académicos. Cada categoría agrupa un conjunto de servicios específicos que representan los tipos de incidentes más recurrentes durante el proceso de nivelación estudiantil, el catálogo constituye el insumo base para la representación formal del conocimiento del dominio institucional.

El presente catálogo de servicios ha sido creado conforme a las buenas practicas definidas según ITIL v4 y COBIT 5, que tiene como objetivo proporcionar información clara y accesible de los servicios tecnológicos disponibles del proceso de Nivelación y Admisión [105]. Según ITIL v4 la gestión del catálogo de servicios debe proporcionar una información consistente y segmentada por audiencia para una mejor comprensión, y COBIT 5 establece que el catálogo de servicios debería documentar las definiciones de cada servicio por niveles aprobadas por la institución, para esto se presenta un formato estructurado que contiene los elementos claves recomendados por ambos marcos (ver Tabla 4). Es así, como se concluye la primera fase del presente proyecto y cumpliendo el primer objetivo específico.

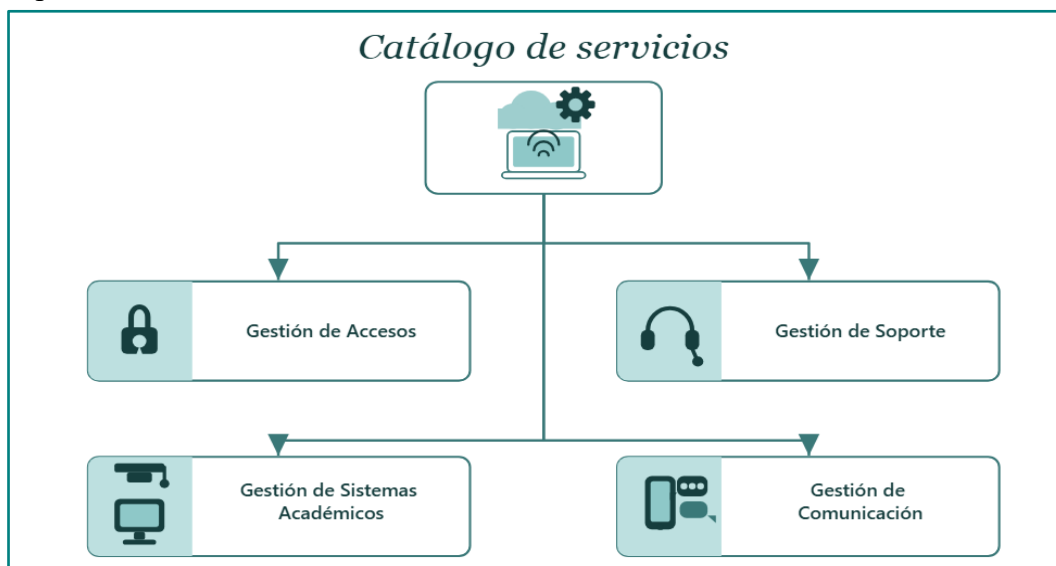


Figura 3. Estructura principal del catálogo de servicios

Catálogo De Servicios Del Proceso De Admisión Y Nivelación							
Categoría	Subcategoría (Servicio)	Descripción del servicio	Público objetivo	SLA (Nivel de servicio)	Condiciones de acceso	Estado	Responsable
Gestión de Accesos	Aula Virtual (Moodle/Eva)	Acceso a la plataforma educativa antes de iniciar sesión.	Docentes y estudiantes	Servicio disponible 24/7. Atención por correo institucional según disponibilidad.	Acceso vía navegador o app móvil.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Accesos	Sistema Institucional	Acceso al sistema institucional antes de iniciar sesión.	Docentes y estudiantes	Servicio disponible 24/7. Atención por correo institucional según disponibilidad.	Acceso vía navegador institucional.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Accesos	Correo Institucional	Acceso al correo institucional antes de iniciar sesión.	Docentes y estudiantes	Servicio disponible 24/7. Atención por correo institucional.	Acceso vía web o app mail institucional.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Accesos	Plataforma Zoom	Acceso a Zoom del curso antes de iniciar sesión(enlace incorrecto, error de autenticación).	Docentes y estudiantes	Atención según horario de clases. Soporte por correo institucional.	Enlace Zoom proporcionado.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Soporte	Asistencia Técnica	Atención a problemas en dispositivos institucionales (solo docentes y personal administrativo) e inconvenientes con el funcionamiento de las herramientas tecnológicas	Personal administrativo, docentes, estudiantes.	Atención presencial en horario laboral.	Solicitud directa en oficina técnica o por correo institucional.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.

		institucionales del proceso de Nivelación y Admisión.					
Gestión de Soporte	Sistema de Gestión Académica (SGA)	Soporte a inconvenientes con uso de las funciones que contiene el sistema institucional SGA.	Docentes y estudiantes	Servicio disponible 24/7. Atención por correo institucional.	Acceso vía navegador institucional.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Soporte	Email Institucional	Soporte a inconvenientes con uso del email institucional	Personal administrativo, docentes, estudiantes	Servicio disponible 24/7. Atención por correo institucional.	Acceso vía web o app mail institucional.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Sistemas Académicos	Moodle	Uso de las funcionalidades que contiene el aula virtual.	docentes y estudiantes	Servicio disponible 24/7. Atención por correo institucional.	Acceso vía navegador o app móvil.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Sistemas Académicos	Calificaciones	Visualización de notas en SGA o Moodle.	Personal administrativo, docentes, estudiantes	Servicio disponible 24/7. Atención por correo institucional.	Acceso vía navegador o app móvil.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Sistemas Académicos	Sílabos	Visualización o edición de sílabos en Moodle o SGA.	Docentes y estudiantes	Servicio disponible 24/7. Atención por correo institucional.	Acceso vía navegador o app móvil.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.
Gestión de Sistemas Académicos	Matrícula	Procesos de matriculación o visualización de matrícula en el sistema institucional.	Personal administrativo, docentes, estudiantes	Atención por correo institucional en fechas definidas.	Acceso vía navegador institucional.	Activo	Departamento de Nivelación y Admisión o Área de TI institucional.

Gestión de Comunicación	Canales Digitales	Difusión de comunicados del proceso de nivelación y admisión de la universidad por sitios oficiales en Facebook, Telegram, y páginas web.	Personal administrativo, docentes, estudiantes	Publicación continua. Atención por mensaje directo en el sitio o por correo institucional.	Acceso libre a redes sociales y páginas web.	Activo	Departamento de Nivelación y Admisión.
Gestión de Comunicación	Solicitud de Documentos	Acceso a formatos de solicitud (validación de documentos por la universidad.) en la página web del curso de Nivelación.	Personal administrativo, docentes, estudiantes	Servicio disponible 24/7. Atención por correo institucional.	Acceso vía página web del curso.	Activo	Departamento de Nivelación y Admisión.

Tabla 4. Catálogo de servicios

Nota aclaratoria:

- Los niveles de servicios están definidos según acuerdos internos y varían según el tipo de usuario.
- Las condiciones de acceso a los servicios pueden incluir restricciones de horario o de perfil según los reglamentos internos de la institución.
- El catálogo de servicio puede ser actualizado periódicamente de acuerdo a los cambios que se requieran con relación a las necesidades de la institución.

Versión: noviembre 2025.

3.2. FASE DE CONCEPTUALIZACIÓN

3.2.1. Modelo Conceptual: Construcción de la Ontología

A partir del catálogo de servicios que permitió identificar los conceptos esenciales del dominio estableciendo las definiciones y relaciones entre categorías y subcategorías, dio paso al cumplimiento del segundo objetivo específico para la construcción del modelo conceptual a través de una estructura ontológica (Figura 4) que representa formalmente los conceptos claves del dominio desarrollada en la herramienta Protégé.

La ontología codifica 19 clases que corresponden a: 1 clase raíz “catálogo”, las 4 categorías y 12 servicios, establecidas mediante etiquetas que establecen sus 19 nombres (*rdfs:label*), 6562 términos alternativos (*skos:altLabel*) distribuidos y relacionados a las 18 clases (categorías y servicios) los mismos que cubren el vocabulario heterogéneo de los usuarios, además de la definición de conceptos de las 18 clases (*skos:definition*), alcanzando un total de 6599 etiquetas.

La relación jerárquica entre las clases se establece con el uso de “*rdfs:subClassOf*” el enfoque estándar en OWL que indica cuando un servicio es subclase de una categoría, además el uso de la propiedad “*perteneceACategoria*” como regla de restricción “*owl:Restriction*” que declara formalmente como un servicio pertenece a una sola categoría permitiendo a la ontología validar, corregir o enriquecer automáticamente la clasificación producida por el modelo Bert.

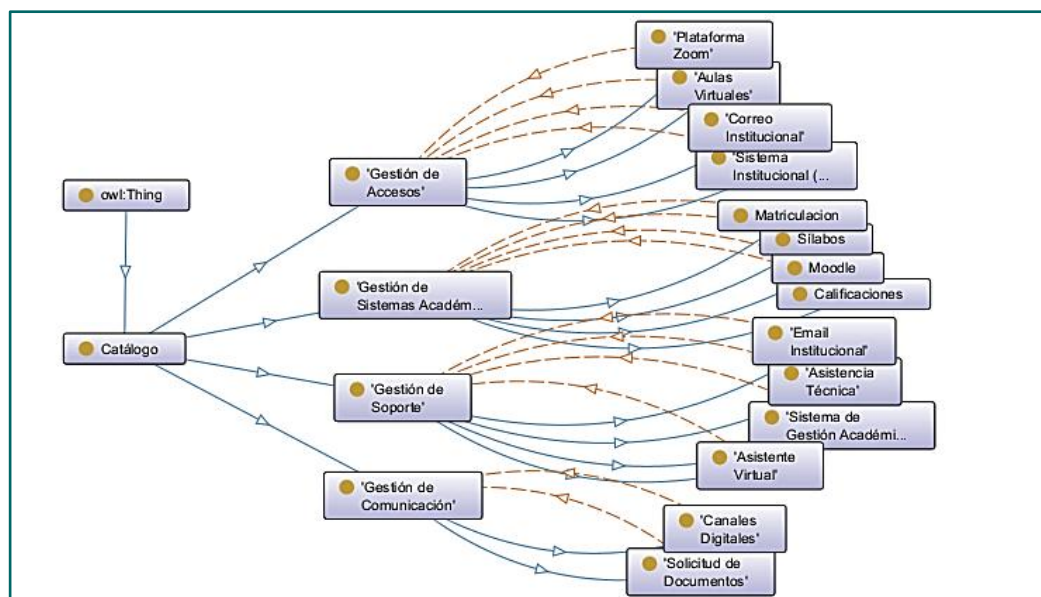


Figura 4. Diagrama ontológico del catálogo de servicios visualizado en PROTÉGÉ.

3.2.2. Datasets

- *Dataset de Incidentes*

El conjunto de datos de entrenamiento utilizado en este proyecto fue construido de forma incremental a lo largo del proceso de investigación pasando por 6 versiones que reflejan el refinamiento progresivo. La Figura 5 muestra como cada versión incorporó correcciones con nuevos incidentes para mejorar la representatividad del lenguaje utilizado por los estudiantes del proceso de nivelación y admisión.

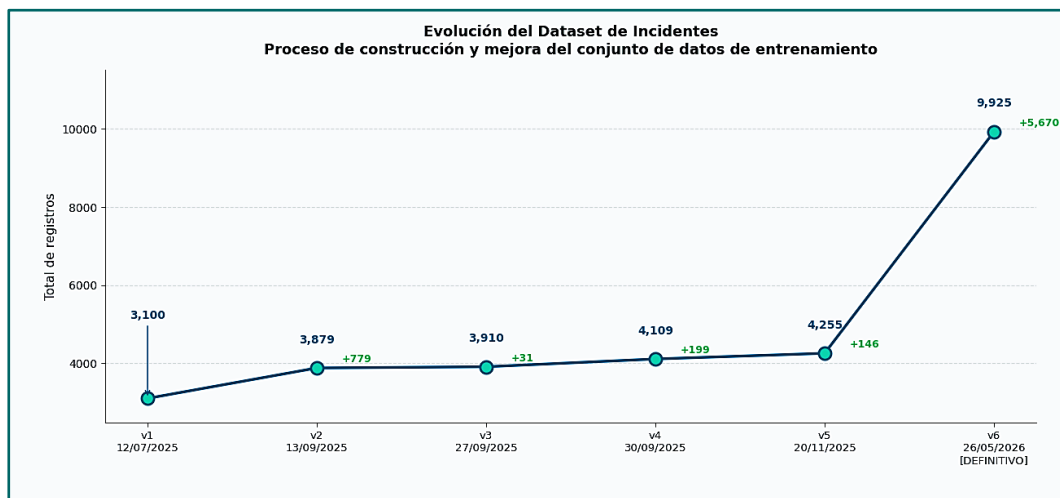


Figura 5. Registros de la evolución del dataset de incidentes a lo largo de sus versiones

En la Figura 6 se presenta la trayectoria de crecimiento del dataset a lo largo de sus versiones, evidenciando un gran incremento en el número de registros desde la versión inicial hasta el dataset definitivo del 26 de mayo de 2026.

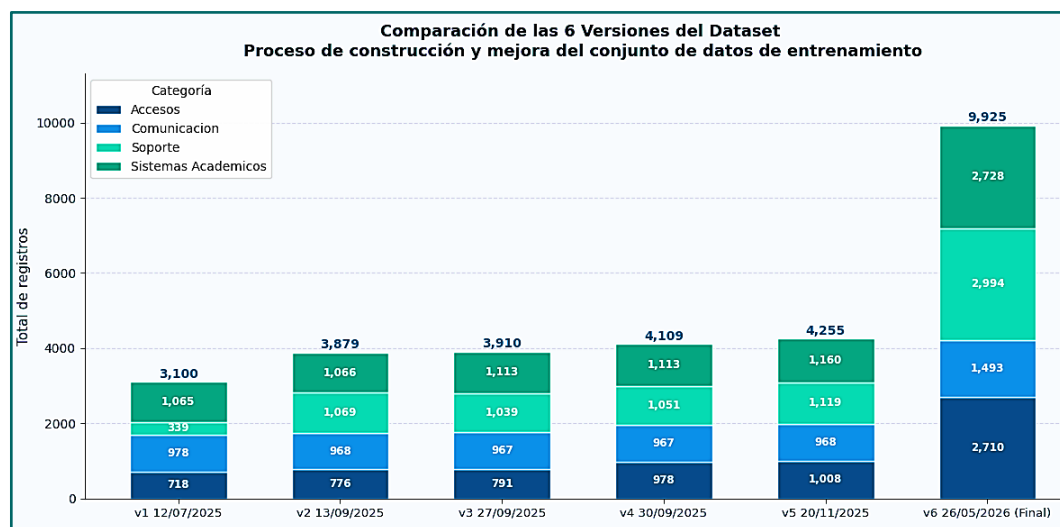


Figura 6. Comparación de las 6 versiones del dataset de incidentes por categorías

Se muestra la distribución de registros por categoría en cada versión, permitiendo observar que las cuatro categorías del dominio Gestión de Accesos, Gestión de Soporte, Gestión de Sistemas Académicos y Gestión de Comunicación.

La evolución del dataset respondió a la necesidad de ampliar la cobertura semántica de las consultas que los usuarios pueden formular en lenguaje natural incluyendo expresiones coloquiales, variaciones ortográficas y terminos informales no contemplados en las versiones iniciales. A medida que se identificaron casos de clasificación incorrecta se incorporaron nuevos ejemplos representativos de esos patrones lingüísticos no representados en las últimas versiones logrando fortalecer así la capacidad de los modelos para clasificar correctamente.

- ***Dataset de Soluciones***

El dataset de soluciones fue construido de forma incremental a lo largo del proyecto pasando por versiones que reflejan el proceso de mejora en su contenido, cada solución registrada incluye un título descriptivo del problema, un texto explicativo de la causa y los pasos detallados para la solución, organizados en relación al servicio respectivo.

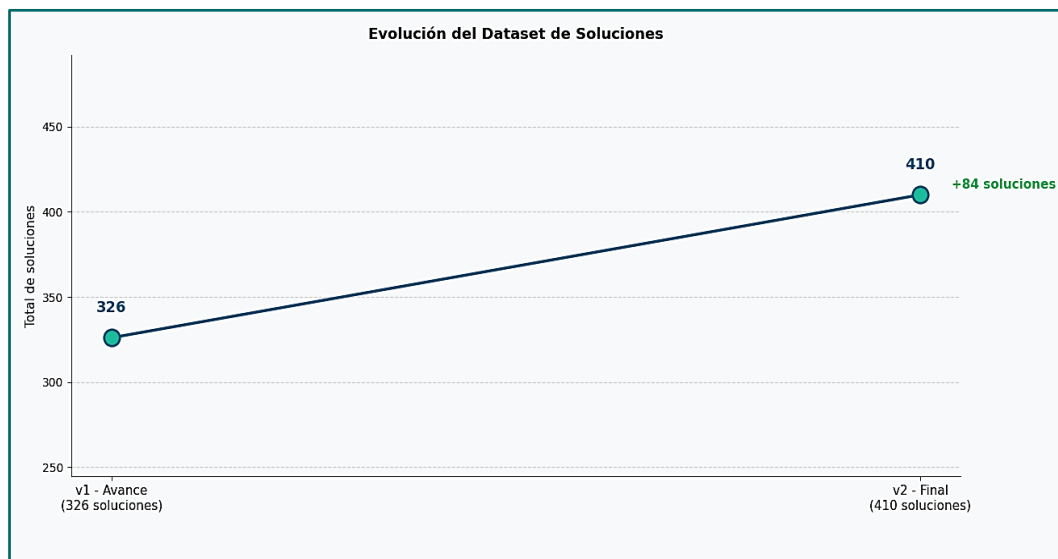


Figura 7. Registro de la evolución del dataset de soluciones

Observamos en la Figura 7 la primera versión del dataset de conocimiento contaba con 326 soluciones distribuidas entre los 14 servicios pertenecientes a las 4 categorías del catálogo del, y la versión final consolidó 410 soluciones

incorporando nuevos casos identificados durante las pruebas. Se presenta la trayectoria de crecimiento del dataset de soluciones entre sus dos versiones, evidenciando un incremento de 84 registros desde la versión de avance hasta el dataset definitivo, lo que representa un crecimiento del 25,8% en la base de conocimiento disponible para el sistema.

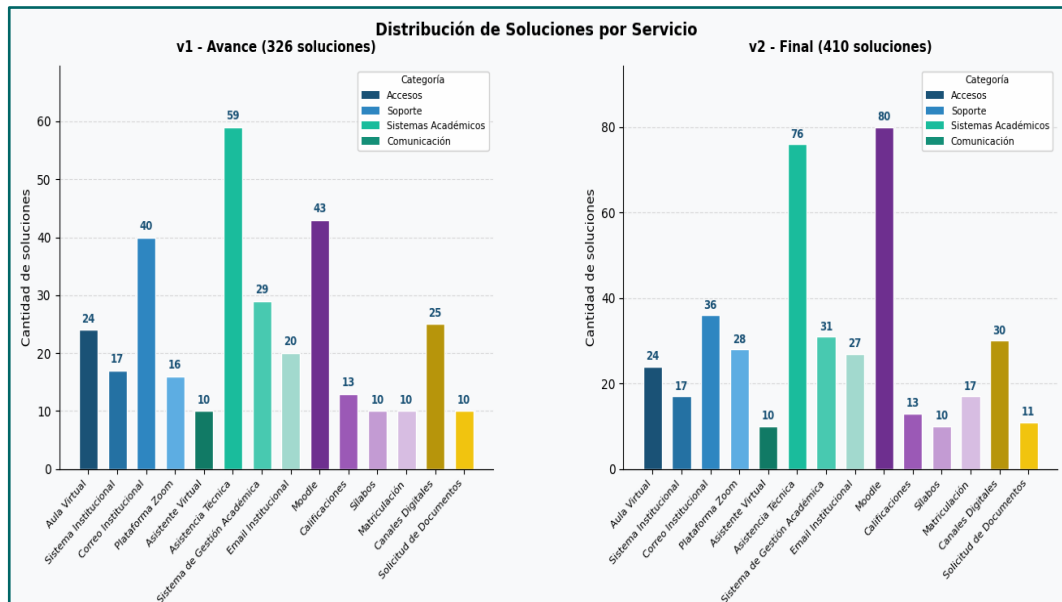


Figura 8. Comparación de la evolución del dataset de soluciones por servicios

Figura 8 muestra una comparación de las 2 versiones del dataset de soluciones, la distribución por servicio en cada versión mostrando su incremento en las cuatro categorías del dominio: Gestión de Accesos, Gestión de Soporte, Gestión de Sistemas Académicos y Gestión de Comunicación. Se observa que la categoría con mayor crecimiento entre las versiones fue la Gestión de Sistemas Académicos que pasó de 76 a 120 soluciones, seguida de la categoría de Gestión de Soporte con un incremento de 118 a 144 registros.

La ampliación del dataset de soluciones respondió a la necesidad de cubrir las soluciones para escenarios de incidentes que no estaban representados en la versión inicial y que fueron identificados a partir de las pruebas del sistema y de la revisión del dataset de soluciones. A medida que se detectaron tipos de problemas sin solución registrada se incorporaron nuevas entradas que fortalecen la capacidad del sistema para ofrecer respuestas pertinentes ante una mayor variedad de consultas de los usuarios.

3.3. FASE DE FORMALIZACIÓN Y DESARROLLO

3.3.1. Arquitectura del Sistema

El proyecto desarrollado utiliza una arquitectura en capas que gestiona los incidentes y el conocimiento del dominio mediante un asistente virtual para la interacción entre usuario y el sistema (Figura 9). Cada consulta que realiza el usuario es enviada a la API desarrollada en Python con FastAPI que conecta todo el sistema, envía el incidente hacia el módulo de “Bert” y luego hacia el módulo de la ontología para clasificarlo correctamente, obteniendo la clasificación final se envía el incidente al módulo de embedding para convertir el texto del usuario en un vector que permita calcular similitud contra el conocimiento almacenado, devolviendo el top 3 de soluciones más opcionales, concluyendo así este ciclo.

Como segunda opción para casos donde el usuario no reciba una solución para su problema, el sistema permite escalarlo a un ticket de soporte técnico que será atendido por medio de la interfaz de zammad y que a la vez se le incluyó en la interfaz un modal para gestionar el conocimiento activándose únicamente cuando el incidente es cerrado extrayendo el conocimiento obtenido de esa solución manteniendo la base de conocimiento siempre este actualizada, así como la estructura ontológica.

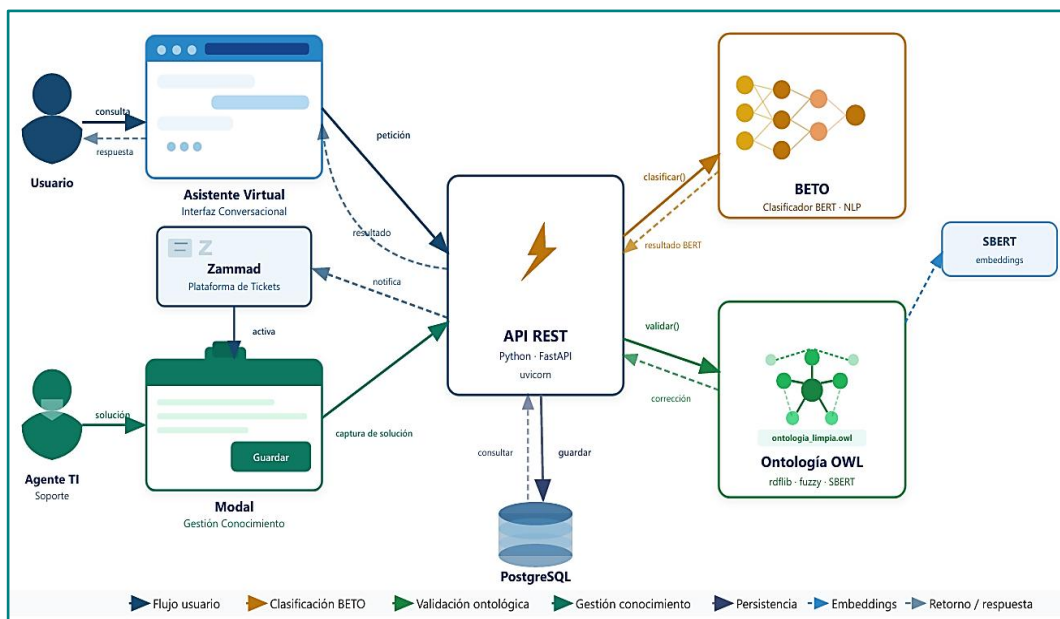


Figura 9. Arquitectura del Sistema

3.3.2. Diseño y Especificaciones del Esquema de Base de Datos Relacional

- Esquema de la Base de Datos

El esquema creado (Figura 10) para el proyecto fue implementado dentro de la base de datos de Zammad que en su implementación y al ser un software libre permite el acceso a su base de datos en PostgreSQL, lo que resulta beneficioso para lograr conectar efectivamente la información desde el asistente virtual con la información generada a través del gestor de tickets.

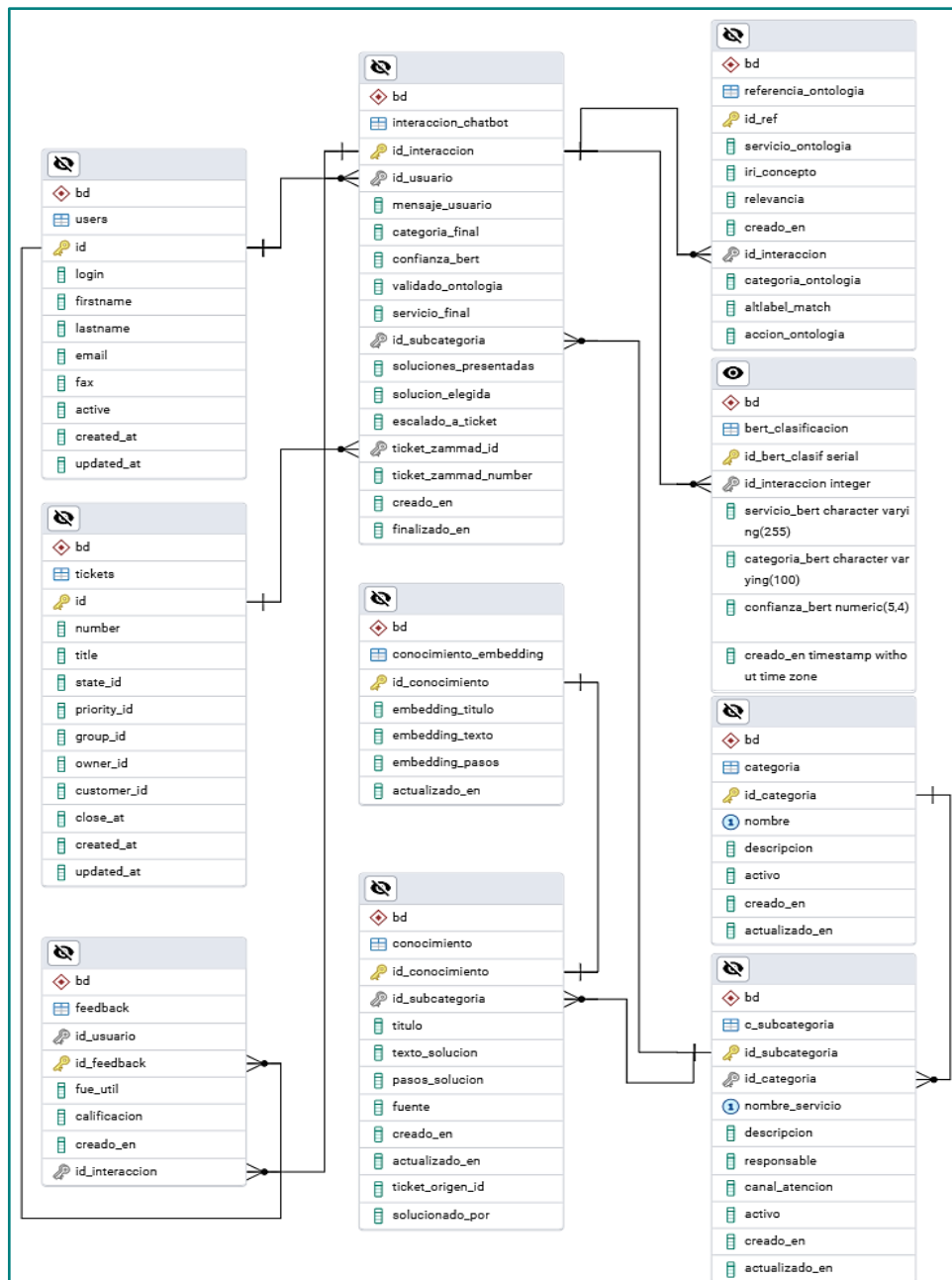


Figura 10. Esquema de base de datos

Este diseño contempla 8 tablas propias que cubren la información generada de cada proceso clave del sistema, organiza los servicios del dominio, tabla del conocimiento en su forma textual y su representación vectorial, las tablas de registro que capturan cada etapa del flujo de atención al usuario como las clasificaciones de bert y ontología, así como la retroalimentación del conocimiento. A la vez permite una relación a 2 tablas “users” y “tickets” pertenecientes a la estructura de las tablas del gestor de tickets Zammad mediante llaves foráneas con comportamiento set null permitiendo que, si en algún caso el usuario o ticket se elimine de Zammad, estos campos queden en null y no ocurra ningún error.

- **Especificaciones del Catálogo de Datos**

El catálogo de datos documenta la estructura formal del esquema de base de datos relacional, permitiendo especificar cada campo, los tipos de datos que guarda, las restricciones y el propósito de cada tabla, con el fin de asegurar su comprensión, el mantenimiento y la trazabilidad del esquema de datos implementado en este proyecto. A continuación, se encuentran las especificaciones de cada tabla:

Nombre de la tabla	categoría			
Descripción	Almacena las categorías que agrupan los servicios institucionales.			
Campo	Tipo	Longitud	Nulo	Detalle
id_categoria	Integer	-	No	Clave primaria
nombre	Varchar	100	No	Nombre único de la categoría
descripción	Text	-	Si	Descripción general de la categoría
activo	Boolean	-	Si	Indica si la categoría está habilitada
creado_en	Timestamp	-	Si	Fecha y hora de creación del registro
actualizado-en	Timestamp	-	Si	Fecha y hora de última actualización

Tabla 5. Especificaciones de la tabla categorías

Nombre de la tabla	c_subcategoría			
Descripción	Almacena los servicios pertenecientes a cada categoría incluyendo información de contacto y atención.			
Campo	Tipo	Longitud	Nulo	Detalle
id_subcategoría	Integer	-	No	Clave primaria
id_categoria	Integer	-	No	Fk – categoría a la que pertenece el servicio
nombre_servicio	Varchar	100	No	Nombre del servicio
descripción	Text	-	No	Descripción detallada del servicio
responsable	Varchar	150	No	Persona o área responsable del servicio
canal_atencion	Text	-	No	Medio por el que se da atención al servicio
activo	Boolean	-	Si	Indica si el servicio está habilitado
creado_en	Timestamp	-	Si	Fecha y hora de creación del registro
actualizado-en	Timestamp	-	Si	Fecha y hora de última actualización

Tabla 6. Especificaciones de la tabla c_subcategoría

Nombre de la tabla	conocimiento			
Descripción	Almacena la base de conocimiento creada en el “dataset de soluciones”, además permite guardar las soluciones nuevas provenientes de los tickets solucionados en Zammad.			
Campo	Tipo	Longitud	Nulo	Detalle
id_conocimiento	Integer	-	No	Clave primaria
id_subcategoría	Integer	-	Si	Servicio al que tiene corresponde la solución
titulo	Varchar	500	No	Título descriptivo de la solución.
texto_solucion	Text	-	No	Descripción de solución.
pasos_solucion	Text	-	Si	Pasos detallados para resolver el problema.
fuentes	Varchar	100	No	Medio por el que se guardó esa solución.

ticket_origen_id	Varchar	50	Si	Id del ticket por el que se obtuvo esa solución.
solucionado_por	Varchar	150		Agente que soluciono el ticket.
creado_en	Timestamp	-	Si	Fecha de registro.
actualizado_en	Timestamp	-	Si	Última modificación

Tabla 7. Especificaciones de la tabla conocimiento

Nombre de la tabla	conocimiento_embedding			
Descripción	Almacena las representaciones vectoriales (embedding) de cada conocimiento guardado.			
Campo	Tipo	Longitud	Nulo	Detalle
id_conocimiento	Integer	-	No	Clave primaria (PK) y relación a la solución(FK) ON DELETE CASCADE
embedding_titulo	Vector	384	Si	Vector semántico del titulo
embedding_texto	Vector	384	Si	Vector semántico de la descripción.
embedding_pasos	Vector	384	Si	Vector semántico de los pasos.
actualizado_en	Timestamp	-	Si	Fecha de generación del vector.

Tabla 8. Especificaciones de la tabla conocimiento_embedding

Nombre de la tabla	interaccion_chatbot			
Descripción	Almacena la base de conocimiento creada en el “dataset de soluciones”, además permite guardar las soluciones nuevas provenientes de los tickets solucionados en Zammad.			
Campo	Tipo	Longitud	Nulo	Detalle
id_conocimiento	Integer	-	No	Clave primaria
id_subcategoria	Integer	-	Si	Servicio al que tiene corresponde la solución
titulo	Varchar	500	No	Titulo descriptivo de la solución.
texto_solucion	Text	-	No	Descripción de solución.
pasos_solucion	Text	-	Si	Pasos detallados para resolver el problema.

fuelle	Varchar	100	No	Medio por el que se guardó esa solución.
ticket_origen_id	Varchar	50	Si	Id del ticket por el que se obtuvo esa solución.
solucionado_por	Varchar	150		Agente que soluciono el ticket.
creado_en	Timestamp	-	Si	Fecha de registro de la solución.
actualizado_en	Timestamp	-	Si	Última modificación

Tabla 9. Especificaciones de la tabla interaccion_chatbot

Nombre de la tabla	bert_clasificacion			
Descripción	Almacena las predicciones de Bert antes de la validación ontológica.			
Campo	Tipo	Longitud	Nulo	Detalle
id_bert_clasif	Integer	-	No	Clave primaria
id_interaccion	Integer	-	Si	Interacción a la que pertenece.
servicio_bert	Varchar	100	Si	Servicio predicho por BERT
categoria_bert	Varchar	100	Si	Categoría predicha por BERT
confianza_bert	Numeric	5,4	Si	Score de confianza de BERT
actualizado_en	Timestamp	-	Si	Fecha de generación del vector.

Tabla 10. Especificaciones de la tabla bert_clasificacion

Nombre de la tabla	referencia_ontologica			
Descripción	Almacena los conceptos owl detectados e interacción de la ontología.			
Campo	Tipo	Longitud	Nulo	Detalle
id_ref	Integer	-	No	Clave primaria
id_interaccion	Integer	-	Si	Interacción a la que pertenece.
servicio_ontologia	Varchar	100	No	Servicio detectado (rdfs:altlabel)
categoria_ontologia	Varchar	100	Si	Categoría padre del servicio detectado
altlabel_match	Varchar	400	Si	Termino exacto que coincidió (skos:altlabel)

iri_concepto	Varchar	400	Si	Iri único del servicio detectado.
relevancia	Numeric	5,4	Si	Porcentaje de precisión del match de la ontología
accion_ontologia	Varchar	20	Si	Acción de la ontología: valida/corriges/complementa/sin_resultad
actualizado_en	Timestamp	-	Si	Fecha de registro.

Tabla 11. Especificaciones de la tabla referencia_ontologica

Nombre de la tabla	feedback			
Descripción	Almacena las calificaciones que da usuario.			
Campo	Tipo	Longitud	Nulo	Detalle
id_feedback	Integer	-	No	Clave primaria
id_interaccion	Integer	-	Si	Interacción a la que pertenece.
id_usuario	Integer	-	Si	Usuario que emitió la calificación.
fue_util	Boolean	-	No	Indica si la solución resolvió el problema.
calificacion	Integer	-	Si	Puntuación del 1 al 5
creado_en	Timestamp	-	Si	Fecha de registro.

Tabla 12. Especificaciones de la tabla feedback

Nombre de la tabla	Users			
Descripción	Almacena los usuarios del sistema: agentes, clientes y administradores.			
Campo	Tipo	Longitud	Nulo	Detalle
id	Integer	-	No	Clave primaria
login	Varchar	255	No	Nombre del usuario para iniciar sesión.
firstname	Varchar	100	Si	Nombre del usuario
lastname	Varchar	100	Si	Apellido del usuario
email	Varchar	255	Si	Correo electronico
fax	Varchar	100	Si	Número de cedula
active	Boolean	-		Estado activo.

created_at	Timestamp	-		Fecha de creación.
update_at	Timestamp	-		Última modificación.

Tabla 13. Especificaciones de la tabla users (de zammad)

Nombre de la tabla	tickets			
Descripción	Almacena los usuarios del sistema: agentes, clientes y administradores.			
Campo	Tipo	Longitud	Nulo	Detalle
id	Integer	-	No	Clave primaria
number	Varchar	60	No	Número del ticket visible en zammad.
title	Varchar	250	No	Título del ticket.
state_id	Integer	-	No	Estado del ticket (abierto, cerrado.....)
priority_id	Integer	-	No	Nivel de prioridad
group_id	Integer	-	No	Grupo agentes asignados
owner_id	Integer	-	No	Agente responsable del ticket.
customer_id	Integer	-	No	Usuario que reportó el problema.
close_at	Timestamp	-	Si	Fecha de cierre del ticket
created_at	Timestamp	-	No	Fecha de apertura.
updated_at	Timestamp	-	No	Ultima modificación.

Tabla 14. Especificaciones de la tabla tickets (de zammad)

3.3.3. Dataset en Formato Tabular

Para el proceso de formalización de los datasets los incidentes descritos en el mismo sirvieron como base para definir las expresiones registradas en la ontología, cada tipo de problema documentado fue analizado para identificar las distintas formas en que un usuario podría formular una consulta las cuales se añadieron como términos alternativos mediante “*skos:altLabel*” según la clase correspondiente (visualizar Figura 11). Este proceso dio como resultado 6562 etiquetas alternativas distribuidas entre los servicios definidos en la ontología lo que amplía la capacidad del sistema para reconocer las diferentes expresiones del usuario.

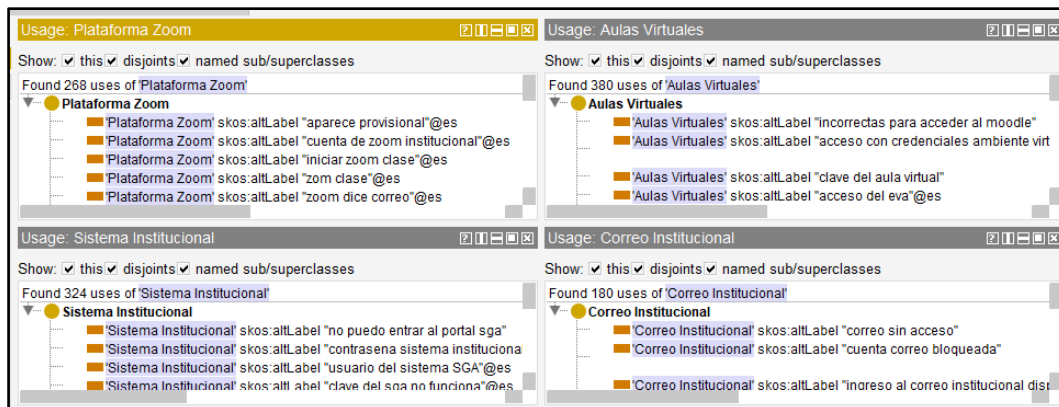


Figura 11. Vista parcial de los skos:altlabel definidos de 4 servicios seleccionados.

El dataset de soluciones fue estructurado en formato tabular a partir del archivo Dataset_soluciones.csv organizando cada solución en los campos: *titulo*, *texto_solucion* y *pasos_solucion*, en esta misma tabla en la base de datos cada conocimiento está vinculado a su respectiva subcategoría(servicio) mediante el campo *id_subcategoria*. La tabla conocimiento contiene 410 registros distribuidos entre las 14 subcategorías del catálogo institucional, con el campo fuente marcado como “dataset” para identificar su origen (Figura 12).

zammad_production/zammad@zammad										
Data Output Messages Notifications										
Showing rows: 1 to 410 Page No: 1 of 1										
	id_conocim [PK] integer	id_subcateg integer	titulo character varying (500)	texto_solucion text	pasos_solucion text	fuentes character	creado_en timestamp with	actualizado_en timestamp with	ticket_origen_id character varying	solucionado_por character varying
1	1	1	No sé cómo ingres...	El usuario no id...	1. Ingresa al port...	dataset	2026-06-10 ...	[null]	[null]	[null]
2	2	1	Error 403 o acceso...	El sistema bloq...	1. Asegúrate de i...	dataset	2026-06-10 ...	[null]	[null]	[null]
3	3	1	Usuario y contrase...	Las credencial...	1. Verifica que e...	dataset	2026-06-10 ...	[null]	[null]	[null]
4	4	1	Cuenta bloqueada ...	La cuenta se blo...	1. Espera unos ...	dataset	2026-06-10 ...	[null]	[null]	[null]
5	5	1	No puedo cambiar...	El usuario no lo...	1. Ingresa a http...	dataset	2026-06-10 ...	[null]	[null]	[null]
6	6	1	No aparecen mis ...	El estudiante no...	1. Ingresa al Aul...	dataset	2026-06-10 ...	[null]	[null]	[null]
7	7	1	El Moodle se qued...	La plataforma n...	1. Verifica tu con...	dataset	2026-06-10 ...	[null]	[null]	[null]
8	8	1	Aula Virtual caída ...	El sistema pres...	1. Verifica si otro...	dataset	2026-06-10 ...	[null]	[null]	[null]
9	9	1	Cuenta suspendid...	El sistema no re...	1. Verifica que in...	dataset	2026-06-10 ...	[null]	[null]	[null]

Figura 12. Dataset de soluciones en formato tabular (tabla conocimiento)

Durante la importación, cada registro insertado en la tabla conocimiento genera automáticamente tres representaciones vectoriales mediante el modelo Sentence-Bert para el “*titulo*”, “*texto_solucion*” y “*pasos_solucion*”. Estas representaciones vectoriales se almacenan en la tabla “*conocimiento_embedding*” (Figura 13) mismas que posteriormente permiten calcular la similitud semántica entre la

consulta de un usuario y las soluciones registradas o en la fase de retroalimentación del conocimiento.

	id_conocimiento [PK] integer	embedding_titulo vector	embedding_pasos vector	embedding_texto vector	actualizado_en timestamp with time zone
2	2	[-0.004247882,-0.035239123,-0.041159...	[0.10549511,-0.19451825,0.028998474,0.0...	[-0.12076649,-0.15053108,-0.323965...	2026-06-10 ...
3	3	[-0.15642503,0.07340492,-0.03725099,-...	[-0.077096485,0.04598653,0.004862601,-0...	[0.101833075,0.02768692,-0.052253...	2026-06-10 ...
4	4	[-0.21720007,0.056557294,-0.0839514,0...	[-0.001903754,0.033198856,-0.08105701,-0...	[-0.02915786,-0.054255042,-0.33431...	2026-06-10 ...
5	5	[-0.08328127,0.1217245,0.19617656,-0...	[-0.059786752,-0.1019369,-0.007226708,-0...	[-0.15387145,-0.22852767,0.155016...	2026-06-10 ...
6	6	[0.33993024,-0.10437185,0.44617575,-0...	[0.2909563,0.05112555,0.09719667,-0.252...	[0.3544067,-0.061285663,0.3501266...	2026-06-10 ...
7	7	[0.008024255,0.043785103,0.0806542...	[0.048917875,-0.13227913,0.10220882,-0.0...	[0.038531493,-0.08952276,-0.02613...	2026-06-10 ...
8	8	[0.18396269,-0.125872,0.07020161,-0.2...	[0.09775059,0.07495762,-0.040468015,-0.2...	[-0.39625946,-0.32343256,0.171723...	2026-06-10 ...
9	9	[-0.02226443,0.15294418,-0.25322706,-...	[-0.0018782134,0.20969346,-0.1472032,-0...	[-0.10784191,-0.056374602,0.22175...	2026-06-10 ...
10	10	[-0.027243622,-0.09715572,-0.0212351...	[-0.041256238,-0.22001694,0.24164082,0.1...	[0.008633295,-0.12962748,0.014716...	2026-06-10 ...
11	11	[0.4431957,-0.003972454,0.14260946,-0...	[0.1902108,-0.0065101893,0.044234194,-0...	[0.45273706,-0.25849664,0.1180037...	2026-06-10 ...
12	12	[-0.2411748,0.39407256,-0.0046937005,-...	[-0.04737253,-0.14948295,0.029762799,-0...	[0.006542098,0.1634102,-0.2859803...	2026-06-10 ...
13	13	[0.009557866,0.14578472,0.1557038,-0...	[-0.048707064,-0.0940915,0.21234657,-0.0...	[-0.29387876,0.08720548,-0.057725...	2026-06-10 ...
14	14	[-0.21245278,0.064576365,-0.03065124,-...	[-0.15009794,0.009721631,0.046678927,-0...	[-0.33661023,-0.3143498,-0.0533438...	2026-06-10 ...
15	15	[0.310968,-0.014216137,0.2824588,-0.0...	[0.004001607,0.023322007,0.031693716,-0...	[0.19615494,-0.13963822,0.2433570...	2026-06-10 ...

Total rows: 410 Query complete 00:00:00.216 CRLF Ln 1, Col 77

Figura 13. Representación vectorial (embedding) de cada conocimientos almacenado.

3.3.4. Reglas Lógicas para la Ontología

1. Jerarquía taxonómica (rdfs:subClassOf)

La ontología organiza el conocimiento mediante una jerarquía de subsunción de tres niveles donde el nodo raíz es Catálogo del cual se derivan las cuatro categorías (Gestión de Accesos, Gestión de Comunicación, Gestión de Sistemas Académicos y Gestión de Soporte), cada una de estas agrupa a los 14 servicios tecnológicos correspondientes. Esta relación SubClassOf garantiza que toda clase más específica hereda las propiedades de la clase más general permitiendo el razonamiento por herencia (ver imagen de referencia en ANEXO 5).

2. Restricción existencial (someValuesFrom)

Cada clase de servicio contiene una restricción existencial que establece: "todo elemento de esta clase debe estar relacionado con al menos una categoría a través de la propiedad perteneceACategoría". Por ejemplo, la clase Aulas Virtuales declara perteneceACategoría some 'Gestión de Accesos', lo que obliga formalmente a que cualquier instancia del servicio esté asociada a su categoría correspondiente. Esta restricción regula la correcta asignación de servicios a categorías (ANEXO 6).

3. Propiedad objeto con dominio y rango (*owl:ObjectProperty*)

La propiedad perteneceACategoría define explícitamente los participantes válidos una relación. El dominio está definido como la unión de los 14 servicios, indicando que solo ellos pueden ser sujeto de esta propiedad. El rango está definido como la unión de las 4 categorías de gestión indicando que solo ellas pueden ser el destino de la relación, esto restringe formalmente las combinaciones válidas dentro de la ontología (referencia ver ANEXO 6).

4. Propiedades de anotación (*skos:altLabel*, *skos:definition*, *skos:scopeNote*)

Se establecen reglas de documentación mediante propiedades de anotación estándar SKOS. Cada servicio posee una definición formal en *skos:definition* (ANEXO 3), sinónimos en *skos:altLabel* (ANEXO 4) utilizados por el sistema de clasificación y aclaraciones de alcance en *skos:scopeNote* (ANEXO 7) que indican qué casos quedan fuera del servicio, regulando de esta manera el vocabulario controlado de la base de conocimiento

3.3.5. Desarrollo del Sistema Neuro-Simbólico de Clasificación

La categorización automática de incidentes pasa por 2 fases, primero haciendo uso de una inteligencia artificial diseñada para el procesamiento de lenguaje natural, específicamente el modelo “*dccuchile/bert-base-spanish-wwm-cased*” denominado “*Beto*”(Bert) entrenado por el Departamento de Ciencias de la Computación de la Universidad de Chile [62], que es el modelo **Bert** adaptado con un gran contenido de texto en español. Sobre éste modelo base se aplicó un proceso de ***fine-tuning*** que consiste en continuar su entrenamiento con el dataset de clasificación del curso de nivelación para que aprenda a identificar a qué servicio y categoría pertenece cada consulta, el resultado de este entrenamiento es el modelo BertForSequenceClassification siendo la versión especializada del modelo base.

La clase *Predictor* carga al iniciar el servidor el modelo ya entrenado localmente mediante *BertForSequenceClassification.from_pretrained (MODELO_PATH)*, junto con el BertTokenizer y el LabelEncoder manteniéndolos en memoria durante toda la sesión. El método *predecir()* recibe el texto del usuario y lo tokeniza con una longitud máxima de 128 tokens, procesa la secuencia a través del modelo y

aplica softmax sobre los logits para obtener una probabilidad por cada clase. El índice de mayor probabilidad se traduce al nombre del servicio y categoría correspondiente a través del *LabelEncoder* devolviendo la tupla (servicio, categoría, confianza).

Continuando con la segunda fase de la clasificación por la que pasa el incidente descrito por el usuario, el uso de la **Ontología** como motor semántico para validar/corregir/complementar o dejar por defecto la clasificación de Bert: Módulo permite interactuar con la ontología desarrollada, es decir carga el archivo *.owl* de la ontología + el cálculo de embedding por cada termino (Figura 14).

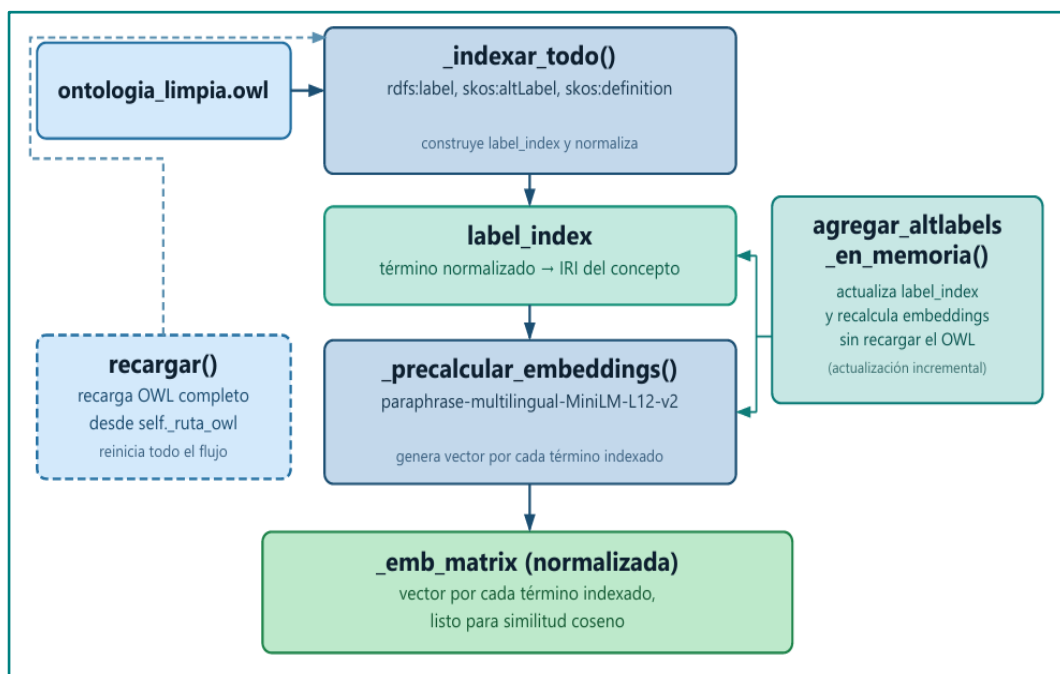


Figura 14. Inicialización de la fase ontológica (carga en memoria los terminos de la ontología)

En este diagrama se describe la fase de arranque de la ontología que permite indexar las etiquetas como nombre (rdfs:label), términos alternativos “skos:altlabel” y definiciones de las clases categoría o servicio (skos:definition) que están definidos dentro del archivo *.owl* de la ontología, a la su vez de manera simultánea se pre calculan los embeddings con S-Bert sobre los mismos términos alternativos de los servicios de la ontología, proceso que ocurre al inicializar la api para que estén listos antes de recibir el mensaje del usuario o cuando se indexa un nuevo termino desde el modal.

Por consiguiente, se recibe el texto del usuario que pasa por el flujo de validación que contiene 4 capas y la regla de protección de Bert para poder determinar la clasificación final (flujo en Figura 15):

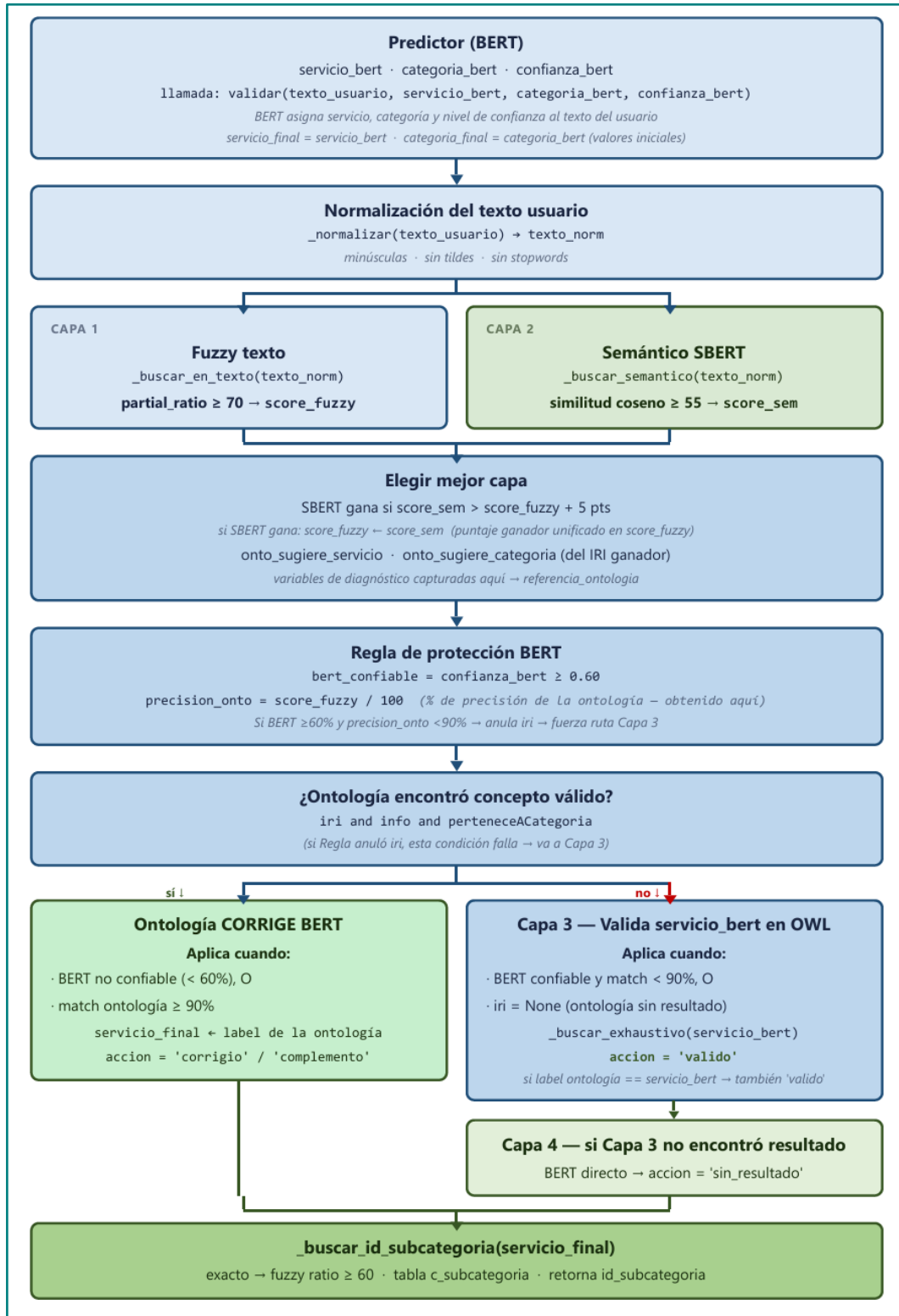


Figura 15. Fase de Validación - ontología

Este diagrama (Figura 15) muestra el flujo interno del método validar() del módulo de la ontología encargado de contrastar la predicción de Bert con la estructura ontológica, proceso que parte recibiendo los parámetros de Bert (servicio, categoría y nivel de confianza), seguido se ejecuta en paralelo 2 capas para la clasificación ontológica: capa 1 coincidencia difusa sobre el texto del usuario, capa 2 comparación semántica con S-bert, esto permite elegir el mejor resultado entre ambas capas para la clasificación de la ontología recibiendo el porcentaje de su presión y luego aplicar la regla de protección de Bert, si su confianza supera el 60% y la precisión ontológica es menor al 90%, se valida el servicio de Bert por medio de búsqueda exhaustiva en la estructura ontológica, caso contrario la ontología corrige a Bert o lo complementa si su clasificación fue incompleta

3.3.6. Desarrollo del Modal de Gestión del Conocimiento

- **Primera Sección del Modal: Gestión del Conocimiento**

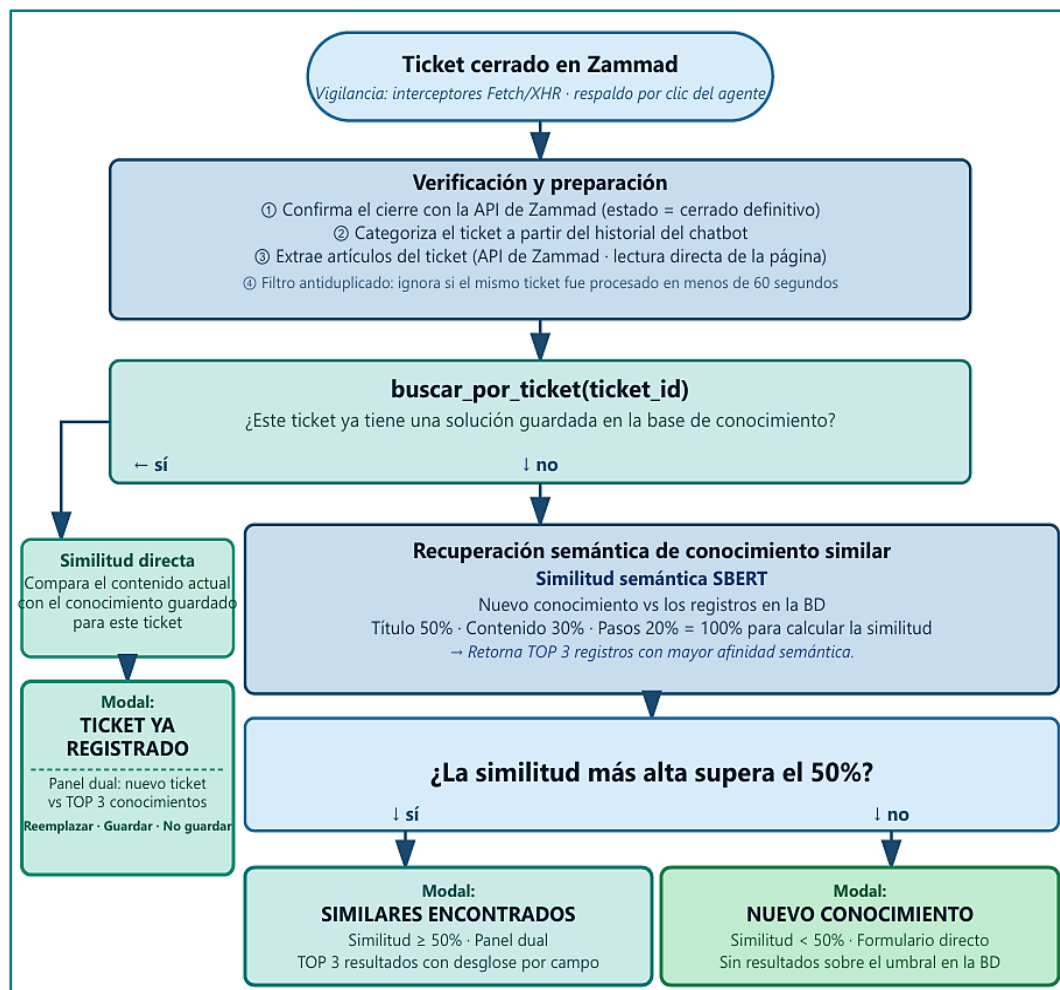


Figura 16. Primera sección del modal: Gestión del conocimiento

En el diagrama (Figura 16) se visualiza el flujo del modal desarrollado que aparece en pantalla al momento de cerrar un ticket en Zammad, permitiendo cumplir con el tercer objetivo específico (ANEXO 11), este sistema intercepta la petición al cerrar el ticket por medio de interceptores Fetch/XHR e inicia una fase que verifica y prepara la información proveniente del ticket.

Continuando con el proceso de recuperación semántica con S-bert que pondera el cálculo de similitud entre los contenidos de las soluciones de la siguiente manera: título (50%), texto de la solución (30%) y los paso (20%) utilizando un umbral mínimo de similitud del 50% con el fin de recuperar el mejor top 3 de soluciones similares al conocimiento nuevo y mostrar la versión del modal según lo encontrado como una “solución guardada”, “similares encontrados”, o “nuevo conocimiento” (soluciones no superan el umbral del 50%), facilitando la retroalimentación constante del conocimiento.

- **Segunda Sección del Modal: Enriquecimiento de la Ontología**

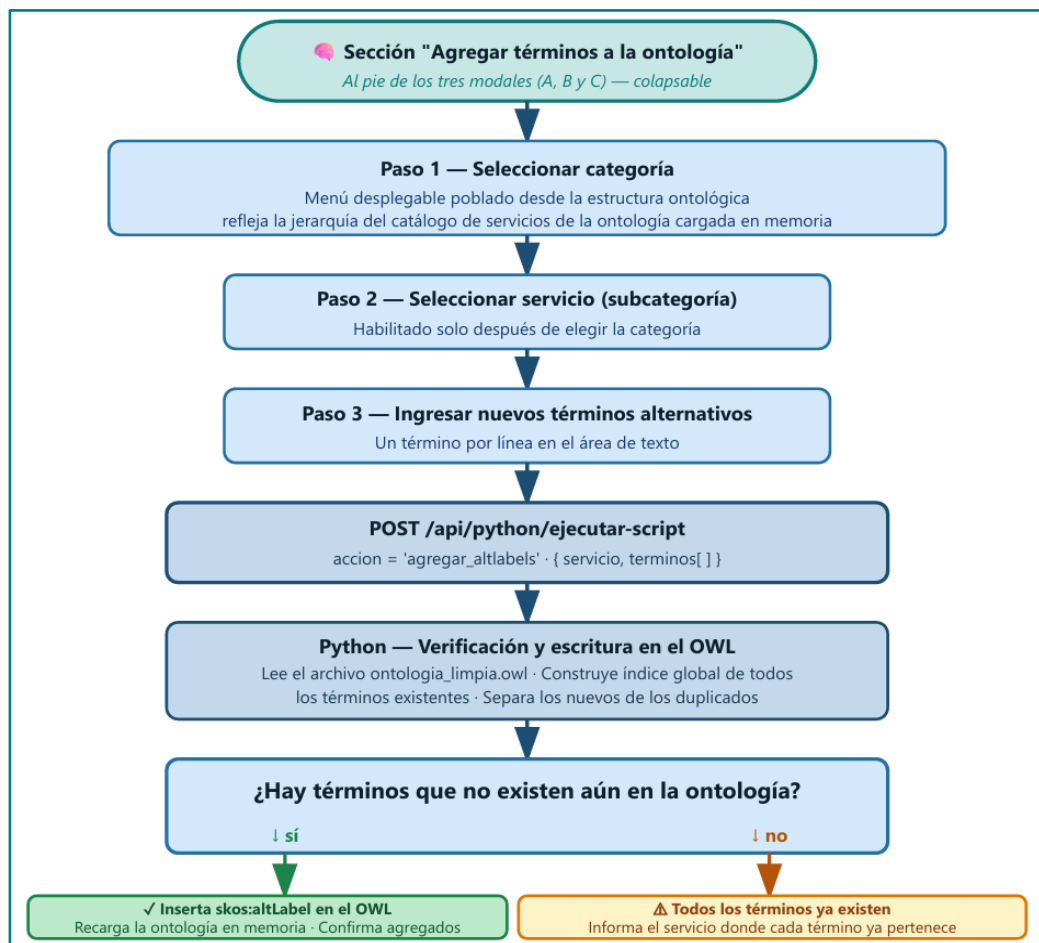


Figura 17. Segunda sección del modal: Enriquecimiento de la ontología

Para la segunda sección (Figura 17) dentro del modal, se describe el mecanismo utilizado para el enriquecimiento ontológico, se encuentra como una sección colapsable y es opcional utilizarla si se desea incorporar o no nuevos términos alterantivos en la ontología. Para esto el sistema permite realizar 3 pasos: iniciando por la selección de la categoría en un menú que muestra las 4 categorías definidas en la ontología, luego elegir el servicio relacionado a la categoría seleccionada e ingresar los nuevos términos (uno por línea) en el área de texto, además esta sección permite identificar si un término ya ha sido guardado antes para también se evitar duplicados dentro de la estructura ontológica.

3.4. FASE DE IMPLEMENTACIÓN

3.4.1. Uso del Sistema Neuro-Simbólico - Canal de Comunicación

Para el uso del sistema neuro-simbólico se implementó un asistente virtual (Figura 18) que brinda una interfaz interactiva y fácil de usar para reportar los incidentes tecnológicos presentados y obtener una solución, o que a su vez si el sistema no logra encontrar una solución efectiva, permita escalar el incidente generando a un ticket de soporte técnico que será atendido por medio de Zammad. Cumpliendo de esta manera con el cuarto objetivo específico.

The screenshot displays the UPSE ARES Bot chat interface. At the top, the header includes the UPSE logo (Universidad Estatal Península de Santa Elena) and the ARES Bot status (En línea). A session ID 'web_76t2zqi7' and a 'Reiniciar' button are also visible. The chat history shows a user input 'hola', followed by a bot response: '¡Hola! Soy el asistente de soporte de la UPSE. Para empezar, verifiquemos tu identidad.' The bot then prompts for a 10-digit ID card number, which the user provides as '0991149228'. Next, the bot asks for a 6-digit code sent to the user's email, which the user provides as '334873'. The bot confirms successful authentication: 'Autenticación exitosa. Bienvenido/a, Andrea.' Finally, the bot asks for details about the user's query or problem. At the bottom, there is a text input field with the placeholder 'Escriba hola para empezar de nuevo' and an 'Enviar' button.

Figura 18. Autenticación para acceder al servicio del asistente virtual

Para poder acceder a los servicios del asistente virtual primero pide autenticación ingresando un número de cédula e ingresando una clave temporal enviada al correo registrado, este registro de los usuarios se realiza a través de la interfaz de zammad proceso que veremos más adelante (Figura 18), una vez obtenida la autenticación procedemos a ingresar la consulta o incidente tecnológico presentado.

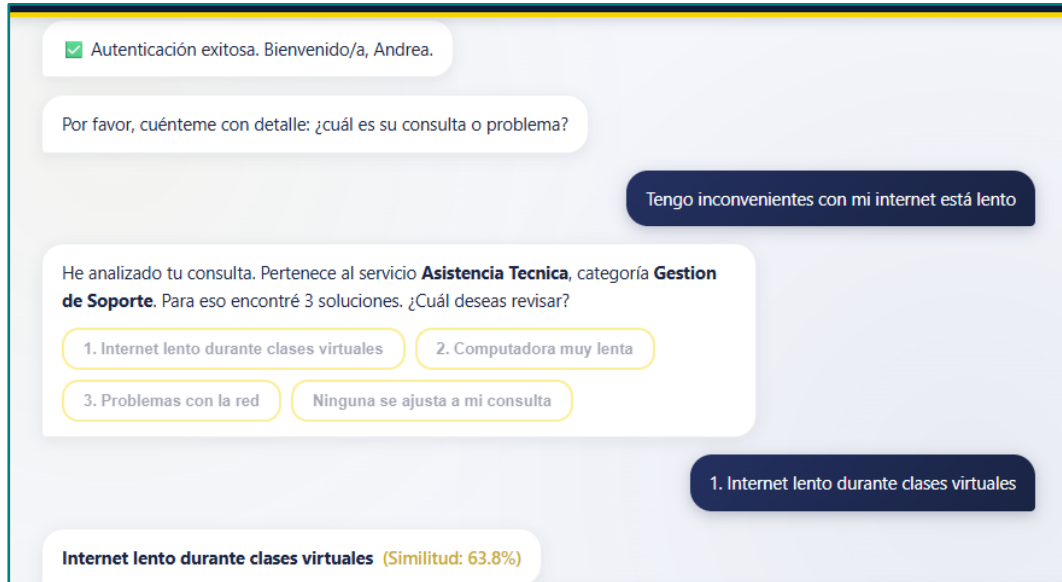


Figura 19. Ingreso de consulta al sistema para obtener una solución.

Como se puede observar en la Figura 19, se ingresa una consulta o la descripción del incidente presentado, el sistema pasa el texto por los filtros de clasificación (predicción de bert y clasificación de la ontología ANEXO 9) y luego busca el top 3 de soluciones más pertinentes utilizando el método de similitud del coseno, cada una de las 3 soluciones muestra el porcentaje de similitud que obtuvo.

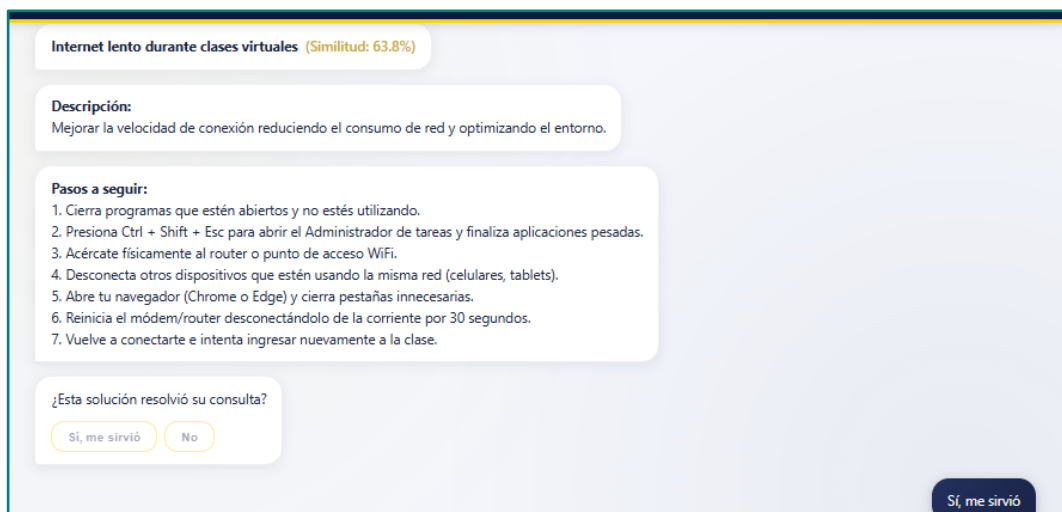


Figura 20. Sistema muestra el contenido de la solución elegida

Una vez elegida una solución del top 3 se muestra el contenido de la misma como se visualiza en la Figura 20, se observa la solución en una descripción corta seguido de los pasos para solucionar el inconveniente, debajo de la solución se observa la pregunta “¿Esta solución resolvió tu consulta?” se puede elegir “sí” o “no”, si esta solución si fue útil el sistema continúa con la fase final de esta interacción.

Me alegra haberte ayudado. Registré tu caso como resuelto en el ticket #80174 (cerrado).

Antes de finalizar, me gustaría conocer su opinión. ¿Cómo calificaría mi ayuda? Del 1 al 5 (1 = muy mala, 5 = excelente).

★ 1 ★★ 2 ★★★ 3 ★★★★ 4 ★★★★★ 5

★★★★★ 5

Gracias por su calificación, nos ayuda a mejorar.

¿Tiene otra consulta?

Sí, tengo otra consulta No, finalizar

No, finalizar

Su sesión ha finalizado. Cuando lo necesite, escriba **hola** para iniciar nuevamente. ¡Hasta pronto!

Figura 21. Calificación del servicio brindado

Después de solucionar el inconveniente o consulta, el sistema automáticamente pide al usuario una calificación del 1 al 5 siendo 1 la calificación más baja y 5 la más alta con el fin de poder analizar cada caso resuelto según su calificación y realizar los respectivos ajustes o mejoras al servicio brindado. Luego de la calificación el usuario tiene la opción de realizar otra consulta o finalizar el chat (Figura 21).

- **Incidente o Consulta no Resuelta por Medio del Asistente Virtual**

Se inicia la conversación para realizar una nueva consulta, siguiendo el mismo flujo que comienza por la autenticación e ingresar la descripción de un nuevo incidente o consulta, el sistema recorre una vez más las fases de clasificación entre Bert y la ontología para mostrar al usuario el servicio al que está relacionado el problema que presenta junto al top 3 de soluciones que podrá elegir, si ninguna de las soluciones se ajusta a las necesidades del inconveniente, así se opta por seleccionar la opción “Ninguna se ajusta a mi consulta” esto permitirá al usuario escalar su incidente a un ticket de soporte que será atendido por Zammad (ver ANEXO 10).

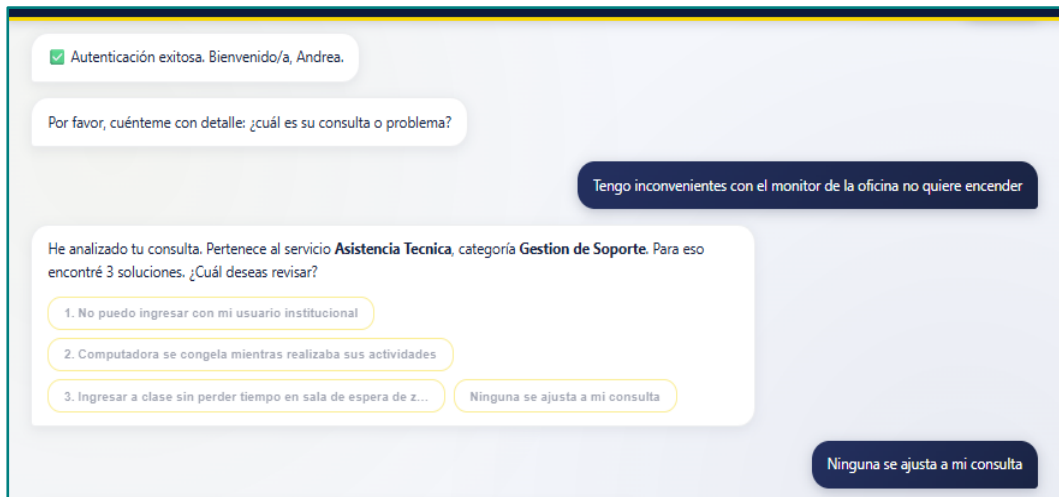


Figura 22. Nuevo incidente, no obtuvo una respuesta válida para su solución.

El incidente presentado no obtuvo una solución que se acople a las necesidades para resolverlo, ante esta situación se indica lo sucedido y espera recibir una alternativa para la solución como se aprecia en la Figura 22.



Figura 23. Incidente no solucionado, escalado a ticket de soporte

Como alternativa para brindar al usuario una solución se muestra en pantalla 4 opciones a elegir según lo que necesite: podrá ver las soluciones disponibles que se le mostraron inicialmente, puede escalar el incidente generando un ticket de soporte, podrá realizar otra consulta o salir de la sesión. Para agendar un ticket de soporte usar la opción “Necesito ayuda de un técnico”, automáticamente el sistema pide ingresar un título y luego una breve descripción del inconveniente, al final se muestra un mensaje con el número de su ticket y los detalles del mismo (Figura 23).

--Tabla 1 - interaccion_chatbot:									
a Output Messages Notifications									
Showing rows: 1 to 1 Page No: 1 of 1									
d_interaccion [PK] integer	mensaje_usuario text	categoria_final character varying (100)	servicio_final character varying (255)	confianza_bert numeric (5,4)	validado_ontologia boolean	escalado_a_ticket boolean	ticket_zammad_number character varying (60)	creado_en timestamp v	
2114	Tengo inconvenientes con el mo...	Gestion de Soporte	Asistencia Tecnica	0.9994	true	true	80176	2026-07-12	
--Tabla 2 - referencia_ontologia:									
Output Messages Notifications									
Showing rows: 1 to 1 Page No: 1 of 1									
id_ref [PK] integer	id_interaccion integer	servicio_ontologia character varying (255)	categoria_ontologia character varying (100)	accion_ontologia character varying (20)	relevancia numeric (5,2)	altlabel_match character varying (500)	url_concepto character varying (500)		
2111	2114	Asistencia Tecnica	Gestion de Soporte	valido	100.00	el monitor	http://www.semanticweb.org/o...		
--Tabla 3 - bert_clasificacion:									
a Output Messages Notifications									
Showing rows: 1 to 1 Page No: 1 of 1									
id_bert_clasif [PK] integer	id_interaccion integer	servicio_bert character varying (255)	categoria_bert character varying (100)	confianza_bert numeric (5,4)					
70	2114	Asistencia Tecnica	Gestión de Soporte	0.9994					

Figura 24. Información almacenada en la base de datos de la consulta y ticket generado

La Figura 24 muestra toda la información obtenida de la interacción almacenada en la base de datos

3.4.2. Implementación del Gestor de Tickets Zammad

Para solucionar de aquellos incidentes no resueltos por medio del asistente virtual se implementó el gestor de ticket “Zammad” (Figura 25) con el propósito de cumplir con los puntos más importantes para el presente proyecto, su interfaz interactiva permite ingresar los usuarios al sistema, recibir notificación de los tickets agendado para su solución además de conceder el uso de un método para la actualización de la base de conocimiento.

Figura 25. Interfaz de acceso al gestor de tickets Zammad

Su interfaz nos permite visualizar información importante sobre la gestión de los tickets empezando por en el panel izquierdo tenemos acceso al Panel Principal que muestra datos estadísticos sobre el tiempo en línea, porcentaje del método de atención del ticket, porcentaje total de tickets asignados al técnico, promedio total de tickets en proceso y porcentaje total de los de tickets que han sido reabiertos (Figura 26). Tenemos una visión general de los tickets abiertos y cerrados, y debajo de esa opción se muestran los tickets que han sido atendidos o están por resolver, dirigiendo la vista hacia en el panel derecho tenemos la sección de notificaciones donde queda el registro de cada interacción en Zammad (Figura 27).

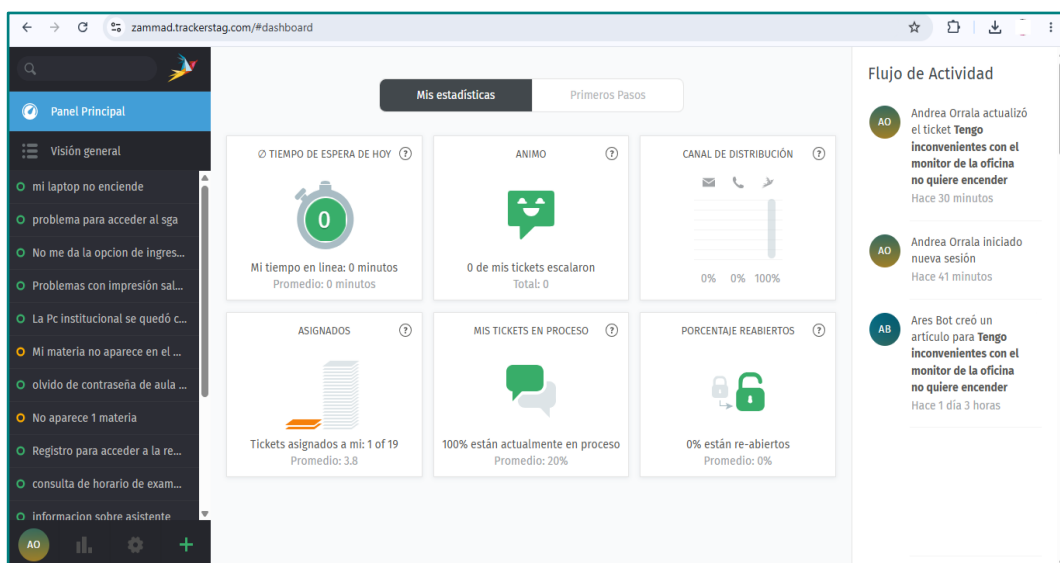


Figura 26. Panel principal de Zammad - Datos estadísticos

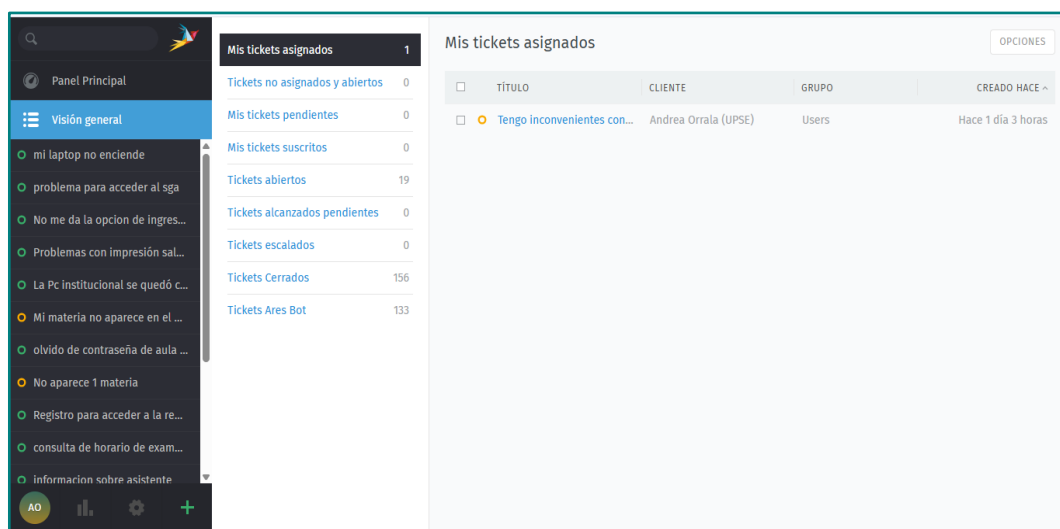
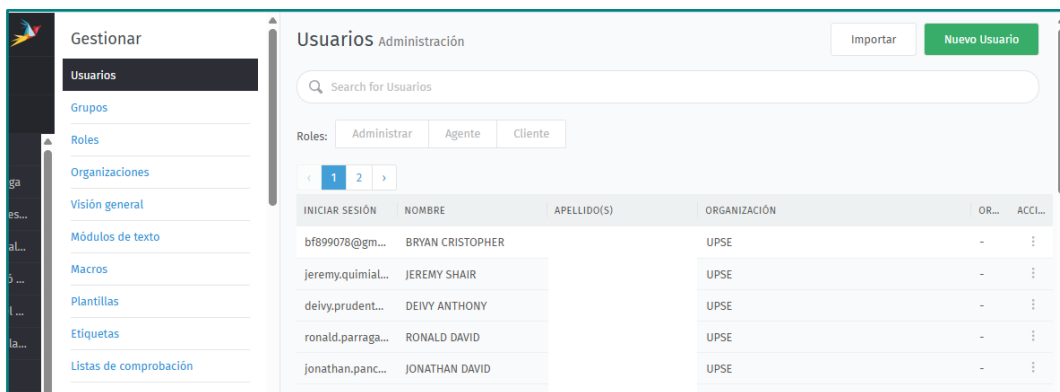


Figura 27. Visión general del estado de los tickets

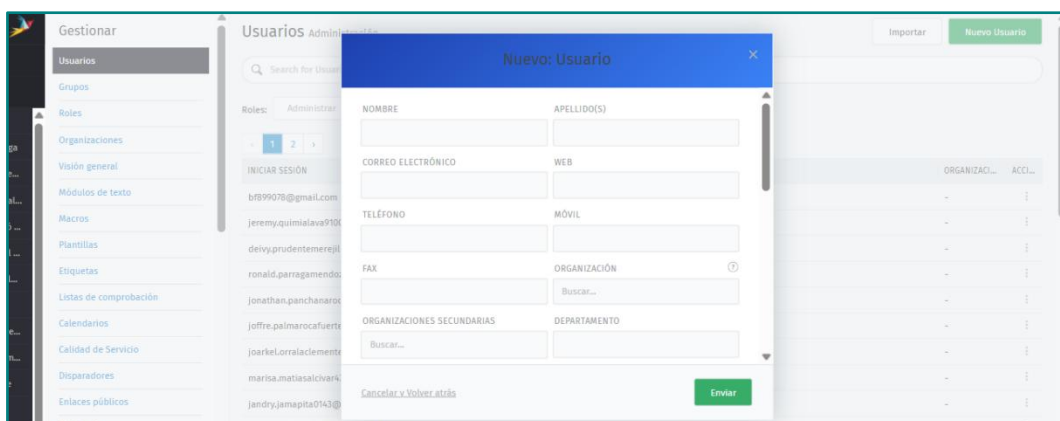
Zammad nos muestra una vista general de usuarios según su rol administrativo (Figura 28), agente y cliente, posibilita ingresar nuevos registros desde la opción “Nuevo Usuario” (Figura 29) o “Importar” (Figura 30) que permite la importación masiva con un archivo csv, cualquiera de los dos métodos solicita como datos más importantes el nombre, el número de cédula, la organización a la que pertenece, su correo electrónico, establecer una contraseña y elegir el rol del usuario.



The screenshot shows the 'Usuarios Administración' (Users Administration) page. On the left is a sidebar with navigation links: Gestionar, Usuarios (selected), Grupos, Roles, Organizaciones, Visión general, Módulos de texto, Macros, Plantillas, Etiquetas, and Listas de comprobación. The main area has a search bar 'Search for Usuarios' and a filter for 'Roles' with options 'Administrar', 'Agente', and 'Cliente'. Below this is a table of users.

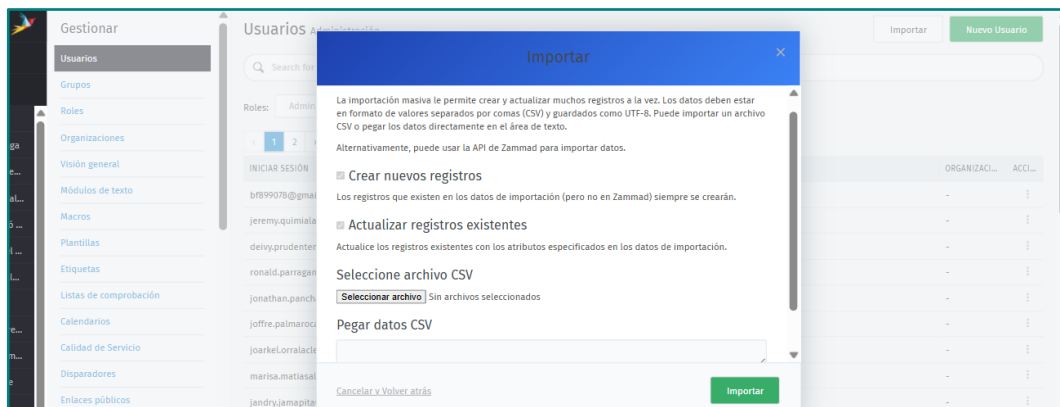
INICIAR SESIÓN	NOMBRE	APELLIDO(S)	ORGANIZACIÓN	OR...	ACCI...
bf899078@gm...	BRYAN CRISTOPHER		UPSE	-	⋮
jeremy.quimial...	JEREMY SHAIR		UPSE	-	⋮
deivy.prudent...	DEIVY ANTHONY		UPSE	-	⋮
ronald.parraga...	RONALD DAVID		UPSE	-	⋮
jonathan.panc...	JONATHAN DAVID		UPSE	-	⋮

Figura 28. Vista general de todos los usuarios ingresados



The screenshot shows the 'Nuevo: Usuario' (New User) form. It has a blue header with the title and a close button. The form contains several input fields: NOMBRE, APELLIDO(S), CORREO ELECTRÓNICO, WEB, TELÉFONO, MÓVIL, FAX, ORGANIZACIÓN (with a search icon), ORGANIZACIONES SECUNDARIAS, and DEPARTAMENTO. At the bottom, there are buttons for 'Cancelar y Volver atrás' and 'Enviar'.

Figura 29. Interfaz para registro de un nuevo usuario



The screenshot shows the 'Importar' (Import) form. It has a blue header with the title and a close button. The form contains a text area with instructions: 'La importación masiva le permite crear y actualizar muchos registros a la vez. Los datos deben estar en formato de valores separados por comas (CSV) y guardados como UTF-8. Puede importar un archivo CSV o pegar los datos directamente en el área de texto. Alternativamente, puede usar la API de Zammad para importar datos.' Below this are two radio buttons: 'Crear nuevos registros' and 'Actualizar registros existentes'. There is a section for 'Seleccione archivo CSV' with a button 'Seleccionar archivo' and a message 'Sin archivos seleccionados'. There is also a section for 'Pegar datos CSV' with a text area. At the bottom, there are buttons for 'Cancelar y Volver atrás' and 'Importar'.

Figura 30. Interfaz para registro masivo de usuarios

- **Interfaz de Solución del Ticket Generado desde el Asistente Virtual**

Una vez que el usuario haya generado el ticket desde el asistente virtual, el técnico recibe la notificación del ticket abierto y podrá abrir la interfaz del mismo, brindando varias funcionalidades como interactuar con el usuario, reasignar la atención del ticket, cambiar el estado, el nivel de prioridad, agregar etiquetas y links (Figura 31). Después de solucionar el incidente, podemos actualizar el estado del ticket a “cerrado” y continuar a la última fase del sistema, el modal como punto de actualización del conocimiento y de la estructura ontológica (Figura 32).

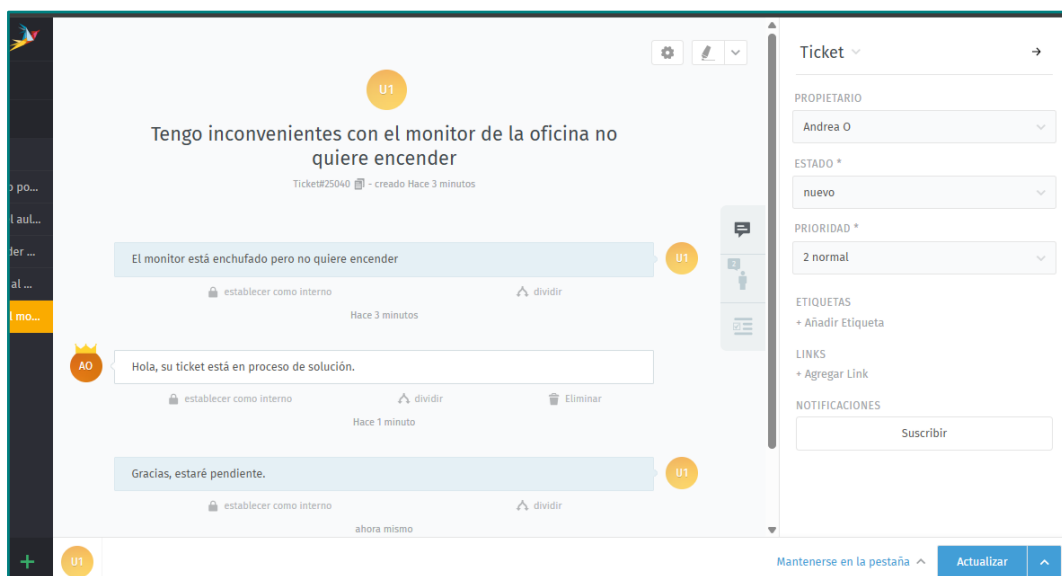


Figura 31. Interfaz del ticket abierto.

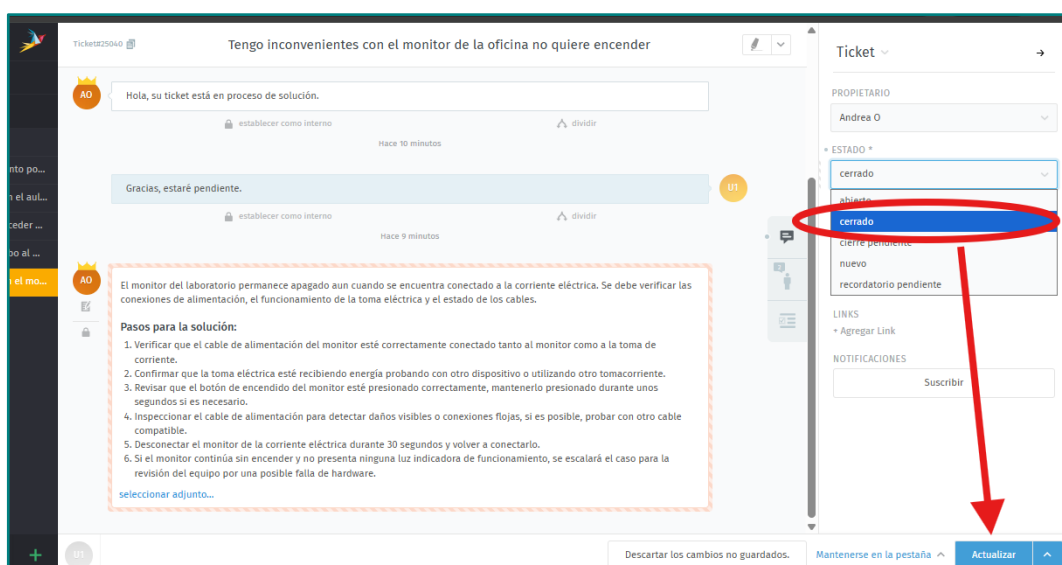


Figura 32. Solución del incidente, actualización del estado del ticket a cerrado.

3.4.3. Implementación del Modal

Como parte final del flujo del sistema, al actualizar el estado del ticket a cerrado se habilita de manera inmediata el modal desarrollado sobre la interfaz de Zammad, esta ventana emergente muestra dos secciones (Figura 33): 1. De la búsqueda por similitud semántica con S-bert presenta el top 3 de las soluciones similares, 2. Permite agregar nuevos términos a la ontología para afinar su estructura.

Soluciones Similares Encontradas 45% mejor match

Se encontraron 3 resultado(s). Haz clic en una tarjeta para ver y editar antes de reemplazar.

CONTENIDO DEL TICKET CERRADO (EDITABLE)

CATEGORÍA / SUBCATEGORÍA DE LA SOLUCIÓN

Gestion de Soporte

Asistencia Técnica

TÍTULO

Tengo inconvenientes con el monitor de la oficina no quiere

SOLUCIÓN / CONTENIDO

El monitor está enchufado pero no quiere encender

PASOS DETALLADOS

Verificar que el cable de alimentación del monitor esté correctamente conectado tanto al monitor como a la toma de corriente.

CONOCIMIENTO SIMILAR — CLIC PARA EXPANDIR Y EDITAR

1 45% Pantalla negra al encender pc
Título: 41% | Texto: 45% | Pasos: 54% ID #138
Gestion de Soporte - Asistencia Técnica

2 45% La pantalla compartida se ve negra o cong...
Título: 35% | Texto: 60% | Pasos: 45% ID #340
Gestion de Soporte - Asistencia Técnica

3 44% Zoom no funciona correctamente en un di...
Título: 36% | Texto: 56% | Pasos: 48% ID #334
Gestion de Soporte - Asistencia Técnica

No guardar Guardar como nueva solución

Agregar términos a la ontología

Figura 33. Interfaz del modal

- **Sección 1: Soluciones Similares Encontradas**

Se observa en la Figura 34 un panel dual en donde el lado izquierdo presenta la categorización relacionada al incidente obtenida desde el asistente virtual (se puede modificar si fuera necesario para guardar o actualizar correctamente el conocimiento) + la información capturada del ticket resuelto. El panel derecho muestra el top 3 de soluciones similares encontradas que se presentan en tarjetas expandibles para ver detalles de cada una de las soluciones, y que a su vez le permitirá al técnico decidir si la solución capturada en ese ticket debe ser guardada como nuevo conocimiento, reemplazar una de esas 3 soluciones similares por la nueva, actualizar algún detalle de una de las soluciones, o no guardar nada.

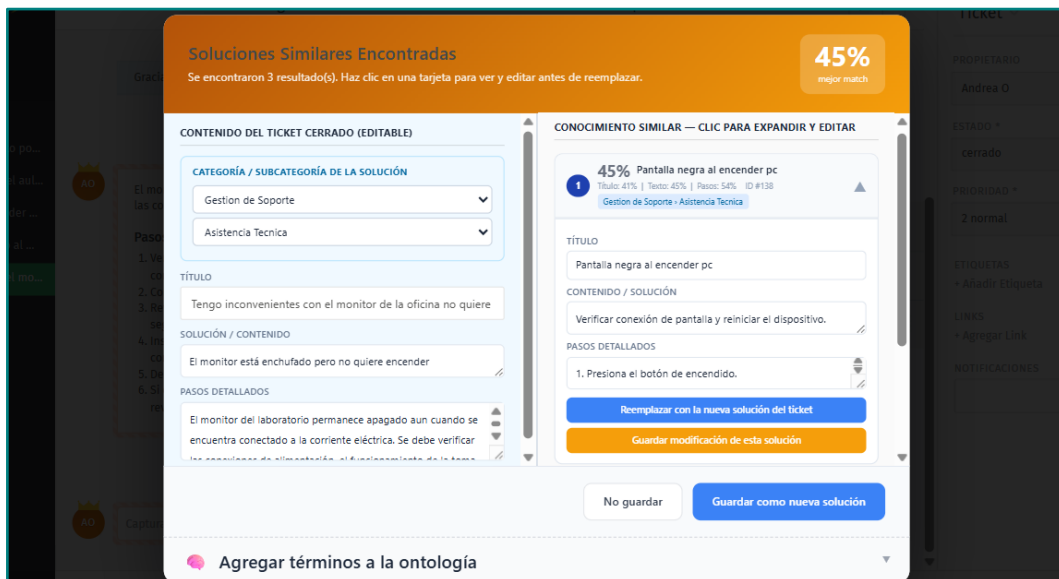


Figura 34. Modal - interfaz de top 3 de soluciones similares encontradas

• Sección 2: Retroalimentación de la Estructura Ontológica

Como segundo paso en la Figura 35 tenemos una interfaz que nos permite ingresar nuevos términos a la ontología para mantener su estructura actualizado en casos aún no han sido cubiertos o casos sobre incidentes de servicios nuevos (este método de retroalimentación ontológica es opcional utilizarlo), de esta forma en futuras consultas similares a ese nuevo caso puedan ser mejor clasificadas desde el asistente virtual lo que incluso podría minorar los casos de incidentes escalados a un ticket de soporte técnico

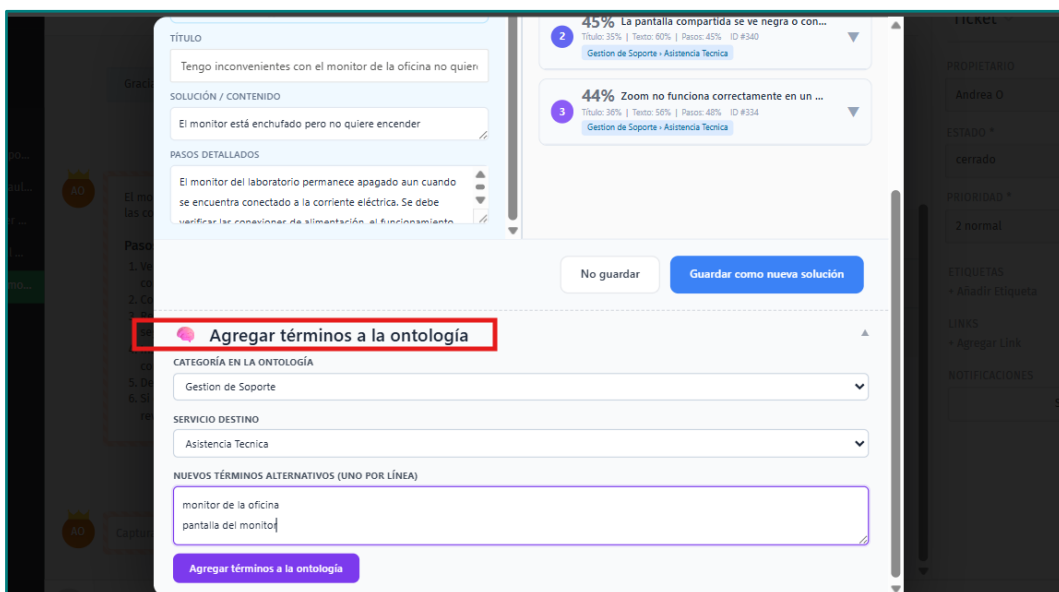


Figura 35. Modal - interfaz de retroalimentación de la estructura ontológica

Para el ingreso de nuevos términos alternativos en la ontología (Figura 36), el sistema primero realiza una normalización del texto ingresado para luego hacer una comparación exacta utilizando un operador de pertinencia “in” del lenguaje Python, este operador permite comparar el termino nuevo que se ingresó contra los elementos ya guardados en la ontología de tal forma se evitar la duplicación de términos y mantiene su estructura limpia.

Figura 36. Modal - previene el ingreso de términos repetidos a la estructura ontológica.

3.5. FASE DE EVALUACION Y VALIDACION

3.5.1. Pruebas Experimentales

Para esta fase se evaluó el correcto funcionamiento del sistema cognitivo, realizando pruebas controladas en un entorno real recopilando 264 interacciones generadas por usuarios pertenecientes a la comunidad universitaria lo que permitió obtener una muestra representativa y diversa para analizar. Estas pruebas fueron procesadas inicialmente por el modelo de clasificación Bert sin base ontológica y posteriormente evaluadas con soporte ontológico para comparar de manera semántica de ambas fases en el proceso de clasificación permitiendo cumplir con el **quinto objetivo específico**.

En el contenido de la Tabla 15 se aprecian los resultados de las clasificaciones asignadas por el modelo “Bert” y sus porcentajes de confianza, antes de la intervención ontológica. A pesar de que Bert logra una clasificación efectiva en la

mayoría de casos evaluados, existen casos especiales en donde la ambigüedad de los mensajes del usuario provoca una clasificación incorrecta.

Clasificación de Bert antes de la ontología - 264 incidencias			
Servicio Detectado	Categoría	Nº	%
Aulas Virtuales	Gestión de Accesos	43	16,3%
Canales Digitales	Gestión de Comunicación	34	12,9%
Moodle	Gestión de Sistemas Académicos	31	11,7%
Asistencia Técnica	Gestión de Soporte	28	10,6%
Calificaciones	Gestión de Sistemas Académicos	22	8,3%
Matriculación	Gestión de Sistemas Académicos	20	7,6%
Sistema Institucional	Gestión de Accesos	19	7,2%
Correo Institucional	Gestión de Accesos	14	5,3%
Sistema de Gestión Académica	Gestión de Soporte	12	4,5%
Plataforma Zoom	Gestión de Accesos	12	4,5%
Solicitud De Documentos	Gestión de Comunicación	9	3,4%
Email Institucional	Gestión de Soporte	8	3,0%
Asistente Virtual	Gestión de Soporte	7	2,7%
Silabos	Gestión de Sistemas Academicos	4	1,5%
Asistencia Tecnica	Desconocida	1	0,4%
Total		264	100%
Nota: En la categoría registrada como “Desconocida” Bert no logra encontrar la categoría relacionada al servicio por algún falla en la búsqueda.			

Tabla 15. Resumen de clasificaciones de Bert antes de la ontología - 264 incidencias

En cuanto a la clasificación final del sistema, después del resultado de la ontología (Tabla 16) se obtuvo que en ciertos casos se alcanzó una precisión superior o igual al 90%, casos en donde la ontología procedió a corregir o complementar la clasificación realizada por Bert para asignar una clasificación más coherente según el incidente presentado.

Clasificación Final después de la ontología - 264 incidencias			
Servicio final	Categoría	Nº	%
Canales Digitales	Gestión de Comunicación	42	15,9%
Aulas Virtuales	Gestión de Accesos	40	15,2%
Asistencia Tecnica	Gestión de Soporte	36	13,6%

Clasificación Final después de la ontología - 264 incidencias			
Servicio final	Categoría	Nº	%
Moodle	Gestión de Sistemas Académicos	25	9,5%
Calificaciones	Gestión de Sistemas Académicos	22	8,3%
Matriculacion	Gestión de Sistemas Académicos	20	7,6%
Correo Institucional	Gestión de Accesos	17	6,4%
Sistema de Gestion Academica	Gestión de Soporte	16	6,1%
Sistema Institucional	Gestión de Accesos	14	5,3%
Solicitud De Documentos	Gestión de Comunicación	9	3,4%
Plataforma Zoom	Gestión de Accesos	8	3,0%
Asistente Virtual	Gestión de Soporte	6	2,3%
Email Institucional	Gestión de Soporte	5	1,9%
Silabos	Gestión de Sistemas Académicos	4	1,5%
TOTAL		264	100%
Nota: La ontología realiza sus 264 clasificaciones, donde notamos una diferencia con Bert en la cantidad clasificaciones detectadas por servicio y categorías			

Tabla 16. Resumen de clasificaciones de la ontología - 264 incidencias

A continuación, sobre los 264 casos evaluados la ontología realiza su intervención en 50, donde su precisión supera el umbral del 90% como regla estricta para que realice una corrección, de estos 50 casos 49 fueron corregidos y 1 caso complementado, en la Tabla 17 se muestra una selección de intervenciones más representativas que hizo la ontología para la clasificación final. Cabe recalcar que entre estos casos se presentaron 2 especiales.

En el caso n°16 el incidente del usuario solo describe una palabra “admisión” siendo insuficiente o demasiado corta y sin contexto, por lo que ambos clasificadores (Bert y ontología) tuvieron resultados diferentes no erróneos, o en el caso n°42 en donde el incidente presenta una descripción amplia y confusa donde une dos frases contradictorias para el sistema mencionando: “cómo funciona el sistema de nivelación” y “donde habrán las actividades y el contenido de clases”, lo que resulta en clasificaciones contradictorias y no acertadas entre Bert y la ontología. Estas muestras indican que, aunque ambos clasificadores son eficientes existen casos en donde la incidencia no muestra detalles claros o suficientes y el sistema determina con certeza una clasificación que aun que puede ser coherente podría no reflejar la intención real del usuario.

N°	Mensaje	Bert Servicio	Bert Categoría	Ontología Servicio	Ontología Categoría	Confianza Bert	Precisión Ontología	Tipo Intervención
1	No puedo adjuntar documentos en mi correo, es como si la opción estuviese bloqueada	Sistema de Gestión Académica	Gestión de Soporte	Email Institucional	Gestión de Soporte	97%	93.0%	Corrigió Servicio
7	Quiero saber mi email institucional	Email Institucional	Gestión de Soporte	Correo Institucional	Gestión de Accesos	99%	100.0%	Corrigió Servicio y Categoría
8	Me piden mi certificado de bachiller notariado, como hago eso?	Solicitud De Documentos	Gestión de Comunicación	Matriculación	Gestión de Sistemas Académicos	99%	100.0%	Corrigió Servicio y Categoría
9	los servicios de microsoft junto al correo institucional me dicen que estan expirados	Email Institucional	Gestión de Soporte	Asistencia Técnica	Gestión de Soporte	99%	95.0%	Corrigió Servicio
10	al momento de ingresar al moodle no me salen todas las materias a las que estoy matriculado	Aulas Virtuales	Gestión de Accesos	Moodle	Gestión de Sistemas Académicos	90%	90.0%	Corrigió Servicio y Categoría
11	donde puedo encontrar información sobre el día del examen	Moodle	Gestión de Sistemas Académicos	Canales Digitales	Gestión de Comunicación	99%	100.0%	Corrigió Servicio y Categoría
12	Como puedo ver que paralelo soy y en que horario me toca si matutino o vespertino?	Canales Digitales	Gestión de Comunicación	Sistema de Gestión Académica	Gestión de Soporte	99%	100.0%	Corrigió Servicio y Categoría
Casos especiales en la clasificación de la ontología								
16	admision	Asistencia Técnica	Gestión de Soporte	Aulas Virtuales	Gestión de Accesos	99%	100.0%	Corrigió Servicio y Categoría
42	Quisiera saber como funciona el sistema de nivelacion donde habran las actividades y el contenido de clases	Aulas Virtuales	Gestión de Accesos	Canales Digitales	Gestión de Comunicación	45%	100.0%	Corrigió Servicio y Categoría
50	no puedo compartir pantalla en zoom	Asistencia Técnica	Desconocida	Asistencia Técnica	Gestión de Soporte	100%	100.0%	Complemento - Categoría faltante

Tabla 17. Muestra representativa de las intervenciones de la ontología

3.5.2. Evaluación Ontológica y del Sistema Cognitivo Desarrollado

- **Indicadores de Calidad Ontológica**

Consistencia Estructural

Dentro de la estructura ontología se definieron reglas lógicas que definen como está organizado jerárquicamente el conocimiento y que relaciones tienen. Para comprobar la consistencia de dichas reglas definidas se utilizó una herramienta de inferencia lógica, el razonador de lógico “HermiT” que viene integrado por defecto en la aplicación de Protégé.

Se realizaron 2 pruebas para analizar la estructura de la ontología y comprobar la pertinencia de la misma: Para la **primera prueba** se evaluó el archivo original de la ontología aplicada en este proyecto, donde queda evidencia como el razonador terminó su evaluación de manera exitosa, en la Figura 37 se observa del lado izquierdo la sección “*Class hierarchy*” que muestra la estructura jerárquica definida en la ontología y del lado izquierdo la sección “*Class hierarchy(inferred)*” donde se visualiza nuevamente lo definido en la ontología más la deducción de HermiT que como se puede apreciar lo reconstruyó correctamente y sin errores, en 131 milisegundos Figura 38 lo que fue muy rápido e indica que la ontología goza de una buena estructura sin inconsistencias lógicas.

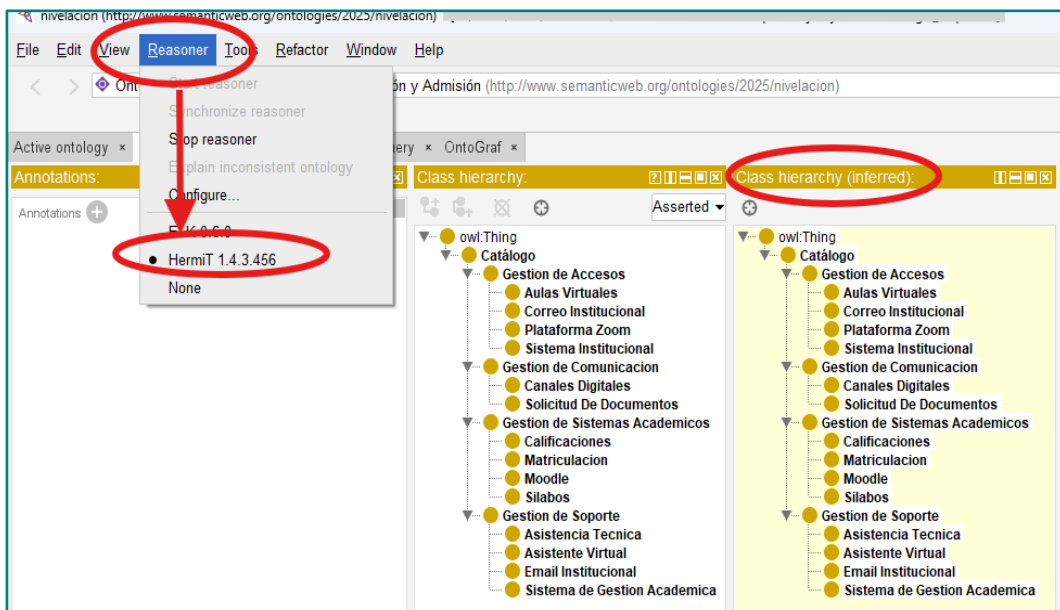


Figura 37. Vista 1 – Resultados prueba 1 de inferencia lógica con el razonador HermiT

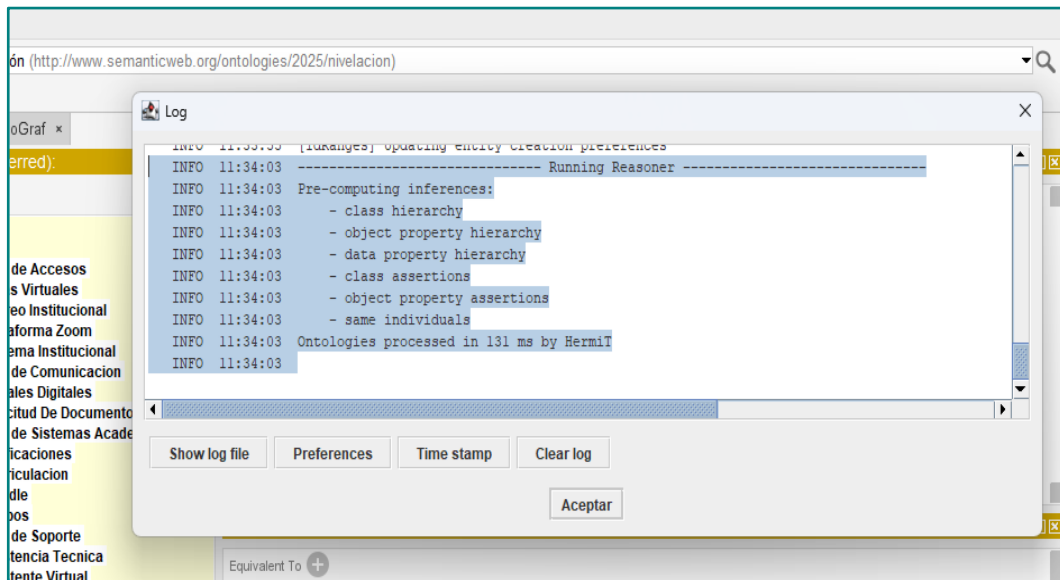


Figura 38. Vista 2 - Resultados prueba 1 de inferencia lógica con razonador HermiT

Como segunda prueba (Figura 39), se utilizó una copia del archivo original de la ontología en donde se realizaron alteraciones deliberadas con el fin de observar las deducciones de HermiT el cual demuestran la existencia de las inconsistencias, un servicio esta repetido en dos categorías y muestra en rojo el error de las clases mal definidas, esto confirma una vez más que la estructura original de la ontología implementada está correctamente construida y cumple con sus reglas lógicas.

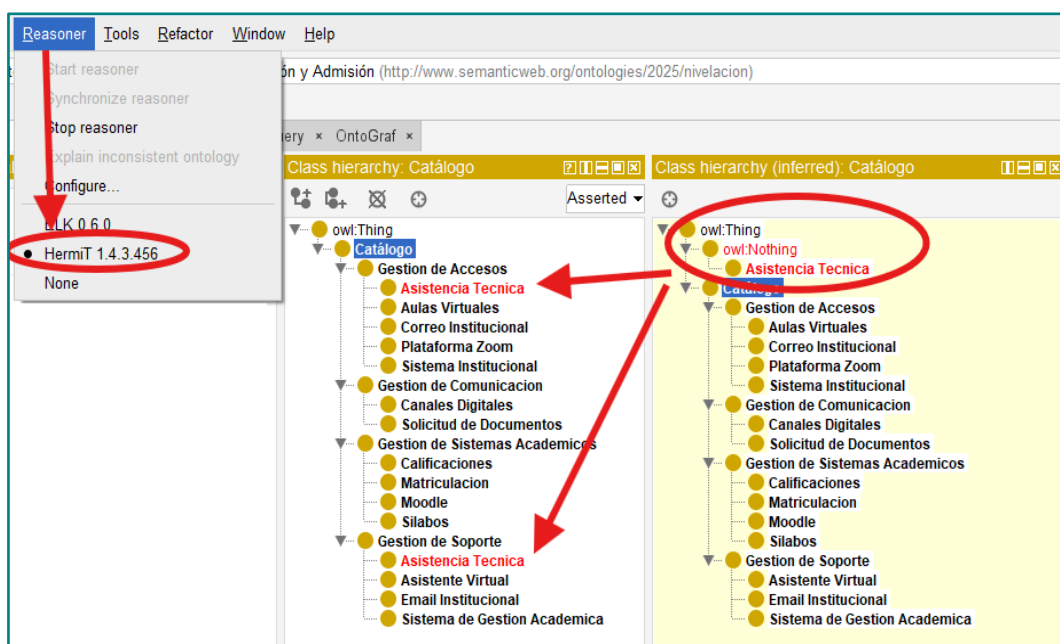


Figura 39. Resultados de prueba 2 premeditada con razonador HermiT

Cobertura del Catálogo

La métrica de cobertura nos permite identificar si los 14 servicios y 4 categorías se encuentran definidos en la ontología, en la Figura 40 observamos los resultados de la comparación entre el contenido del catálogo en la base de datos y las clases definidas en la ontología, el cual se podría considerar como cubiertas solo si existe la definición con la etiqueta equivalente de dichas clases `rdfs:label`” (ANEXO 2).

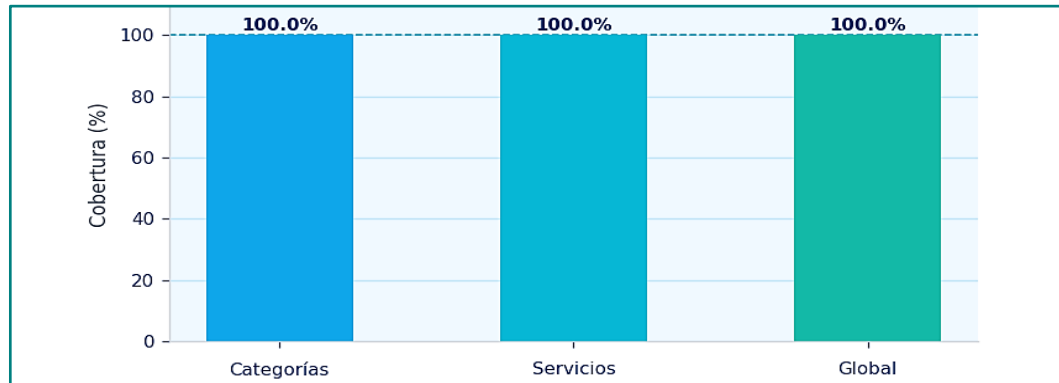


Figura 40. Cobertura ontológica

Compleitud Descriptiva

Para evaluar la existencia de los atributos de la ontología utilizamos el método de completitud descriptiva, se comprueba por cada clase de la ontología si tiene al menos un valor que la identifique. En esta evaluación (Figura 41) se muestra como la estructura ontológica obtiene un 100% de completitud descriptiva demostrando que cada concepto del catálogo de servicios está bien definido en los parámetros: “`rdfs:label`” nombre de la clase (ANEXO 2) “`skos:definition`” definición formal de la clase (ANEXO 3), “`skos:altLabel`” términos alternativos (ANEXO 4).

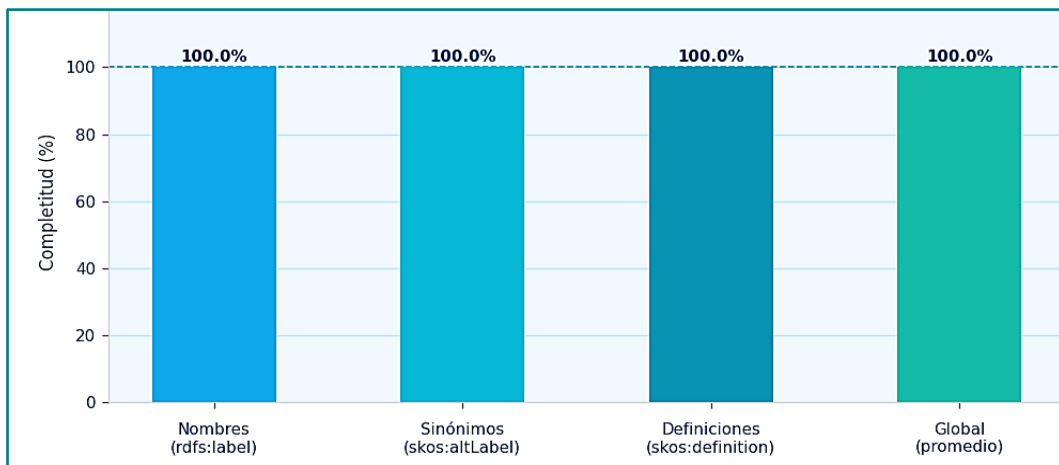


Figura 41. Compleitud descriptiva en la estructura ontológica

Concisión

Métrica de concisión posibilita la medición e identificación de redundancias y vacíos en la ontología, para medir concisión se verificaron problemas de calidad con las etiquetas repetidas y en más de una clase o identificar clases huérfanas es decir sin propiedad descriptiva (label, altlabel, definition) obteniendo un 100% de efectividad, es decir no se presentaron deficiencias en la estructura.(Figura 42).

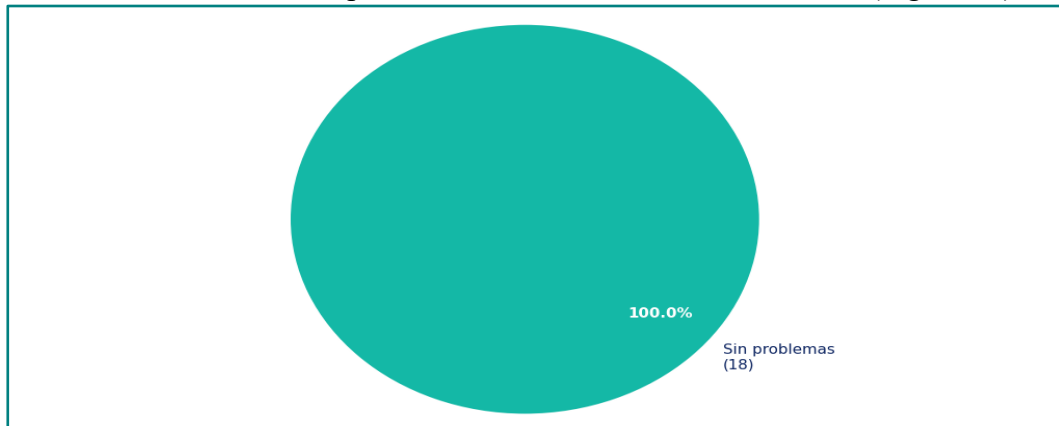


Figura 42. Concisión de la estructura ontológica

Relevancia Ontológica

Después de las pruebas experimentales controladas para evaluar el sistema podemos conocer la relevancia que ha tenido la ontología dentro del contexto de la gestión del conocimiento, la efectividad de su interacción en la recuperación y mantenimiento del conocimiento para lo cual se evaluaron 246 interacciones donde la ontología (valida, complementa o corrige la clasificación de Bert). En los resultados notamos claramente la utilidad de la ontología, logrando alcanzar un 100% de efectividad que indica que su interacción siempre formó parte de la del proceso para llegar a la recuperación del conocimiento (visualizar Figura 43).

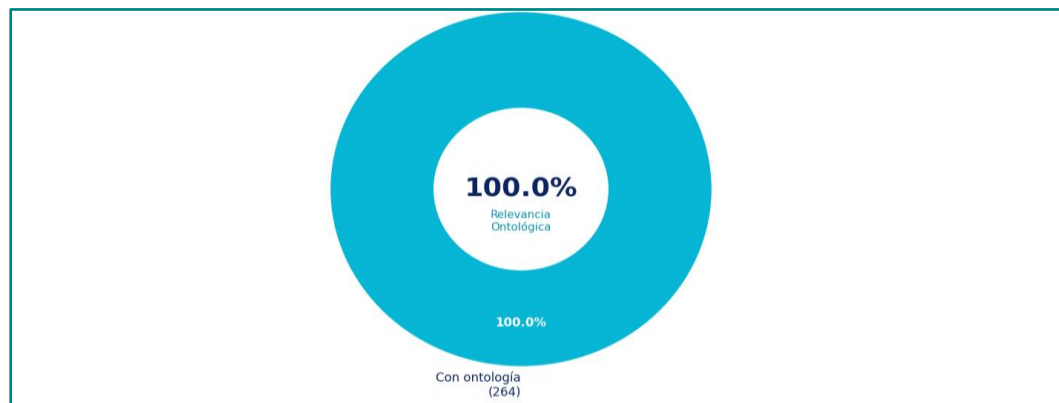


Figura 43. Proporción de interacciones con intervención ontológica

Según los registros de la intervención ontológica por cada uno de los 264 incidentes, actuó en 50 casos (19%) “corrige” 49 casos, 1 caso “complementa” y 236 “valida” la clasificación de Bert revisando la estructura semántica ontológica, lo que indica que ejecuta correctamente su función dentro del proyecto (Figura 44).

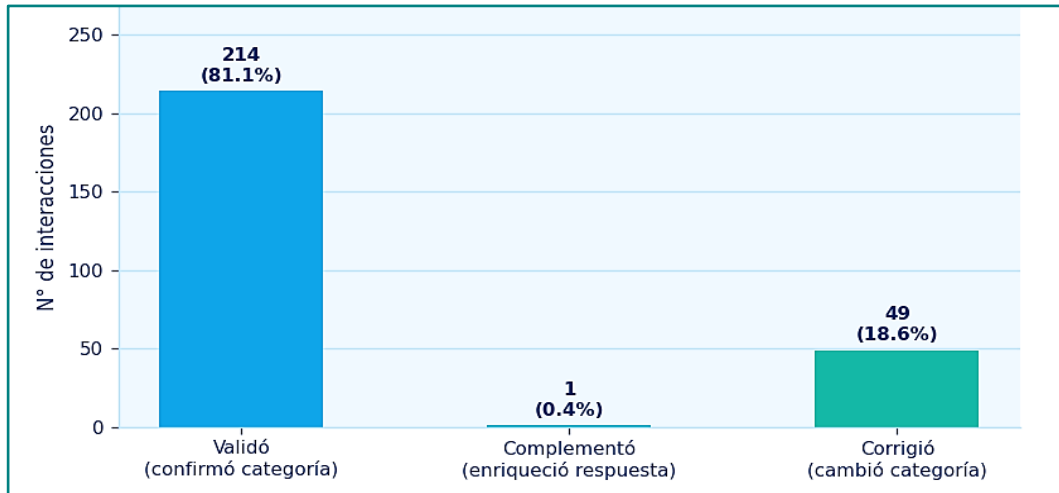


Figura 44. Distribución de interacciones por tipo de intervención de la ontología

- **Indicadores de Desempeño del Sistema:**

Para evaluar la eficacia del sistema cognitivo ante los incidentes tecnológicos presentados, es necesario utilizar métricas que nos permitan conocer la efectividad y el cumplimiento de su propósito, por lo cual se realizaron pruebas experimentales en un ambiente controlado donde se obtuvo 246 interacciones a evaluar (ANEXO 8).

Tasa de Aceptación

Identifica cuantos usuarios eligieron al menos 1 solución, de las 264 interacciones evaluadas en 198 casos el usuario seleccionó al menos una de las 3 soluciones presentadas por el sistema, representando una tasa de aceptación del 75% (Figura 45) lo que significa que 3 de cada 4 usuarios encontraron al menos una opción relevante dentro las presentadas.

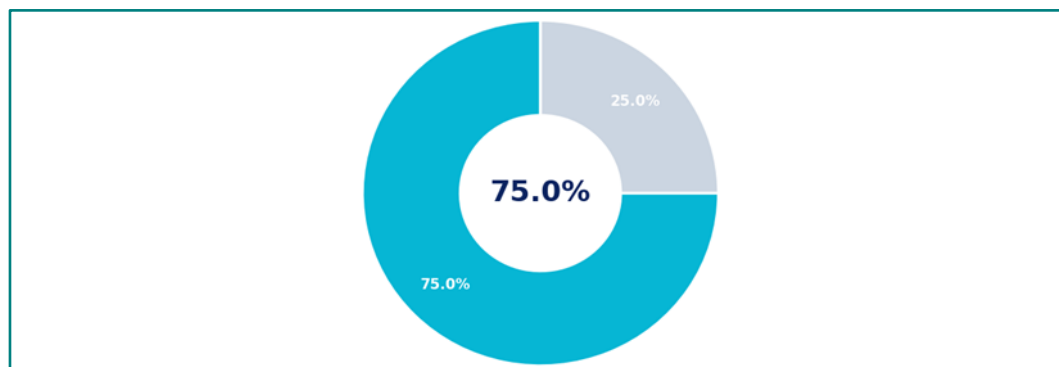


Figura 45. Tasa de aceptación del desempeño del sistema

Hit@3

Porcentaje de utilidad del sistema, de las 264 interacciones se identificación que 128 registraron la retroalimentación por parte del usuario y 126 casos registran que al menos una de las soluciones presentadas del top 3 fue útil, alcanzando un Hit@3 con un 98,44% reflejando una alta efectividad practica del sistema (Figura 46).

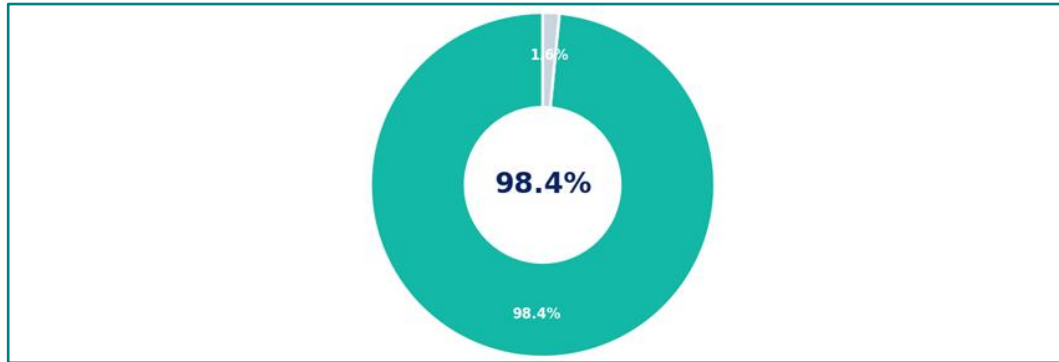


Figura 46. Hit@3 - desempeño del sistema

MRR

El *Rango reciproco medio* indica en promedio la posición en la que aparece la solución útil, y según los resultados el MRR obtenido fue de 0,6640 (Figura 47) indicando que en su mayoría el sistema sugiere la respuesta más acertada en una posición cercana o en la primera posición, según las medidas la posición 1= 1.0 y la posición 2 o 3 = 0,5.

MAP

Posición media promedia calcula la consistencia del ranking en general del orden entregado de las soluciones por pertinencia, MAP alcanzó un 0,66772 (Figura 47) lo que quiere decir que el sistema recupera las soluciones más relevantes y además las posiciones en los primeros lugares del top 3 indicando que 'posee una buena capacidad para ordenar los resultados por pertinencia.

nDCG@3

Ganancia acumulada descontada normalizada evalúa la calificación puesta por el usuario (1 puntuación más baja y 5 puntuación excelente) de las soluciones presentadas, el indicador incorpora en el cálculo la calificación otorgada por el usuario y una penalización de las soluciones mejores valoradas que aparecieron en las mejores posiciones, lo cual mostró un 0,7508 sobre 1.0 lo que indica que el sistema presenta un ranking de calidad alta (Figura 47).

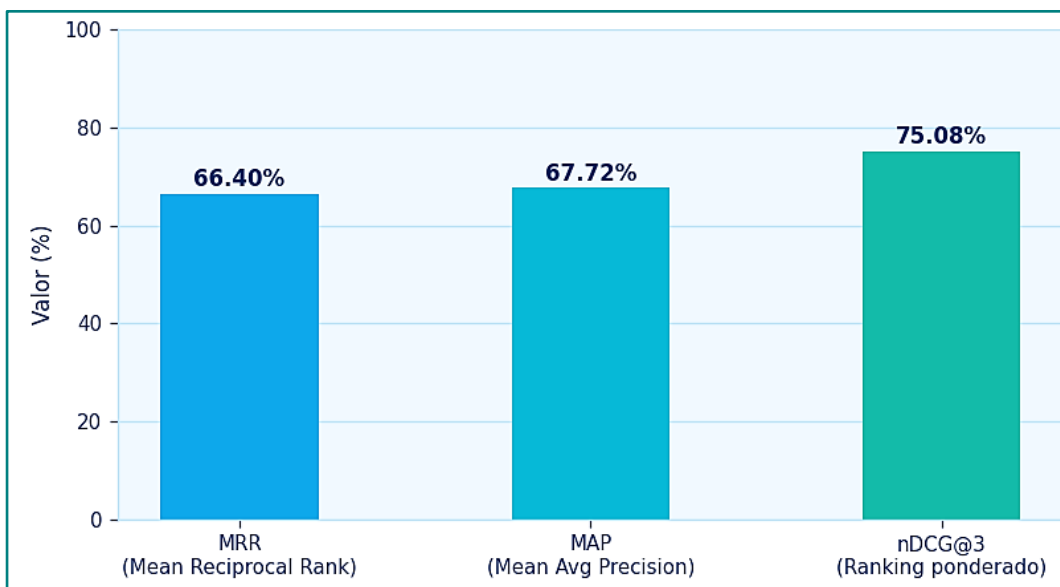


Figura 47. Métricas de calidad del ranking de Top 3 de respuestas entregadas

Precisión Semántica (%)

Demostración de la pertinencia del sistema en la recuperación de los conocimientos presentados como soluciones efectivas al usuario, aplicando un umbral de similitud del coseno del 40% las soluciones presentadas matemáticamente similares a la consulta realizada dieron un 78,54% (Figura 48), lo que significa que en un promedio de 2,36 de las 3 soluciones mostradas son similares a la consulta, corroborando la pertinencia del motor de recuperación semántica S-bert.

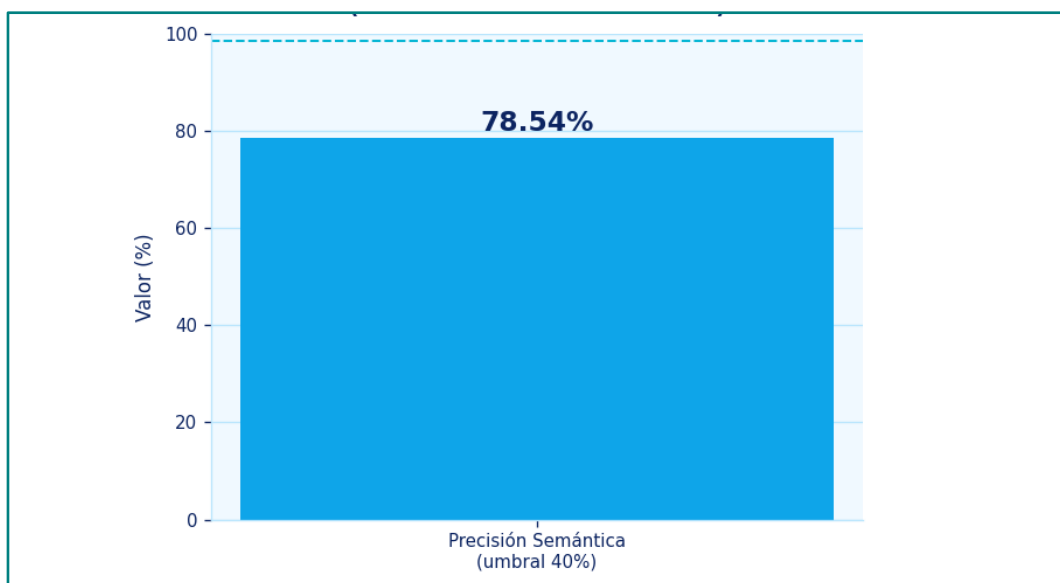


Figura 48. Métrica de recuperación semántica

CONCLUSIONES

Según los resultados obtenidos en este proyecto, se puede determinar que desde la fase de adquisición del conocimiento se logró constituir efectivamente el catálogo de servicios tecnológicos lo que reveló que el conocimiento institucional sobre incidentes tecnológicos durante el proceso de admisión y nivelación universitaria se encontraba disperso en su mayoría, permitiendo la formalización de esta información como proceso fundamental para que el sistema desarrollado integre un vocabulario ordenado según la estructura de los servicios del dominio y lo que posteriormente guiaría el diseño del modelo ontológico así como el entrenamiento del algoritmo de clasificación automática “Bert”.

A partir de esto, se construyó una base ontológica que demostró una cobertura global del 100% de las 18 clases de la estructura del catálogo (4 categorías y 14 servicios) además de una completitud descriptiva y concisión del 100% capaz de representar formal, válida y coherentemente el catálogo de servicios tecnológicos mediante jerarquías de conceptos con relaciones explícitas que brindan una guía organizada para la correcta clasificación de los incidentes, determinada no solo como un componente que estructura la organización del conocimiento sino que también es capaz de detectar y corregir clasificaciones equívocas del modelo Bert en casos donde la confianza del mismo no muestra niveles altos de certeza, clasificación importante por definir para la correcta recuperación de soluciones.

La estructura del sistema elaborado que une componentes tecnológicos efectivos como el clasificador Bert y la validación ontológica demuestran que sus resultados son mucho mejor a los que cualquiera de las dos herramientas lograría de manera independiente, afirmación fundamentada en los resultados de las pruebas realizadas en donde se obtuvo que a pesar de que el modelo Bert demuestra una capacidad robusta para interpretar incidentes y clasificarlos al servicio perteneciente según la estructura ontológica se detectaron inconsistencias en un 19% que corresponde a un total de 50 casos del total de las 264 interacciones analizadas por lo que se procede a corregir el resultado del modelo para garantizar la integridad en la recuperación semántica del conocimiento.

La integración de los elementos que conforman este sistema cognitivo desarrollado bajo una arquitectura modular evidenció que la operación de dos flujos complementarios donde el primero es dirigido a la atención del usuario que recibe las consultas, clasifica y devuelve soluciones de manera eficaz y automatizada con una tasa de aceptación del 75%, un Hit@3 de 98,44%, y un segundo flujo útil para el técnico de soporte que soluciona los casos especiales escalados a un ticket permitiéndole capturar cada nuevo conocimiento e incorporarlo en la base, cerrando un ciclo de aprendizaje continuo que valida la arquitectura diseñada como una técnica viable y escalable para los entornos educativos en un contexto de soporte tecnológico.

Los resultados de las evaluaciones realizadas al sistema validan que la metodología aplicada constituye un marco apropiado para la gestión del conocimiento en entornos educativos, en donde se obtuvo un MRR de 0,6640, Map de 0,66771, y un nDCG@3 de 0,7508, junto a una precisión semántica del 78,54%, llegando a la conclusión que el proceso de gestión del conocimiento basado en ontologías formales y modelos de procesamiento de lenguaje natural representan un factor determinante para la calidad y eficiencia de la gestión de conocimiento generado en los entornos de servicios tecnológicos universitarios.

RECOMENDACIONES

En base a los estudios del presente proyecto se recomienda aprovechar estas tecnologías a un alcance a nivel institucional iniciando por el proceso de adquisición del conocimiento en otros departamentos institucionales como el departamento de Tics, con el fin de obtener más información relevante para alimentar el conocimiento institucional con aquellos registros históricos de incidentes tecnológicos presentados a en la institución según el departamento.

Se sugiere establecer un ciclo de revisión y actualización semestral de la estructura ontológica incorporando nuevos términos que emerjan de los nuevos incidentes reportados con el fin de mantener la correspondencia entre la representación formal del conocimiento y la realidad operativa del dominio. Además, para de garantizar la sostenibilidad del modelo ontológico más allá del alcance actual de este proyecto se recomienda incorporar información proveniente de otros departamentos de la universidad, tomando en cuenta que la estructura actual es claramente replicable a otros dominios de la institución que de igual manera gestionan incidentes tecnológicos de nivel 1 sin necesidad de rediseñar la jerarquía ontológica actual

Para robustecer el sistema neuro-simbólico implementado, se propone una evolución tecnológica que se basa en la técnica **“ontology learning”**, esta técnica permitirá analizar periódicamente las interacciones en el sistema en casos especiales que escalaron a tickets que representan los vacíos de cobertura en el sistema, dando paso a la automatización del ingreso de términos nuevos para la estructura ontológica según los incidente solucionados eficazmente, reduciendo la dependencia a la retroalimentación manual solo para casos especiales.

Para la automatización de la recuperación semántica se propone una arquitectura RAG (Retrieval- Augmented Generation) en la que en base al top de soluciones recuperadas por el modelo S-Bert sirvan como contexto para que un modelo de lenguaje (adaptados a su versión en español) como por ejemplo Mistral-7B o Llama 3.1 generen respuestas adaptadas a cada consulta.

Para respaldar la continuidad del servicio del sistema se recomienda implementar un proceso de monitoreo para visualizar el comportamiento del sistema en periodos

de alta demanda y con base a los resultados se realicen mantenimientos programados en tiempos de baja demanda académica como reentrenamientos, actualizaciones o migraciones sin afectar la disponibilidad del sistema en etapas activas del proceso educativo.

REFERENCIAS

- [1] D. y. G. J. Rodríguez-Gómez, «Desentrañando la creación y gestión del conocimiento en las organizaciones educativas: barreras y facilitadores,» *Knowledge Management Research & Practice*, vol. 13, nº 2, pp. 149 - 159, 19 Dic 2015.
- [2] S. P. S. y. T. F. Quarchioni, «Knowledge management in higher education: a literature review and further research avenues,» *Knowledge Management Research & Practice*, vol. 20, nº 4, pp. 304 - 319, 02 Mar 2020.
- [3] P. V. R. y. J. Rowley, «Knowledge management in higher education institutions: enablers and barriers in Mauritius,» *The Learning Organization*, vol. 24, nº 5, pp. 366- 377, 10 July 2017.
- [4] U. E. P. d. S. Elena, «Sitio oficial de la Universidad Estatal Península de Santa Elena,» 2025. [En línea]. Available: <https://www.upse.edu.ec/>.
- [5] U. d. N. y. Admisión, «Nivelación y Admisión,» 2025. [En línea]. Available: <https://www.upse.edu.ec/nivelacion-admision/index.php/quienes-somosr/acerca-de-nosotros>.
- [6] B. B. y. F. Getahun, «"Ontology-driven Intelligent IT Incident Management Model",» *International Journal of Information Technology and Computer Science (IJITCS)*, vol. Vol. 15, nº No. 1, p. pp. 30–41, 2023.
- [7] J. C. H. P. y. J. M. P. Hernández, «Creación e implementación de una aplicación web ontológica a partir de datos enlazados,» Universidad Francisco de Paula Santander, Cúcuta, Colombia , 2024.
- [8] J. Á. Tomala Tigreros, «Desarrollo de una interfaz basada en algoritmos de procesamiento de lenguaje natural NPL para la gestión de conocimiento.,» Universidad Estatal Península de Santa Elena, La Libertad, 2025.

- [9] F. I. B. P. Allen, «Standardizing Knowledge Engineering Practices with a Reference Architecture,» *Transactions on Graph Data and Knowledge*, vol. 2, n° 1, p. 23, 2024.
- [10] M. U. y. M. King, «Towards a Methodology for Building Ontologies,» de *Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canadá, 1995.
- [11] T. N. V. N. M. V. T. D. N. T. H. Pham, «The impacts of knowledge management enablers and knowledge management processes on university performance in Vietnam,» *Knowledge Management Research & Practice*, vol. 21, n° 3, pp. 512- 524, 2022.
- [12] M. C. S. M. Eleonora Santos, «Sustainable Enablers of Knowledge Management Strategies in a Higher Education Institution,» *Sustainability*, vol. 16, n° 12, 2024.
- [13] A. Rommel, «Impacto del sistema de mesa de ayuda basado en el conocimiento: chatbot de Facebook Messenger para el soporte informático estudiantil en Pamantasan ng Lungsod ng Valenzuela,» de 10.^a *Conferencia Internacional sobre Tecnologías de la Información y de Redes (ICINT) de 2025*, Melbourne, Australia, 2025.
- [14] A. P. N. Herrera, «Propuesta de diseño de ontología para formalizar conocimiento y gestionar respuestas en una interfaz conversacional de la Iniciativa Open Campus,» *XV Conferencia Latinoamericana de Tecnologías de Aprendizaje (LACLO)*, pp. 1-8, 2020.
- [15] C. M. d. O. R. a. A. S. L. H. de A. Monteiro, «An Information System for Law Integrating Ontological Bases with a Legal Reasoner Chatbot,» *Brazilian Symposium on Information Systems*, vol. 44, pp. 1-8, 2022.

- [16] A. C. Mendoza de los Santos, «Implementación de un chatbot basado en modelo de lenguaje de inteligencia artificial para responder preguntas frecuentes de estudiantes universitarios,» *Revista científica de sistemas e informática*, 2023.
- [17] S. J. R. y. P. Norvig, *Artificial Intelligence: A Modern Approach*, Upper Saddle River, NJ, USA: Pearson: Pearson, 2020.
- [18] M. T. Biagetti, «Ontologies as Knowledge Organization Systems,» Knowledge Organization,» *Knowledge Organization*, vol. 48, n° 2, pp. pp.152-176, 2021.
- [19] S. N. d. l. A. P. y. Planificación, «Secretaría Nacional de la Administración Pública y Planificación,» 08 2025. [En línea]. Available: <https://www.planificacion.gob.ec/plan-nacional-de-desarrollo-2025-2029-ecuador-no-se-detiene/>.
- [20] T. Y. Zhu Ping, «Método de construcción de digitalización de sistemas de bases de conocimiento a gran escala,» de *Conferencia Internacional de Ciencias de la Computación y Computacionales (ICCCS)*, Greater Noida, India, 2015.
- [21] S. S. D. P. S. B. Neelam Chandolika, «Creación semiautomatizada de ontologías mediante aprendizaje profundo para proporcionar búsqueda de conocimiento específico del dominio en el idioma marathi,» de *Conferencia Internacional de Aprendizaje Automático Aplicado (ICAML)*, Bhubaneswar, India, 2019.
- [22] C. F. P. B. Roberto Hernández, *Metodología de la investigación*, México, D.F.: McGraw-Hill Interamericana, 2014.
- [23] D. H. Romero García, «Elaboración de un proyecto de investigación con metodología cualitativa,» *Enfermería Intensiva*, vol. 32, pp. 164-169, 021.

- [24] O. N. ., K. O. Naoto Kobayashi, «Formulación de la estrategia de investigación y evaluación sintética de la investigación basada en dicha estrategia,» *Instituto Nacional de Ciencia y Tecnología Industrial Avanzada (AIST)*, vol. 4, nº 1, pp. 11-25, 2011.
- [25] A. B. Albarran, «Population Sample,» de *The SAGE Encyclopedia of Communication Research Methods*, Thousand Oaks, CA, SAGE Publications, Inc., 2017, pp. pp. 1224-1226.
- [26] S. M. R. A. A Etikan, «Comparison of Convenience Sampling and Purposive Sampling,» *American Journal of Theoretical and Applied Statistics*, vol. 5, nº 1, pp. 1-4, 2016.
- [27] L. S. S. Neha Jain, «Una ontología basada en la metodología propuesta por Ushold y King,» *International Journal of Synthetic Emotions*, vol. 7, nº 1, pp. 13-26, 2016.
- [28] H. J. V. Meter, «Revising the DIKW Pyramid and the Real Relationship Between Data, Information, Knowledge and Wisdom,» *Willamette University*, vol. 2, nº 2, pp. 69-80, 2020.
- [29] E. Shange, «Conocimiento sobre el conocimiento,» *Edward Elgar Publishing eBooks*, pp. 8-50, 2022.
- [30] G. Y. K. Akrofi, «Compartición de conocimiento tácito: revisión de la literatura y marco integrador,» *International Journal of Knowledge Management Studies*, vol. 14, nº 4, pp. 38-415, 2023.
- [31] D. A. y. J. A. J. H. Stokvik, «Conocimiento tácito, aprendizaje organizacional e innovación en las organizaciones,» *Problemas y perspectivas en la gestión*, vol. 14, nº 1, pp. 246-255, 2017.

- [32] F. C. d. C. Rafael Rodrigues Alves, «Gestión del conocimiento y prácticas de explicitación de lo tácito a lo explícito: una revisión sistemática de la literatura de los últimos 20 años,» *Exacta* , vol. 19, nº 4, pp. 911-932, 20021.
- [33] S. Ron, «"Conocimiento tácito" versus "Conocimiento explícito": Enfoques para la práctica de la gestión del conocimiento,» *Edward Elgar Publishing eBooks*, 2003.
- [34] P. Vyas, «Gestión del conocimiento e institutos de educación superior: Revisión y análisis temático,» *Journal of Open Innovation: Technology, Market and Complexity*, vol. vol. 10, nº no. 03, sep 2024.
- [35] A. B. N. I. y. S. A.-S. A. Al-Hunaiyyan, «Un marco cognitivo basado en el conocimiento para el apoyo social y metacognitivo en el aprendizaje móvil,» *Revista interdisciplinaria de información, conocimiento y gestión*, vol. 12, pp. 075-098, 2017.
- [36] M. Sabri, «Mapeo del ciclo de vida del conocimiento a la arquitectura de procesos.,» *Actas de la 24ª Conferencia Europea sobre Gestión del Conocimiento*, vol. 24, nº 2, pp. 1173-1177, 2023.
- [37] O. M. F. Z. R. H. Y. H. I. H. J. Chen, «Incrustaciones de ontologías: un estudio de métodos, aplicaciones y recursos.,» *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. vol. 37, 2025.
- [38] A. F.-I. a. ., M. F. R. G. María Poveda, «LOT: Un marco de ingeniería ontológica orientado a la industria,» *Aplicaciones de la inteligencia artificial en la ingeniería*, vol. 111, 2022.
- [39] B. D. V. T. Adelina Ivanova, «Ontología de dominio del desarrollo social sostenible,» *AIP Conferencia Proceedings*, vol. 2333, nº 1, 03 2021.

- [40] X. H. R. A. J. L. J. S. J. L. F. U. D. K. Xiaochen Zheng, «Un marco de espacio de diseño impulsado por la semántica para acelerar el diseño de sistemas de fabricación de,» *Journal of Intelligent Manufacturing* , vol. 35, pp. 175-198, 2022.
- [41] M. S. N. N. Razieh Adelkhah, «La ontología del procesamiento del lenguaje natural,» de 5.^a *Conferencia Internacional sobre Investigación Web (ICWR)*, Teherán, Irán, 2019.
- [42] W. W. W. C. (W3C), «Marco de descripción de recursos (RDF),» World Wide Web Consortium (W3C), 15 02 2014. [En línea]. Available: <https://www.w3.org/RDF/>. [Último acceso: 2026].
- [43] W. W. W. C. (W3C), «Lenguaje de descripción de vocabulario RDF 1.0: Esquema RDF (RDFS),» World Wide Web Consortium (W3C), 10 02 2004. [En línea]. Available: <https://www.w3.org/2001/sw/wiki/RDFS>. [Último acceso: 2026].
- [44] W. W. W. C. (W3C), «Lenguaje de Ontología Web (OWL),» World Wide Web Consortium (W3C), 11 12 2012. [En línea]. Available: <https://www.w3.org/2001/sw/wiki/OWL>. [Último acceso: 2026].
- [45] W. W. W. C. (W3C), «Sistema Simple de Organización del Conocimiento (SKOS),» World Wide Web Consortium (W3C), 18 08 2009. [En línea]. Available: <https://www.w3.org/2001/sw/wiki/SKOS>. [Último acceso: 2026].
- [46] M. H. Juan Pell, «Un sistema de clasificación de taxonomía y facetas,» *Library Resources & Technical Services*, vol. 61, n° 3, pp. 134-142, 201.
- [47] K. C. Will Orr, «La construcción social de dataset: sobre las prácticas, los procesos y los desafíos de la creación de dataset para el aprendizaje automático,» *New Media & Society* , 2023.

- [48] M. F. G. N. V. L. S. Miloš Bogdanović, «Generación de estructuras de conocimiento a partir de las etiquetas de conjuntos de datos abiertos: un enfoque basado en el análisis formal de conceptos,» *Facta Universitatis, Series: Automatic Control and Robotics*, vol. 20, nº 1, pp. 021-031, 2021.
- [49] S. Z. Z. H. T. R. N. F. B. V. N. J. Kristen Moore, «Una solución integral para la construcción de chatbots basados en recuperación de información,» *Computer Speech & Language*, vol. 83, pp. 3189-3194, 2023.
- [50] S. B. M. D. V. y. S. B. Kumar, «Técnicas útiles y aplicaciones de la inteligencia computacional,» de *Inteligencia computacional en el modelado de software*, De Gruyter, 2022, pp. 19-32.
- [51] L. Ashok, «Más allá de las pruebas: métricas y mediciones para garantizar la excelencia del software,» *Revista Internacional de Investigación en Ingeniería y Ciencias*, vol. 9, nº 5, pp. 30-36, 2025.
- [52] L. B. Y. Y. Jorge Ortiz, «Nivelación propuesta por la SENESCYT: Vivencias en la Universidad Central del Ecuador,» *INNOVA Research Journal*, vol. 2, nº 10, pp. 42-53, 2017.
- [53] N. Plaza, «Desarrollo De Un Sistema Web Para El Soporte Técnico Remoto De Primer Nivel, Orientado A La Gestión De Incidentes Informáticos, Basado En Inteligencia Artificial,» Universidad De Guayaquil.Facultad De Ciencias Matemáticas Y Físicas, Guayaquil, 2017.
- [54] AXELOS, ITIL Foundation, ITIL 4 Edition, Axelos, 2019, 2019.
- [55] ISACA, COBIT 5: Enabling Processes, ISACA,2012, 2012.
- [56] U. Lohmann, «La Cartera de Servicios, el Catálogo de Servicios y los Acuerdos de Servicio,» de *Arquitecturas de digitalización administrativa*, Wiesbaden, Springer Vieweg, 2021, pp. 145-154.

- [57] N. O.-K. V. N. David Máté Hargitai, «La relación entre las tecnologías innovadoras y la gestión del conocimiento en el contexto de la propiedad,» *Space and Society*, vol. 38, nº 2, pp. 29-49, 2024.
- [58] S. M. M. F. Alexander Kell, «Aplicaciones de aprendizaje automático para modelos basados en agentes del mercado eléctrico: una revisión sistemática de la literatura,» de *2022 IEEE International Conference on Big Data (Big Data)*, Osaka, Japón, 2022.
- [59] R. S. D. P. Gagan Gurung, «Retos y avances recientes en el procesamiento del lenguaje natural,» *EDITORIOS CIENTÍFICOS BENTHAM*, pp. 350-369, 2024.
- [60] A. Upadhye, «Un estudio exhaustivo de las técnicas de limpieza de datos de texto: desafíos, métodos y mejores prácticas,» *Revista de Investigación Científica y de Ingeniería*, vol. 7, nº 8, pp. 205-210, 2024.
- [61] S. S. H. Aftan, «Un estudio sobre BERT y sus aplicaciones,» de *20ª Conferencia sobre Aprendizaje y Tecnología (L&T)*, Jeddah, Arabia Saudita, 2023.
- [62] G. C. R. F. J.-H. H. H. K. J. P. José Cañete, «Modelo BERT preentrenado en español y datos de evaluación,» de *PML4DC en ICLR*, 2020.
- [63] D. B. Antonio Carta, «Incrustaciones secuenciales de oraciones para la similitud semántica», *IEEE Symposium Series on Computational Intelligence*,» de *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Xiamen, China, 2019.
- [64] J. G. A. K. A. J. R. J. Samruddhi Deode, «Un enfoque simple para aprender representaciones de oraciones multilingües usando BERT multilingüe,» *arXiv(Universidad de Cornell)*, 2023.

- [65] G. Crocetti, «Similitud textual espacial del coseno,» de *arXiv: Recuperación de información* , New York, 2015.
- [66] Petter Christian Sogn Bjelland, «Búsqueda difusa y sus aplicaciones». 23 05 2019.
- [67] A. d. S. B. H. B. P. D. P. H. K.-U. K. L. L. D. L. P. M. L. d. P. G. P. H. P. G. Z. Tarek Besold, «Aprendizaje y razonamiento neurosimbólico: una revisión e interpretación,» *Cornell University – arXiv*, p. 58, 2017.
- [68] V. M. Dhivya Chandrasekaran, «Evolución de la similitud semántica: una revisión,» *ACM Computing Surveys*, vol. 54, nº 2, pp. 1-37, 2021.
- [69] M. V. Chandrasekaran Dhivya, «“Evolución de la similitud semántica: una revisión,» *arXiv*, p. 29, 2022.
- [70] S. Geng, «Análisis de las diferentes métricas estadísticas en el aprendizaje automático,» *Highlights in Science, Engineering and Technology*, vol. 88, 2024.
- [71] M. Ayyarrappan, «Diseño de API REST para aplicaciones de alto rendimiento,» *Revista de Avances en Investigación del Desarrollo*, 2025.
- [72] S. Ramirez, «FastAPI: El marco de trabajo web moderno y rápido (de alto rendimiento) para crear API con Python basado en sugerencias de tipos estándar de Python,» FastAPI Project, 2021. [En línea]. Available: <https://fastapi.tiangolo.com/>. [Último acceso: 2026].
- [73] S. Assegaff, «Sistema de emisión de tickets del servicio de asistencia técnica de TI Berbasis Web Pada Bagian ERP (planificación de recursos empresariales) PT Perkebunan Nusantara VI,» *Jurnal manajemen sistem informasi*, vol. 7, nº 3, pp. 340-354, 2022.

- [74] A. S. Z. I. T. K. Esada Licina, «HelpAgent: Un marco explicable para la clasificación transparente y centrada en el usuario de los tickets de soporte técnico,» *Procedia Ciencias de la Computación*, vol. 280, pp. 654-662, 2026.
- [75] Z. Community, «"Zammad Community — Home & Download",» Zammad Community, [En línea]. Available: <https://zammad.org/>. [Último acceso: 2026].
- [76] P. G. D. Group, «PostgreSQL Documentation (Version 16),» PostgreSQL Global Development Group, 09 2023. [En línea]. Available: <https://www.postgresql.org/docs/16/>.
- [77] D. T. Aradhya KC, «Bases de datos vectoriales,» *Revista de investigación de acceso abierto en ingeniería y tecnología*, 2024.
- [78] L. T. F. G. W. H. Yang Wen, «Extensión de búsqueda aproximada del vecino más cercano de ultra alta dimensión para PostgreSQL,» de *Association for Computing Machinery*, Nueva York, NY, EE. UU, 2020.
- [79] H. P. S. B. y. C. P. Priyanshu Singh, «Python para el desarrollo web,» *International Journal of Computer Science and Mobile Computing*, vol. 11, n° 04, pp. 36-48, 2022.
- [80] A. W.-B. a. B. Eich, «JavaScript: the first 20 years,» *Association for Computing Machinery*, vol. 4, pp. 1-189, 06 2020.
- [81] T. T. C. N. J. V. N. N. M. M. Mateo Horridge, «WebProtégé: una plataforma web colaborativa para la edición de ontologías biomédicas,» *Bioinformatics (Oxford University Press)*, vol. 30, n° 16, pp. 2384-2385, 2014.

- [82] I. H. B. M. G. S. Z. W. Birte Glimm, «HermiT: Un razonador de OWL 2,» *Revista de razonamiento automatizado(Springer Países Bajos)*, vol. 53, nº 3, pp. 245-269, 2014.
- [83] M. A. C. B. Y. O. H. d. S. A. L. J. d. S. Gustavo Oliveira, «Aplicación de la gestión del conocimiento para dar soporte a una aplicación de chatbot de inteligencia artificial.,» *Proceedings of the European Conference on Knowledge Management (ECKM)*, vol. 25, pp. 582-590, 2023.
- [84] R. F. a. S. H. Senen, «Sistemas de gestión del conocimiento en la educación superior: un estudio exhaustivo mediante revisión sistemática de la literatura y análisis bibliométrico (2019-2023),» *The Eastasouth Management and Business (ESMB)*, vol. vol. 2, nº no. 02, pp. pp. 184- 201, 2024.
- [85] J. Y. S. K. S. J. S. I. Z. Syed Ahsan Ali Zaman, «Creación de conocimiento en las PYMES en la era de la Industria 4.0: un estudio comparativo de Pakistán y China.,» *Revista de la Economía del Conocimiento*, vol. 16, pp. 11520-11546, 2024.
- [86] T. S. V. S. N. Ranjini S Nadig, «La integración de las prácticas de gestión del conocimiento en la educación superior: de la innovación local al impacto global.,» *Revista de la Economía del Conocimiento*, 2026.
- [87] R. B. R. H. Donald Hislop, *Gestión del conocimiento en las organizaciones : una introducción crítica*, Oxford: Oxford University Press , 2018.
- [88] B. M. M. B. M. B. K. Fatemeh Serpush, «Sistema de aprendizaje profundo e inferencia ontológica difusa para el reconocimiento de actividades en personas mayores,» *Multimed Tools Appl*, vol. 85, nº 519, 2026.
- [89] W. L. J. L. S. C. Pandilla Xiao, «Razonamiento de cadena de pensamiento guiado por ontología para la construcción de grafos de conocimiento con

un modelo de lenguaje extenso.,» *Sistemas de Información Cooperativos*, vol. 15535, pp. 562-572, 2026.

- [90] A. A. S. A. A. P. Robert David, «OWL stric: : Un fragmento OWL restringido para evitar ambigüedades para los profesionales de los grafos de conocimiento,» *La web semántica (ESWC 2025)*, vol. 15719, pp. 47-64, 2025.
- [91] Y. G. A. J. A. N. A. P. J. T. Natasha Noy, «Grafos de conocimiento a escala industrial: lecciones y desafíos: cinco empresas tecnológicas diversas muestran cómo se hace,» *ACM Queue*, vol. 17, nº 2, pp. 48-75, 2019.
- [92] R. M. C. G. Sarah Ayad, «Un marco de selección de características híbrido basado en ontologías para mejorar la precisión predictiva en modelos de regresión,» *Knowl Inf Syst*, vol. 67, pp. 9111-9145, 2025.
- [93] S. M. C. O. H. E. K. hristine Lahoud, «Un sistema de recomendación híbrido basado en razonamiento de casos y ontología para la orientación de estudiantes en la educación superior,» *Investigación y desarrollo de tecnología educativa*, vol. 73, pp. 2494-2522, 2025.
- [94] L. X. J. A. Md Saiful Islam, «BERT: avances en la comprensión del lenguaje para diferentes tareas de PLN: desafíos y perspectivas futuras,» *Revista de Sistemas Eléctricos y Tecnología de la Información*, vol. 13, nº 9, 2026.
- [95] C. L. S. K. Aikaterini Karanikola, «Reducción de los requisitos de datos etiquetados en la clasificación de texto con aprendizaje activo y transformadores basados en BERT,» *Computación neuronal y aplicaciones*, vol. 38, nº 26, 2026.

- [96] E. C. Y. L. C. S. H. J. Eunbeen Jo, «Preentrenamiento adaptativo de dominio e idioma de modelos BERT para el análisis de textos clínicos bilingües coreano-inglés.,» *Informática médica y toma de decisiones de BMC*, vol. 25, nº 428, 2025.
- [97] Z. Z. Y. J. R. J. H. W. y. Fei Chen, «Investigación sobre el cálculo de similitud de textos cortos en el dominio de la terminología basada en el modelo BERT siamés.,» *Informes científicos*, vol. 15, nº 36954, 2025.
- [98] D. Y. Yu A. Malkov, «Búsqueda eficiente y robusta del vecino más cercano aproximado mediante grafos de mundo pequeño navegables jerárquicos.,» *EEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, nº 4, pp. 824-8, 2020.
- [99] Z. Z. Z. S. Z. W. C. H. Y. L. Y. Y. X. C. R. G. Ruqiang Yan, «Aprendizaje automático basado en el conocimiento para la obtención de pronósticos inteligentes interpretables y la gestión de la salud: revisión y estudio de caso.,» *Revista China de Ingeniería Mecánica*, vol. 38, nº 5, 2025.
- [100] P. C. R. K. Len Bass, *Arquitectura de software en la práctica*, 4.^a edición, Addison-Wesley Professional, 2021.
- [101] C. Maheshwari, «Del procesamiento por lotes al procesamiento en tiempo real: Creación de pipelines de inferencia en tiempo real para el aprendizaje automático.,» *Revista internacional de investigación científica en ciencias de la computación, ingeniería y tecnología de la información*, vol. 11, nº 1, pp. 3546-3555, 2025.
- [102] V. Sarcar, *Patrones de diseño creacionales en Java*, Prensa Berkeley, CA, 2026.

- [103] G. R. Fabio Caffaro, «Agentes conversacionales con conocimiento mejorado,» *Revista de Ciencias de la Computación y Tecnología*, vol. 39, pp. 585-609, 2024.
- [104] H. K. ., O. ., T. W.-J. v. d. ndika Kumara, «Modelos de lenguaje a gran escala para la computación orientada a servicios (LLM4SOC): Revisión y líneas de investigación,» de *Service-Oriented Computing. SummerSOC 2025* , Cham, Suiza, 2025.
- [105] A. Chahid, «Alineación de los marcos COBIT 5 e ITIL v4 mediante una ontología: Hacia una mayor madurez de los procesos de TI,» de 5.^a *Conferencia Internacional sobre Investigación Innovadora en Ciencia Aplicada, Ingeniería y Tecnología (IRASET)*, Fez, Marruecos, 2025.

ANEXOS

ESTUDIO DE SATISFACCIÓN DE LOS SERVICIOS TECNOLÓGICOS DEL DEPARTAMENTO DE NIVELACIÓN Y ADMISIÓN DE LA UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA

Objetivo: Levantar información de los diversos servicios tecnológicos del Departamento de Nivelación y Admisión de la Universidad Estatal Península de Santa Elena, lo que permitirá tomar decisiones para mejorarlos.

Entrevistado: Ing. Iván Coronel

Rol del Entrevistado: Director de Admisión y Nivelación

FECHA: 18/6/2025

- **Recursos Humanos**

¿Cuáles son sus responsabilidades como director del departamento de Nivelación y Admisión de la UPSE?

Rol del director:

Supervisar y garantizar el correcto desarrollo del proceso de admisión, así como su continuidad con el periodo de Nivelación, asegurando el cumplimiento efectivo de los directrices establecidas

¿Con cuántas personas cuenta actualmente el Departamento de Nivelación y Admisión?

1. Director
2. Analista informática
3. Analista administrativa

¿Podría indicar la cantidad de estudiantes y docentes que iniciaron en el proceso de nivelación y admisión durante los siguientes periodos académicos?

Periodo Académico	Estudiantes	Docentes
2025 - 1	1ra Matricula: 3530 2da Matricula: 58 Total: 3588	En curso, esperan el reporte final
2024 - 2	1ra Matricula: 3230 2da Matricula: 70 Total: 3302	127
2024 - 1	1ra Matricula: 4368 2da Matricula: 79 Total: 4447	132

¿Podría indicar la cantidad de estudiantes que aprobaron en el proceso de nivelación y admisión durante los siguientes periodos académicos?

Periodo Académico	Estudiantes
2025 - 1	En curso, esperan el reporte final
2024 - 2	2169
2024 - 1	3204

¿Cuáles considera usted han sido las razones por las cuales los estudiantes no aprobaron el curso de nivelación y admisión?

Principal razón por la que reprueban es la falta de disciplina en los estudiantes y de parte de los docentes la falta de compromiso con Nivelación.

Por otro lado, la falta de conocimiento del uso de herramientas virtuales por parte de los estudiantes que inician el curso.

- **Servicios Tecnológicos**

¿Cuáles son los principales servicios tecnológicos que ofrece actualmente el Departamento de Nivelación y Admisión?

- Página Web
- Redes Sociales: Facebook Página Oficial y Telegram Como Canal de Difusión
- Correo Institucional (Canal De Comunicación)
- Zoom (Inducción y Clases)
- Plataforma SGA
- Aula Virtual- Moodle

- **Problemas Frecuentes**

¿Cuáles son los inconvenientes más comunes que se presentan con los servicios tecnológicos?

- solicitan cambios de datos personales por que se olvidan de sus credenciales o las registran de manera correcta como: las contraseñas, registro de correo electrónico personal, número de celular
- no saben cómo usar el aula virtual (plataforma Moodle)

- **Medición de la Satisfacción**

No cuentan con un método con el cual puedan saber cuántos casos (incidentes) se atienden diaria o semanalmente para lograr evaluar la atención o los servicios brindados en el Departamento.

rdfs:label	
Usage: rdfs:label	
Show: ☒ this ☒ disjoints	
Found 40 uses of rdfs:label	
Asistencia Tecnica	'Asistencia Tecnica' rdfs:label "Asistencia Tecnica"
Asistente Virtual	'Asistente Virtual' rdfs:label "Asistente Virtual"
Aulas Virtuales	'Aulas Virtuales' rdfs:label "Aulas Virtuales"
Calificaciones	Calificaciones rdfs:label "Calificaciones"
Canales Digitales	'Canales Digitales' rdfs:label "Canales Digitales"
Catálogo	Catálogo rdfs:label "Catálogo"
Correo Institucional	'Correo Institucional' rdfs:label "Correo Institucional"
Email Institucional	'Email Institucional' rdfs:label "Email Institucional"@es

ANEXO 2. Rdfs:label - Etiqueta de definición del nombre por cada clase (categoría/servicio) del catálogo

skos:definition	
Usage: skos:definition	
Show: ☒ this ☒ disjoints	
Found 42 uses of skos:definition	
Asistencia Tecnica	'Asistencia Tecnica' skos:definition "Soporte presencial para fallas de hardware dispositivos fis...
Asistente Virtual	'Asistente Virtual' skos:definition "Chatbot conversacional para resolver preguntas sobre el funci...
Aulas Virtuales	'Aulas Virtuales' skos:definition "Acceso e ingreso a la plataforma educativa EVA/Moodle de la U...
Calificaciones	Calificaciones skos:definition "Visualizar, subir y editar las notas de estudiantes en Moodle/eva/...
Canales Digitales	'Canales Digitales' skos:definition "Difusión de comunicados oficiales por Facebook, Telegram,
Catálogo	Catálogo skos:definition "Clase abstracta raíz de la cual heredan las categorías de gestión que
Correo Institucional	

ANEXO 3. Skos:definition - Etiqueta de definición del concepto por cada clase (categoría/servicio) del catálogo

☰ skos:altlabel - [redacted]

Usage: skos:altlabel

Show: ☒ this ☒ disjoints

Found 84 uses of skos:altlabel

- **Aulas Virtuales**
 - 'Aulas Virtuales' skos:altlabel "mis credenciales de acceso al sistema moodle"
 - 'Aulas Virtuales' skos:altlabel "No me se la contraseña del aula virtual"
 - 'Aulas Virtuales' skos:altlabel "contraseña del aula para saber"
 - 'Aulas Virtuales' skos:altlabel "cual es la contraseña del moodle"
 - 'Aulas Virtuales' skos:altlabel "cual es mi contraseña del aula"
 - 'Aulas Virtuales' skos:altlabel "mi contraseña del sistema eva"
 - 'Aulas Virtuales' skos:altlabel "No me se la contraseña del eva"
 - 'Aulas Virtuales' skos:altlabel "cual es mi contraseña del sistema moodle"
 - 'Aulas Virtuales' skos:altlabel "contraseña del moode para ingresar"
 - 'Aulas Virtuales' skos:altlabel "contraseña del aula para ingresar"
 - 'Aulas Virtuales' skos:altlabel "mi contraseña del aula"
 - 'Aulas Virtuales' skos:altlabel "cual es la contraseña del aula"
 - 'Aulas Virtuales' skos:altlabel "contraseña del moodle para saber"
 - 'Aulas Virtuales' skos:altlabel "mis credenciales de acceso al aula"
 - 'Aulas Virtuales' skos:altlabel "mi contraseña del sistema moodle"
 - 'Aulas Virtuales' skos:altlabel "No me se la contraseña del moodle"

ANEXO 4. Etiqueta de definición de términos alternativos por cada clase (categoría/servicio) del catálogo

Description: Asistencia Tecnica

SubClass Of +

- 'Gestion de Soporte'

Description: Moodle

SubClass Of +

- 'Gestion de Sistemas Academicos'

Description: Aulas Virtuales

SubClass Of +

- 'Gestion de Accesos'

Description: Solicitud De Documentos

SubClass Of +

- 'Gestion de Comunicacion'

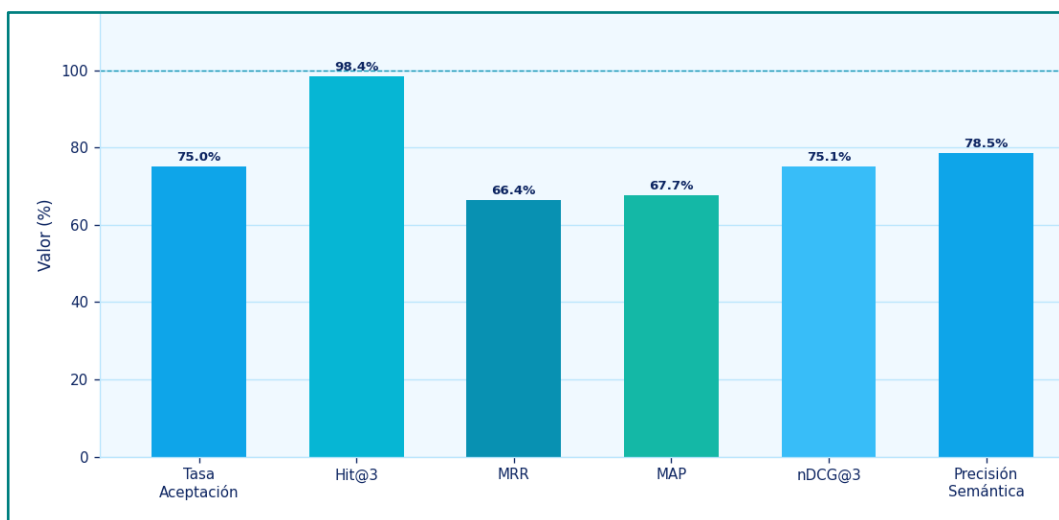
ANEXO 5. Propiedades lógicas de la ontología - "subclassof"

Usage: pertenece a categoría	
Show: ☒ this ☒ disjoints	
Canales Digitales	Canales Digitales' SubClassOf 'pertenece a categoría' some 'Gestion de Comunicacion'
Correo Institucional	Correo Institucional' SubClassOf 'pertenece a categoría' some 'Gestion de Accesos'
Email Institucional	Email Institucional' SubClassOf 'pertenece a categoría' some 'Gestion de Soporte'
Matriculacion	Matriculacion SubClassOf 'pertenece a categoría' some 'Gestion de Sistemas Academicos'

ANEXO 6. Propiedades lógicas de la ontología - "someValueFrom" y "pertenece a categoría"

Usage: skos:scopeNote	
Show: ☒ this ☒ disjoints	
Found 14 uses of skos:scopeNote	
Aulas Virtuales	'Aulas Virtuales' skos:scopeNote "No incluye el uso de herramientas dentro del EVA como tareas, foros o cuestionarios. Es
Correo Institucional	'Correo Institucional' skos:scopeNote "No incluye la configuración ni el uso interno del correo. Las funciones de Outlook, ca
Email Institucional	'Email Institucional' skos:scopeNote "No incluye problemas de acceso o login al correo. Las fallas de autenticación corres
Moodle	Moodle skos:scopeNote "No incluye problemas de acceso o login al EVA. Las fallas de autenticación corresponden al serv
Sistema de Gestion Academica	'Sistema de Gestion Academica' skos:scopeNote "No incluye problemas de acceso o login al sistema. Los fallos de auten

ANEXO 7. Propiedades lógicas de la ontología - "skos:scopeNote"



ANEXO 8. Resumen de métricas de desempeño del sistema

```

#
# MÉTODO PRINCIPAL
#
def validar(self, texto_usuario, servicio_bert, categoria_bert, confianza_bert=0.0,
id_interaccion=None):
    """
    Valida la predicción de BERT con 4 capas:
    Capa 1: Fuzzy en texto del usuario (n-gramas, mejor match)
    Capa 2: Semántico SBERT en texto del usuario (expresiones no listadas)
    Capa 3: Validación de la predicción BERT contra ontología
    Capa 4: Devolución del resultado BERT si todo falla
    Regla de protección: si BERT tiene alta confianza ( $\geq 85\%$ ) y la ontología
    solo encuentra coincidencia débil ( $< 90\%$ ), se respeta a BERT.
    La ontología SOLO sobreescrbe a BERT cuando:
    a) La confianza de BERT es baja ( $< 85\%$ ), O
    b) La ontología tiene coincidencia muy alta ( $\geq 90$ , término explícito)
    Parámetros:
    texto_usuario — texto original del usuario
    servicio_bert — servicio detectado por BERT
    categoria_bert — categoría detectada por BERT
    confianza_bert — confianza de BERT (0.0 a 1.0); protege a BERT si  $\geq 0.85$ 
    """
    servicio_final = servicio_bert
    categoria_final = categoria_bert
    info_servicio = None
    encontrado = False
    iri_concepto = None
    metodo = "bert_directo"

    # Capa 1: fuzzy n-grama en texto del usuario
    iri, score_fuzzy, texto_match, info = self._buscar_en_texto(texto_usuario)
    termino_ontologia = None # frase específica del OWL que hizo match

    # Capa 2: semántico SBERT en texto del usuario
    iri_sem, score_sem, texto_sem = self._buscar_semantico(
        self._normalizar(texto_usuario)
    )

    # Decidir cuál capa gana (la de mayor score, con preferencia al fuzzy
    # si ambas tienen score similar)
    usar_semantico = (
        iri_sem is not None and
        (iri is None or score_sem > score_fuzzy + 5) # SBERT gana si supera fuzzy por 5+ pts
    )
    if usar_semantico and iri_sem:
        info_sem = self._obtener_relaciones(iri_sem)
        if info_sem.get('perteneceACategoria'):
            iri, score_fuzzy, texto_match, info = iri_sem, score_sem, texto_sem, info_sem
            metodo = "semantico_sbert"

    # — Guardar clasificación independiente de la ontología (diagnóstico)
    onto_sugiere_categoria = None
    if iri and info and info.get('perteneceACategoria'):

```

```

if info.get('labels'):
    onto_sugiere_servicio = info['labels'][0]
if info.get('perteneceACategoriaLabel'):
    onto_sugiere_categoria = info['perteneceACategoriaLabel'][0]

# Regla de protección de BERT
# La ontología SOLO corrige cuando BERT tiene baja confianza (< 60%).
# Si BERT es confiable se respeta su clasificación pero igual se guarda
# lo que la ontología habría clasificado para fines de evaluación.
bert_confiable = confianza_bert >= 0.60
# Precisión real del match de la ontología, calculada antes de aplicar regla BERT
if score_fuzzy >= 70:
    precision_onto = round(score_fuzzy / 100, 2)
elif score_sem >= 55:
    precision_onto = round(score_sem / 100, 2)
else:
    precision_onto = 0.0

if bert_confiable and precision_onto < 0.90:
    print(f" [ONTOLOGIA] BERT confiable ({confianza_bert:.0%}), match ontología
({precision_onto:.0%}) insuficiente → respetando BERT")
    if onto_sugiere_servicio:
        print(f" [ONTOLOGIA] (diagnóstico) ontología habría clasificado:
{onto_sugiere_servicio} | {onto_sugiere_categoria}")
        iri, info = None, None # no anular BERT
    elif bert_confiable and precision_onto >= 0.90:
        print(f" [ONTOLOGIA] BERT confiable ({confianza_bert:.0%}) pero match ontología
fuerte ({precision_onto:.0%}) → ontología corrige")

if iri and info and info.get('perteneceACategoria'):
    encontrado = True
    info_servicio = info
    iri_concepto = str(iri)
    termino_ontologia = texto_match # frase del OWL que hizo match
    if info.get('labels'):
        servicio_final = info['labels'][0]
    if info.get('perteneceACategoriaLabel'):
        categoria_final = info['perteneceACategoriaLabel'][0]
    metodo = metodo if metodo == "semantico_sbert" else "fuzzy_texto"
    print(f" [ONTOLOGIA] [{metodo}] '{texto_match}' → {servicio_final} |
{categoria_final} (score={score_fuzzy:.0f})")
else:

# Capa 3: validar predicción BERT
iri_bert, _, _ = self._buscar_exhaustivo(servicio_bert)
if iri_bert:
    info_servicio = self._obtener_relaciones(iri_bert)
    iri_concepto = str(iri_bert)
    if info_servicio.get('labels'):
        servicio_final = info_servicio['labels'][0]
    if info_servicio.get('perteneceACategoriaLabel'):
        categoria_final = info_servicio['perteneceACategoriaLabel'][0]
    metodo = "validacion_bert"

```

```

termino_ontologia = None # Capa 3 no matchea un altLabel, solo valida estructura
OWL

# Capa 4: si todo falla, queda el resultado BERT directo
# Precisión de validación — basada en calidad real del match de la ontología
def _sin_tildes(t):
    return unicodedata.normalize('NFKD', t).encode('ascii', 'ignore').decode('ascii').lower()

if score_fuzzy >= 70:
    precision = round(score_fuzzy / 100, 2)
    tipo = f"Identificacion directa [{metodo}]"
elif score_sem >= 55:
    precision = round(score_sem / 100, 2)
    tipo = f"Identificacion semantica [{metodo}]"
elif encontrado and info_servicio and info_servicio.get('perteneceACategoriaLabel'):
    precision = 1.0
    tipo = f"Identificacion directa [{metodo}]"
else:
    s_coincide = _sin_tildes(servicio_bert) == _sin_tildes(servicio_final)
    c_coincide = _sin_tildes(categoria_bert) == _sin_tildes(categoria_final)
    if not s_coincide or not c_coincide:
        precision = 0.9 if info_servicio else 0.7
        tipo = "Correccion semantica mediante estructura ontologica"
    else:
        precision = 0.6 if (info_servicio and info_servicio.get('definitions')) else 0.3
        tipo = "Confirmacion con validacion semantica" if precision > 0.3 else "Confirmacion
simple"

id_subcategoria = self._buscar_id_subcategoria(servicio_final)
validado_ontologia = id_subcategoria is not None

if id_subcategoria is None:
    accion_ontologia = 'sin_resultado'
elif _sin_tildes(servicio_final) != _sin_tildes(servicio_bert):
    accion_ontologia = 'corrigio'
elif _sin_tildes(categoria_final) != _sin_tildes(categoria_bert):
    accion_ontologia = 'complemento'
else:
    accion_ontologia = 'valido'

```

ANEXO 9. Capas de Validación ontológica

```

# =====
# SECCIÓN 1 — ASISTENTE VIRTUAL Y ZAMMAD PARA GENERAR TICKET
# =====
import os
import re
import unicodedata
import requests
from requests.exceptions import ConnectionError, Timeout, HTTPError
from fastapi import APIRouter
from psychopg2.extras import RealDictCursor

```

```

from api.config import ZAMMAD_URL, ZAMMAD_TOKEN, conectar_db,
UMBRAL_SIMILITUD
from api.services.embeddings import (
    generar_embeddings_triple, embedding_to_pgvector,
    guardar_embeddings
)

class ZammadClient:
    # Conexión a la API REST de Zammad: URL base + token de autenticación.
    # Todas las llamadas HTTP (requests.get / requests.post) usan estos valores.
    def __init__(self):
        self._base = ZAMMAD_URL.rstrip('/')
        self._headers = {
            'Authorization': f'Token token={ZAMMAD_TOKEN}',
            'Content-Type': 'application/json',
        }

    def buscar_por_cedula(self, cedula: str) -> dict | None:
        #Busca un usuario en Zammad por cédula.
        #La cédula debe estar almacenada en el campo 'fax' del usuario.
        #Devuelve el dict del usuario o None si no existe.
        #Usa conexión directa a la BD porque la API REST de Zammad no permite
        #buscar Customers con un token de Agent.
        cedula = (cedula or "").strip()
        if not cedula:
            return None

        conn = None
        try:
            # conectar_db() abre psycopg2 directo al PostgreSQL compartido con
            # Zammad (base zammad_production, puerto 5440) y consulta su tabla 'users'.
            conn = conectar_db()
            cur = conn.cursor()
            cur.execute("""
                SELECT u.id,
                       u.firstname,
                       u.lastname,
                       u.email,
                       u.organization_id,
                       (SELECT gu.group_id
                        FROM groups_users gu
                        WHERE gu.user_id = u.id
                        LIMIT 1) AS id_grupo
                FROM users u
                WHERE u.fax = %s
                LIMIT 1
            """, (cedula,))
            row = cur.fetchone()
            if not row:
                return None
            return {
                'id_usuario': row[0],
                'nombre': row[1] or "",
            }
        except:
            if conn:
                conn.close()
            return None

```

```

        'apellido': row[2] or "",
        'email': row[3] or "",
        'id_organizacion': row[4],
        'id_grupo': row[5],
    }
    finally:
        if conn:
            conn.close()

def obtener_usuario_por_id(self, id_usuario: int) -> dict | None:
    """Devuelve nombre y email de un usuario dado su ID."""
    resp = requests.get(
        f'{self._base}/api/v1/users/{id_usuario}',
        headers=self._headers,
        timeout=10,
    )
    if resp.status_code == 404:
        return None
    resp.raise_for_status()
    u = resp.json()
    return {
        'nombre': u.get('firstname', ""),
        'email': u.get('email', ""),
    }

# crea el ticket con título, descripción y estado
def crear_ticket(
    self,
    titulo: str,
    id_grupo: int,
    id_cliente: int,
    id_organizacion: int,
    id_dueno_ticket: int,
    descripcion: str,
    servicio: str,
    categoria: str,
    estado: str | None = None,
    solucion: str | None = None,
) -> dict:
    payload = {
        'title': titulo,
        'group_id': id_grupo,
        'customer_id': id_cliente,
        'organization_id': id_organizacion,
        'owner_id': id_dueno_ticket,
        'article': {
            'type': 'web',
            'customer_id': id_cliente,
            'subject': titulo,
            'body': descripcion,
            'content_type': 'text/plain',
            'internal': False,
        },
    },

```

```

    }

    if estado and estado.lower() in ('cerrado', 'closed'):
        payload['state'] = 'closed'

    try:
        resp = requests.post(
            f'{self._base}/api/v1/tickets',
            json=payload,
            headers=self._headers,
            timeout=10,
        )
        resp.raise_for_status()
    except ConnectionError:
        raise RuntimeError(f'No se pudo conectar con Zammad en {self._base}')
    except Timeout:
        raise RuntimeError("Zammad no respondió a tiempo (timeout 10s)")
    except HTTPError:
        raise RuntimeError(f'Zammad devolvió error {resp.status_code}: {resp.text}')

    ticket = resp.json()

    if solucion and solucion.strip():
        self._agregar_articulo_solucion(ticket['id'], id_cliente, solucion)

    return ticket

```

ANEXO 10. Integración entre asistente virtual y Zammad para crear ticket

```

#=====
# zammad_legacy.py, puente traductor entre el Modal Zammad (Json) y la API FastAPI
# =====
# Flujo completo:
#   Agente cierra ticket en Zammad (interfaz web)
#   → Modal detecta el cierre y envía petición HTTP
#   → zammad_legacy.py recibe la petición (rutas Node.js style)
#   → llama a las funciones ep_* de zammad.py
#   → devuelve respuesta envuelta en {error, data}
#   → el modal la lee y muestra opciones al agente
#=====
from fastapi import APIRouter, Query
from psycpg2.extras import RealDictCursor
from api.config import conectar_db
from api.services.zammad import (
    ep_validar, ep_guardar, ep_reemplazar,
    ep_obtener_archivos, ep_buscar_por_ticket, ep_calcular_similitud,
    ep_agregar_altlabels,
)
router = APIRouter(prefix="/api", tags=["Zammad-Legacy"])

# Tabla de despacho: el modal envía { args: { accion: "validar", ... } }
# y este diccionario decide qué función de zammad.py ejecutar según la acción.

```

```

_DISPATCH = {
    'validar':          ep_validar,          # compara semánticamente la solución nueva
    'guardar':          ep_guardar,          # guarda solución nueva en la BD
    'reemplazar':       ep_reemplazar,       # actualiza solución existente
    'obtener_archivos': ep_obtener_archivos, # devuelve archivos adjuntos de una solución
    'buscar_por_ticket': ep_buscar_por_ticket, # verifica si el ticket ya tiene solución
guardada
    'calcular_similitud_directa': ep_calcular_similitud, # calcula % de similitud puntual
    'agregar_altlabels': ep_agregar_altlabels, # agrega términos alternativos a la ontología
    OWL
}

# Endpoint 1: principal
# El modal siempre llama a esta única ruta para todas las acciones.
# Lee el campo "accion" de la petición enviada, busca la función en _DISPATCH y la ejecuta.
@router.post("/python/ejecutar-script")
def ejecutar_script(payload: dict):
    args = payload.get('args', {}) or {}
    accion = args.get('accion')
    handler = _DISPATCH.get(accion)
    if not handler:
        return {'error': True, 'mensaje': f'Accion no soportada: {accion}', 'data': None}
    datos = {k: v for k, v in args.items() if k != 'accion'}
    try:
        resultado = handler(datos)
    except Exception as e:
        return {'error': True, 'mensaje': str(e), 'data': None}

    if isinstance(resultado, dict) and resultado.get('error'):
        return {'error': True, 'mensaje': resultado.get('mensaje', 'Error'), 'data': resultado}
    return {'error': False, 'data': resultado}

# Endpoint 2: Categoría por ticket
# Busca en interaccion_chatbot el servicio y categoría que el bot clasificó
# para ese ticket. El modal lo usa para preseleccionar la categoría correcta
# cuando el agente va a guardar la solución.
@router.get("/python/categoria-por-ticket")
def categoria_por_ticket(ticket_id: str = Query(...)):
    conn = None
    try:
        conn = conectar_db()
        cur = conn.cursor(cursor_factory=RealDictCursor)
        cur.execute("""
            SELECT s.id_subcategoria, s.nombre_servicio AS subcategoria_nombre,
                   c.nombre AS categoria_nombre
            FROM interaccion_chatbot i
            JOIN c_subcategoria s ON s.id_subcategoria = i.id_subcategoria
            JOIN categoria      c ON c.id_categoria   = s.id_categoria
            WHERE i.ticket_zammad_id = %s
            ORDER BY i.creado_en DESC
            LIMIT 1
            """, (ticket_id,))
        row = cur.fetchone()
    except:
        if not row:

```

```

        return {'error': False, 'data': None}
    return {'error': False, 'data': dict(row)}
except Exception as e:
    return {'error': True, 'mensaje': str(e), 'data': None}
finally:
    if conn: conn.close()

```

Endpoint 3: Subcategoría por nombre

Busca el ID de una subcategoría dado su nombre exacto.

El modal lo usa para obtener el id_subcategoría antes de guardar una solución.

@router.get("/python/subcategoría-por-nombre")

def subcategoría_por_nombre(nombre: str = Query(...)):

```

    conn = None
    try:
        conn = conectar_db()
        cur = conn.cursor(cursor_factory=RealDictCursor)
        cur.execute("""
            SELECT id_subcategoría, nombre_servicio AS nombre, id_categoría
            FROM c_subcategoría
            WHERE LOWER(nombre_servicio) = LOWER(%s) AND activo = TRUE
            LIMIT 1
            """, (nombre.strip(),))
        row = cur.fetchone()
        if not row:
            return {'error': False, 'data': None}
        return {'error': False, 'data': dict(row)}
    except Exception as e:
        return {'error': True, 'mensaje': str(e), 'data': None}
    finally:
        if conn: conn.close()

```

Endpoint 4: Listar categorías

Devuelve todas las categorías activas de la BD.

El modal lo usa para poblar el menú desplegable de categorías.

@router.get("/categorias")

def listar_categorías():

```

    conn = None
    try:
        conn = conectar_db()
        cur = conn.cursor(cursor_factory=RealDictCursor)
        cur.execute("""
            SELECT id_categoría, nombre, descripción
            FROM categoría
            WHERE activo = TRUE
            ORDER BY nombre
            """)
        return {'error': False, 'data': [dict(r) for r in cur.fetchall()]}
    except Exception as e:
        return {'error': True, 'mensaje': str(e), 'data': []}
    finally:
        if conn: conn.close()

```

```

# Endpoint 5: Listar subcategorías de una categoría
# Devuelve los servicios (subcategorías) activos que pertenecen a una categoría.
# El modal lo usa para poblar el segundo menú desplegable al seleccionar una categoría.
@router.get("/categorias/{id_categoria}/subcategorias")
def listar_subcategorias(id_categoria: int):
    conn = None
    try:
        conn = conectar_db()
        cur = conn.cursor(cursor_factory=RealDictCursor)
        cur.execute("""
            SELECT id_subcategoria, nombre_servicio AS nombre, descripcion
            FROM c_subcategoria
            WHERE id_categoria = %s AND activo = TRUE
            ORDER BY nombre_servicio
            """, (id_categoria,))
        return {'error': False, 'data': [dict(r) for r in cur.fetchall()]}
    except Exception as e:
        return {'error': True, 'mensaje': str(e), 'data': []}
    finally:
        if conn: conn.close()

```

ANEXO 11. Puente traductor entre el Modal de Zammad (Json) y la API FastAPI