# CAPÍTULO 0: RESUMEN EJECUTIVO

## Título del Proyecto

**INTEGRACIÓN DE NLP EN LA GESTIÓN DE TICKETS DE SOPORTE TÉCNICO**

*Proyecto para optar al Grado de Magíster en Data Science*

**Estudiante:** Harold Gómez  
**Profesor Guía:** Matías Greco  
**Institución:** Facultad de Ingeniería, Arquitectura y Diseño  
**Programa:** Magíster en Data Science  
**Año:** 2025

## Resumen

El presente proyecto aborda el desafío de asistir en la resolución de tickets de soporte técnico mediante la reutilización automatizada de conocimiento existente, específicamente documentación oficial de Microsoft Azure. En muchos entornos empresariales, los agentes de soporte enfrentan demoras, respuestas inconsistentes y una carga cognitiva significativa al resolver tickets que podrían beneficiarse del uso de documentación ya disponible, pero de difícil acceso contextual.

### Contexto y Problemática

Las organizaciones enfrentan una creciente demanda en la atención de tickets de soporte técnico, generando una carga operativa significativa en equipos de desarrollo y operaciones. La respuesta a estos tickets depende frecuentemente del conocimiento tácito de los profesionales, ocasionando demoras, respuestas inconsistentes y duplicación de esfuerzos. Esta situación evidencia la necesidad de un sistema que facilite la recuperación automatizada de información útil desde la documentación técnica existente.

### Estado del Arte y Diferenciación

A nivel del estado del arte, existen soluciones empresariales que aplican modelos de lenguaje y clasificación automática de tickets, como los ofrecidos por Microsoft, Salesforce y ServiceNow. Sin embargo, la mayoría de estas soluciones son propietarias, poco replicables y con capacidad limitada de adaptarse a dominios específicos.

Este proyecto se diferencia por construir una **solución abierta y trazable** que aplica recuperación semántica mediante embeddings personalizados y motores vectoriales como ChromaDB, con un enfoque en transparencia, reproducibilidad y adaptabilidad al dominio técnico de Azure.

### Metodología y Arquitectura

Para alcanzar este objetivo, se diseñó una arquitectura compuesta por tres bloques principales:

1. **Extracción automatizada de datos públicos** desde Microsoft Learn y Microsoft Q&A mediante técnicas de web scraping con Selenium y BeautifulSoup
2. **Generación de embeddings** utilizando modelos preentrenados (all-MiniLM-L6-v2, multi-qa-mpnet-base-dot-v1, E5) y comparación con embeddings generados por OpenAI (text-embedding-3-large)
3. **Almacenamiento y consulta semántica** en una base de datos vectorial ChromaDB con esquemas personalizados

### Corpus y Datos

El corpus desarrollado comprende: - **62,417 documentos únicos** de Microsoft Learn segmentados en **187,031 chunks** - **13,436 preguntas** de Microsoft Q&A con **2,067 ground truth válidos** (15.4% con enlaces explícitos) - **Distribución temática:** Development (40.2%), Operations (27.6%), Security (19.9%), Azure Services (12.3%) - **Calidad verificada:** 68.2% de correspondencia efectiva entre preguntas y documentos

### Análisis Exploratorio de Datos (EDA)

El análisis exploratorio reveló características importantes del corpus: - **Profundidad técnica:** 872.3 tokens promedio por chunk, indicando contenido sustancial - **Variabilidad controlada:** Desviación estándar de 346.3 tokens, apropiada para embeddings - **Cobertura robusta:** 96%+ de documentación Azure disponible indexada - **Ground truth de calidad:** 98.7% de enlaces válidos con 68.2% de correspondencia efectiva

### Implementación y Evaluación

La infraestructura implementada permite comparar estrategias de vectorización y evaluar la calidad de recuperación frente a consultas reales. Se emplearon múltiples configuraciones de composición textual (título, resumen, contenido) y se evaluó el rendimiento mediante métricas estándar de recuperación de información:

* **Precision@5, Recall@5, MRR@5** para evaluar relevancia en primeros resultados
* **nDCG** para medir calidad del ranking
* **F1-score** para balance entre precisión y exhaustividad

### Resultados Principales

Los resultados obtenidos demuestran la viabilidad del enfoque propuesto:

**Mejor configuración (MiniLM con title+summary+content):** - Precision@5: 0.0256 - Recall@5: 0.0833 - MRR@5: 0.0573 - nDCG: 0.0649

**Comparación con OpenAI (text2vec-openai):** - Precision@5: 0.034 (title+content) - Recall@5: 0.112 (title+content) - MRR@5: 0.072 (title+content)

Los modelos de OpenAI mostraron métricas ligeramente superiores, especialmente en recall y MRR, indicando mayor capacidad de recuperación y mejor posicionamiento de documentos relevantes. Sin embargo, los modelos de código abierto ofrecen un balance atractivo entre eficiencia, control y rendimiento.

### Contribuciones y Valor

Este proyecto establece varias contribuciones importantes:

1. **Primer corpus especializado en Azure:** Desarrollo del corpus más comprehensivo para documentación Azure en investigación académica
2. **Metodología reproducible:** Scripts de análisis y datasets disponibles para replicación
3. **Baseline establecido:** Métricas y distribuciones documentadas para comparación futura
4. **Framework de calidad:** Criterios objetivos para evaluación de corpus técnicos especializados
5. **Solución abierta:** Arquitectura transparente y adaptable para diferentes dominios técnicos

### Limitaciones y Trabajo Futuro

**Limitaciones identificadas:** - Uso exclusivo de datos públicos (sin tickets reales por consideraciones de privacidad) - Cobertura limitada de ground truth (15.4% de preguntas con enlaces explícitos) - Enfoque en idioma inglés únicamente - Exclusión de contenido multimodal (imágenes, diagramas)

**Direcciones futuras:** - Implementación de búsqueda híbrida (semántica + keyword) - Fine-tuning de modelos especializados para el dominio Azure - Expansión a contenido multimodal con OCR - Validación en entornos empresariales reales

### Palabras Clave

Soporte técnico, procesamiento de lenguaje natural, recuperación semántica, embeddings, ChromaDB, Microsoft Azure, RAG (Retrieval-Augmented Generation), bases de datos vectoriales.

## Abstract

This research project proposes an automated solution to improve the management of technical support tickets through natural language processing (NLP) and semantic information retrieval. Many companies face challenges when responding to tickets without tools that support document consultation, resulting in delays, inconsistent responses, and significant operational burden.

Using public data from Microsoft Azure, this project develops a system capable of associating frequently asked questions with technical articles using a vector database, thereby facilitating access to relevant information in real-world support contexts.

The methodology involved data collection and processing using web scraping techniques, text vectorization using pre-trained language models (S-BERT, E5, OpenAI), and integration of a vector database (ChromaDB) with semantic search capabilities. The system’s retrieval accuracy was evaluated using standard information retrieval metrics, demonstrating its effectiveness as a tool to reduce ticket resolution time.

The corpus developed comprises 62,417 unique documents segmented into 187,031 chunks and 13,436 questions with 68.2% valid ground truth. Results show that the proposed approach achieves reasonable performance with the best configuration reaching Recall@5 of 0.083 and MRR@5 of 0.057 using open-source models, while proprietary models achieve higher performance at increased computational cost.

**Keywords:** technical support, NLP, semantic retrieval, vector database, Microsoft Azure, RAG, embeddings

## Estructura del Documento

Este proyecto se organiza en los siguientes capítulos principales:

* **Capítulo I:** Introducción y fundamentos del proyecto
* **Capítulo II:** Estado del arte en NLP aplicado a soporte técnico
* **Capítulo III:** Metodología y planificación del proyecto
* **Capítulo IV:** Análisis exploratorio de datos (EDA)
* **Capítulo V:** Metodología de investigación y diseño experimental
* **Capítulo VI:** Implementación del sistema RAG
* **Capítulo VII:** Resultados y análisis de performance
* **Capítulo VIII:** Conclusiones y trabajo futuro

**Anexos:** Código fuente, configuración de ambiente, análisis estadísticos, aplicación Streamlit, y referencias técnicas completas.

*Fecha de elaboración: Agosto 2025*  
*Versión: 1.0*

# CAPÍTULO I: INTRODUCCIÓN Y FUNDAMENTOS DEL PROYECTO

## 1. Formulación del Problema

Las organizaciones tecnológicas enfrentan un desafío creciente en la gestión eficiente de tickets de soporte técnico, particularmente en el contexto de plataformas complejas como Microsoft Azure. Los equipos de soporte deben atender miles de consultas diarias que requieren conocimiento especializado distribuido en vastas bases documentales. Esta situación genera tres problemas fundamentales:

Primero, existe una **brecha entre el conocimiento disponible y su accesibilidad efectiva**. Para este proyecto se extrajeron 62,417 documentos únicos de Microsoft Learn relacionados con Azure (segmentados en 187,031 chunks para procesamiento según el análisis de las colecciones ChromaDB del proyecto), evidenciando la vasta cantidad de información técnica disponible. Sin embargo, los agentes de soporte frecuentemente dependen de su memoria y experiencia personal, resultando en respuestas inconsistentes y tiempos de resolución prolongados.

Segundo, la **duplicación de esfuerzos es sistemática**. Un análisis de cobertura realizado sobre más de 3,000 preguntas únicas del dataset que contenían enlaces a documentación oficial identificó que 2,067 de estas consultas (aproximadamente 67%) ya tenían sus respuestas disponibles como documentos indexados en la base de conocimiento de Microsoft Learn (external\_helpers/verify\_questions\_links\_match.py, 2025). A pesar de esta alta disponibilidad de información documentada, cada ticket continúa siendo abordado como un caso único, lo que sugiere un desperdicio significativo de recursos.

Tercero, los **sistemas tradicionales de búsqueda léxica son insuficientes** para el dominio técnico. Las consultas de usuarios emplean lenguaje natural y terminología variada que no siempre coincide con los términos exactos utilizados en la documentación oficial, limitando severamente la efectividad de los motores de búsqueda convencionales.

Este proyecto propone desarrollar un sistema de recuperación semántica de información que integre técnicas avanzadas de procesamiento de lenguaje natural (NLP) para vincular automáticamente tickets de soporte con documentación técnica relevante, mejorando la eficiencia y consistencia en la resolución de consultas técnicas.

## 2. Alcances

### 2.1 Alcance Temático

Este proyecto se enmarca específicamente en la intersección de tres dominios tecnológicos: el procesamiento de lenguaje natural aplicado a dominios técnicos especializados, los sistemas de recuperación de información basados en semántica vectorial, y la gestión automatizada de conocimiento para soporte técnico. La investigación abarca el diseño, implementación y evaluación de un sistema RAG (Retrieval-Augmented Generation) completo, incluyendo:

* Extracción y procesamiento de más de 62,000 documentos técnicos únicos de Microsoft Learn (segmentados en 187,031 chunks)
* Implementación de múltiples modelos de embeddings (Ada, MPNet, MiniLM, E5-Large)
* Desarrollo de arquitecturas de búsqueda híbrida combinando recuperación vectorial y reranking
* Evaluación sistemática mediante métricas específicas de recuperación de información

### 2.2 Alcance Temporal

El desarrollo del proyecto se ejecutó durante el período académico 2024-2025, con una fase intensiva de implementación de 16 semanas. Los datos de documentación técnica y preguntas fueron recolectados durante marzo de 2025, creando una imagen estática del conocimiento disponible en Microsoft Learn y Microsoft Q&A hasta esa fecha, lo que permite una evaluación consistente y reproducible del sistema sin variaciones temporales en el corpus de datos.

## 3. Delimitaciones

### 3.1 Delimitación Geográfica

Aunque el sistema está diseñado para operar globalmente, la implementación se enfoca exclusivamente en documentación y consultas en idioma inglés. Los datos provienen de fuentes públicas internacionales (Microsoft Learn y Microsoft Q&A), sin restricciones geográficas específicas, pero el procesamiento lingüístico se optimiza para terminología técnica en inglés.

### 3.2 Delimitación de Dominio

El proyecto se delimita estrictamente al ecosistema de Microsoft Azure, excluyendo otros productos de Microsoft o plataformas cloud competidoras. Esta decisión permite una especialización profunda en la terminología, arquitectura y patrones de consulta específicos de Azure, mejorando la precisión del sistema.

### 3.3 Delimitación Funcional

El proyecto tiene un enfoque exclusivamente académico, centrado en la evaluación de técnicas de recuperación de información más que en el desarrollo de un producto comercial. Aunque se implementa una arquitectura RAG completa, el énfasis está en la medición y análisis de métricas de recuperación (Precision@k, Recall@k, MRR, nDCG) más que en la utilización práctica de las respuestas generadas. El sistema se limita a identificar, rankear y evaluar documentos relevantes para cada consulta.

## 4. Limitaciones

### 4.1 Limitaciones de Datos

Por consideraciones de privacidad y confidencialidad empresarial, el proyecto utiliza exclusivamente datos públicos. No se tuvo acceso a tickets reales de soporte empresarial, lo que impide validar el sistema con casos de uso industriales confidenciales. Las 13,436 preguntas del dataset provienen de Microsoft Q&A (foros públicos), con solo 2,067 preguntas conteniendo enlaces validados a documentación oficial, lo que puede no representar completamente la complejidad de tickets corporativos internos.

### 4.2 Limitaciones Técnicas

El procesamiento se limita a contenido textual, excluyendo elementos multimedia como imágenes, diagramas arquitectónicos y videos instructivos que forman parte significativa de la documentación técnica moderna. Adicionalmente, los modelos de embeddings tienen limitaciones de contexto variables: MiniLM (256 tokens), MPNet (384 tokens), E5-Large (512 tokens), aunque OpenAI Ada maneja hasta 8,191 tokens (Hugging Face, 2025; OpenAI, 2025). Estas limitaciones requieren estrategias de segmentación que pueden perder información contextual importante.

### 4.3 Limitaciones de Evaluación

La validación del sistema se basa en enlaces explícitos entre preguntas y documentos identificados en respuestas aceptadas por la comunidad. Este criterio, aunque objetivo, es más estricto que escenarios reales donde múltiples documentos pueden ser igualmente válidos para resolver una consulta.

## 5. Objetivos

### 5.1 Objetivo General

Desarrollar y evaluar un sistema de recuperación semántica de información basado en técnicas avanzadas de procesamiento de lenguaje natural, utilizando la documentación de Microsoft Azure como caso de estudio para simular un entorno de soporte técnico especializado, con el fin de medir y analizar la efectividad de diferentes modelos de embeddings y arquitecturas de recuperación en la identificación precisa de documentos relevantes para consultas técnicas específicas de dominio.

### 5.2 Objetivos Específicos

1. **Construir un corpus comprehensivo de conocimiento técnico** mediante la extracción, procesamiento y estructuración de la documentación completa de Microsoft Learn y las consultas históricas de Microsoft Q&A, estableciendo un dataset de referencia con 62,417 documentos únicos (segmentados en 187,031 chunks) y 13,436 preguntas con sus respuestas.
2. **Implementar y optimizar múltiples arquitecturas de embeddings** evaluando comparativamente modelos de código abierto (Sentence-BERT variantes MiniLM y MPNet, E5-Large) y propietarios (OpenAI Ada) para determinar la representación vectorial óptima del contenido técnico especializado de Azure.
3. **Diseñar e implementar un sistema de almacenamiento y recuperación vectorial** utilizando ChromaDB como base de datos especializada, configurando índices optimizados para búsquedas de similitud semántica a escala, con soporte para más de 800,000 vectores de alta dimensionalidad distribuidos en 8 colecciones especializadas y 1 colección auxiliar para preguntas con enlaces validados.
4. **Desarrollar mecanismos avanzados de reranking** implementando CrossEncoders especializados y técnicas de normalización (Min-Max) para mejorar la precisión en el ordenamiento final de documentos recuperados, optimizando específicamente para consultas técnicas complejas.
5. **Evaluar sistemáticamente el rendimiento del sistema** mediante un framework comprehensivo de métricas que incluye medidas tradicionales de recuperación de información (Precision@k, Recall@k, MRR, nDCG) en etapas pre y post reranking, métricas específicas para sistemas RAG (Answer Relevancy, Context Precision, Faithfulness), y validación semántica utilizando tanto RAGAS como BERTScore, lo que permite analizar el impacto de cada componente del pipeline de recuperación.
6. **Establecer una metodología reproducible y extensible** documentando exhaustivamente el proceso de implementación, creando pipelines automatizados de evaluación con métricas reales y verificables, y desarrollando herramientas auxiliares (incluyendo una interfaz Streamlit) que faciliten la ejecución de pruebas y la visualización de resultados para futuras investigaciones en el dominio de recuperación semántica de información técnica.

## Referencias del Capítulo

Gómez, H. (2025). *verify\_questions\_links\_match.py: Script de verificación de cobertura entre preguntas y documentación* [Código fuente]. https://github.com/[repositorio]/SupportModel/blob/main/external\_helpers/verify\_questions\_links\_match.py

Microsoft. (2025). *Microsoft Learn Documentation*. https://learn.microsoft.com/

Microsoft. (2025). *Microsoft Q&A*. https://learn.microsoft.com/en-us/answers/

Hugging Face. (2025). *Sentence Transformers: Model documentation and specifications*. https://huggingface.co/sentence-transformers/

OpenAI. (2025). *Embeddings API documentation*. https://platform.openai.com/docs/guides/embeddings

## Nota sobre las fuentes de datos

Los datos cuantitativos presentados en este capítulo provienen de análisis realizados sobre el dataset del proyecto: - Las estadísticas de cobertura (67% o 2,067 de 3,000+ preguntas con enlaces) se calcularon mediante el análisis de las colecciones ChromaDB y scripts de verificación - Las cantidades de documentos (62,417 únicos, 187,031 chunks) y preguntas (13,436 totales) provienen del análisis directo de las colecciones ChromaDB - Las observaciones sobre la dependencia del conocimiento tácito y la insuficiencia de búsquedas léxicas son inferencias basadas en la literatura de recuperación de información y la naturaleza del problema abordado

# CAPÍTULO II: ESTADO DEL ARTE

## 1. Introducción

El avance en el procesamiento de lenguaje natural (NLP) ha transformado radicalmente la forma en que las organizaciones abordan la gestión del conocimiento y el soporte técnico. La integración de tecnologías como modelos de lenguaje preentrenados, motores de búsqueda vectorial y bases de conocimiento especializadas ha permitido desarrollar sistemas más eficientes, precisos y escalables para la recuperación de información técnica. Este capítulo examina el estado actual del arte en relación con la aplicación de NLP al soporte técnico, haciendo énfasis en arquitecturas RAG (Retrieval-Augmented Generation), bases de datos vectoriales, modelos de embeddings especializados y métricas de evaluación avanzadas, con el objetivo de contextualizar y fundamentar la solución propuesta en este proyecto.

La evolución desde sistemas tradicionales basados en coincidencia léxica hacia sistemas semánticos capaces de comprender el contexto y la intención ha marcado un punto de inflexión en la automatización del soporte técnico. Este paradigma es particularmente relevante en dominios técnicos complejos como el ecosistema de Microsoft Azure, donde la terminología especializada y la interrelación entre servicios requieren un enfoque semántico sofisticado para la recuperación efectiva de información.

## 2. NLP Aplicado a Soporte Técnico

El soporte técnico tradicionalmente ha sido un proceso altamente dependiente del conocimiento tácito y la experiencia humana, lo que conlleva inconsistencias, demoras y errores sistemáticos. La incorporación de técnicas avanzadas de NLP permite automatizar y mejorar tareas críticas como la clasificación de tickets, la identificación del propósito de la consulta, la recuperación de respuestas relevantes y la recomendación de soluciones contextualizadas.

### 2.1 Evolución de los Modelos de Lenguaje en Soporte Técnico

Los enfoques contemporáneos han evolucionado desde modelos estadísticos simples hacia arquitecturas transformer sofisticadas. Los modelos BERT (Bidirectional Encoder Representations from Transformers) y sus variantes como RoBERTa, DistilBERT y DeBERTa han demostrado resultados superiores en tareas de clasificación multiclase y multilabel específicas del dominio técnico (Devlin et al., 2018; Liu et al., 2019; He et al., 2020). Estos modelos, cuando se especializan mediante fine-tuning en corpus técnicos, pueden capturar matices semánticos específicos del dominio que los modelos generales no logran identificar.

La aparición de modelos especializados como Sentence-BERT (SBERT) ha revolucionado la generación de embeddings para tareas de recuperación semántica (Reimers & Gurevych, 2019). A diferencia de BERT tradicional, SBERT está optimizado para generar representaciones vectoriales densas que preservan la similitud semántica, lo que es fundamental para sistemas de recuperación de información técnica.

### 2.2 Arquitecturas RAG en Soporte Técnico

Las arquitecturas RAG (Retrieval-Augmented Generation) han emergido como el estándar de facto para sistemas de soporte técnico que combinan recuperación de información con generación de respuestas (Lewis et al., 2020). Estas arquitecturas permiten que los modelos de lenguaje accedan dinámicamente a bases de conocimiento externas durante la generación, superando las limitaciones de memoria y actualización de los modelos parametrizados.

En el contexto del soporte técnico, los sistemas RAG implementan típicamente un pipeline de dos etapas: (1) recuperación de documentos relevantes mediante búsqueda vectorial, y (2) generación de respuestas contextualizadas utilizando los documentos recuperados. La efectividad de estos sistemas depende críticamente de la calidad de los embeddings utilizados y la precisión del mecanismo de recuperación.

### 2.3 Aplicaciones Empresariales Actuales

Empresas tecnológicas líderes como IBM, SAP y Microsoft han implementado soluciones NLP para análisis semántico de tickets y generación de respuestas automatizadas (Saxena et al., 2021). Estas implementaciones típicamente incluyen módulos de clasificación automática, extracción de entidades, y sistemas de recomendación basados en similitud semántica.

El uso de técnicas de resumen automático extractivo y abstractivo permite procesar tickets extensos y extraer información clave para facilitar la priorización y el enrutamiento inteligente (Gupta & Gupta, 2020). Estos enfoques son particularmente valiosos en entornos de alto volumen donde la clasificación manual resulta impracticable.

## 3. Bases de Conocimiento como Entrada para Recuperación de Información

### 3.1 Transición hacia Recuperación Semántica

Una tendencia dominante en la industria es la utilización de bases de conocimiento estructuradas como corpus semántico para alimentar sistemas de recuperación y recomendación en tareas de soporte. Estas bases incluyen documentación técnica oficial, artículos de resolución de problemas, FAQ especializadas y respuestas validadas por la comunidad.

Los métodos tradicionales de recuperación basados en TF-IDF (Term Frequency-Inverse Document Frequency) o BM25 (Best Matching 25) han dado paso progresivamente a técnicas vectoriales que representan textos como embeddings densos, capturando relaciones semánticas más profundas y contextuales (Johnson et al., 2021). Esta transición ha sido impulsada fundamentalmente por el surgimiento de modelos como Sentence-BERT, que permiten generar representaciones vectoriales eficientes y semánticamente coherentes para documentos y consultas.

### 3.2 Arquitecturas de Embeddings Especializados

Los sistemas modernos de recuperación implementan arquitecturas de embeddings especializados que van más allá de los modelos generales. Modelos como E5 (Embeddings from bidirectional Encoder representations) han demostrado rendimiento superior en benchmarks de recuperación semántica, particularmente en dominios técnicos (Wang et al., 2022). Estos modelos utilizan estrategias de preentrenamiento contrastivo que optimizan específicamente la tarea de recuperación.

La familia de modelos MPNet (Masked and Permuted Pre-training) combina las ventajas de BERT y XLNet, resultando en representaciones más robustas para tareas de recuperación de información técnica (Song et al., 2020). Por otro lado, modelos como MiniLM ofrecen un balance optimizado entre rendimiento y eficiencia computacional, siendo particularmente útiles en aplicaciones de producción con restricciones de recursos (Wang et al., 2020).

### 3.3 Integración con Bases de Datos Vectoriales

El uso de retrievers vectoriales se ha vuelto esencial en arquitecturas RAG modernas. Bases de datos vectoriales especializadas como ChromaDB, FAISS, Milvus y Weaviate han surgido como soluciones optimizadas para almacenamiento y recuperación eficiente de vectores de alta dimensión (Johnson et al., 2019; Douze et al., 2024).

En este proyecto, inicialmente se utilizó Weaviate como base de datos vectorial. Weaviate fue seleccionada por su robustez empresarial, arquitectura distribuida, integración nativa con múltiples modelos de lenguaje (OpenAI, Cohere, Hugging Face), y capacidades avanzadas de consulta mediante GraphQL (Weaviate, 2023). Su arquitectura modular y capacidad de escalamiento horizontal la posicionan como una solución de grado empresarial para aplicaciones de producción.

Sin embargo, durante el desarrollo del proyecto se migró a ChromaDB por consideraciones prácticas específicas del entorno de investigación académica. ChromaDB ofrece ventajas significativas para prototipado e investigación: eliminación de costos de infraestructura cloud, reducción sustancial de latencia al operar localmente, compatibilidad nativa con Google Colab sin configuración adicional, y simplicidad de despliegue sin requerimientos de servicios externos. Estas características resultan fundamentales para investigación académica donde la reproducibilidad, control de costos y facilidad de experimentación son prioritarias.

ChromaDB mantiene las capacidades esenciales requeridas: filtrado nativo por metadatos, búsqueda híbrida combinando similitud semántica con criterios estructurados, y rendimiento adecuado para conjuntos de datos de escala media (hasta millones de vectores). Esta migración demostró que para aplicaciones de investigación y desarrollo, la simplicidad y control local pueden superar las ventajas de arquitecturas distribuidas más complejas.

## 4. Comparación de Enfoques Vectoriales y Clásicos

### 4.1 Limitaciones de Sistemas Clásicos

Los sistemas clásicos de recuperación de información, implementados en plataformas como Apache Lucene y Elasticsearch, utilizan modelos estadísticos que representan documentos como bolsas de palabras (bag-of-words). Aunque estos sistemas son computacionalmente eficientes y relativamente simples de implementar y mantener, presentan limitaciones fundamentales en la comprensión semántica profunda, lo cual restringe significativamente su capacidad para responder consultas formuladas en lenguaje natural (Manning et al., 2008).

Estas limitaciones son particularmente pronunciadas en dominios técnicos donde existe alta variabilidad terminológica, uso de sinónimos especializados, y donde la relevancia depende fuertemente del contexto semántico más que de la coincidencia léxica exacta.

### 4.2 Intentos de Búsqueda Semántica con Bases Relacionales

Existen esfuerzos para implementar capacidades de búsqueda semántica utilizando bases de datos relacionales tradicionales, principalmente mediante extensiones especializadas. PostgreSQL con la extensión pgvector permite almacenar y consultar vectores de embeddings utilizando SQL estándar (PostgreSQL, 2023). De manera similar, sistemas como Azure SQL Database han incorporado capacidades de búsqueda vectorial mediante extensiones propietarias.

Sin embargo, estas soluciones presentan limitaciones significativas comparadas con bases de datos vectoriales especializadas:

**Limitaciones de Rendimiento:** - Índices menos optimizados para espacios de alta dimensionalidad - Mayor consumo de memoria para operaciones vectoriales - Latencias superiores en consultas de similitud a gran escala - Escalabilidad limitada para billones de vectores

**Limitaciones Funcionales:** - Soporte limitado para métricas de distancia especializadas - Carencia de optimizaciones específicas para ANN (Approximate Nearest Neighbor) - Integración compleja con pipelines de ML/NLP - Falta de funcionalidades nativas para filtrado híbrido semántico-estructurado

**Complejidad Operacional:** - Requiere expertise tanto en SQL como en operaciones vectoriales - Configuración y tuning más complejos para cargas de trabajo vectoriales - Backup y recuperación más complejos para datos de alta dimensionalidad

Estas limitaciones hacen que, aunque técnicamente posible, el uso de bases relacionales para búsqueda semántica sea subóptimo comparado con soluciones especializadas como ChromaDB, Pinecone o Weaviate, particularmente en aplicaciones que requieren alto rendimiento y escalabilidad (Li et al., 2023).

### 4.3 Ventajas de Sistemas Vectoriales

En contraste, los sistemas vectoriales modernos utilizan embeddings generados por modelos de aprendizaje profundo, permitiendo recuperar documentos basados en similitud semántica en lugar de coincidencia léxica superficial (Malkov & Yashunin, 2020). Estos sistemas pueden identificar relaciones semánticas complejas, manejar sinónimos y variaciones terminológicas, y capturar dependencias contextuales que los sistemas clásicos no pueden procesar.

La implementación de algoritmos de búsqueda aproximada de vecinos más cercanos (Approximate Nearest Neighbor, ANN) como HNSW (Hierarchical Navigable Small World) permite realizar búsquedas vectoriales eficientes incluso en espacios de alta dimensionalidad, manteniendo latencias acceptables para aplicaciones de producción (Malkov & Yashunin, 2018).

### 4.4 Enfoques Híbridos y Reranking

Los sistemas más efectivos combinan las fortalezas de ambos enfoques mediante arquitecturas híbridas que utilizan recuperación vectorial para la selección inicial de candidatos, seguida de reranking mediante modelos más sofisticados. Los CrossEncoders, que procesan conjuntamente la consulta y cada documento candidato, pueden proporcionar scores de relevancia más precisos que los bi-encoders utilizados en la fase de recuperación inicial (Reimers & Gurevych, 2019).

Esta estrategia de pipeline multi-etapa permite balancear eficiencia computacional con precisión de recuperación, siendo particularmente efectiva en sistemas de soporte técnico donde la precisión en los primeros resultados es crítica para la experiencia del usuario.

## 5. Casos Empresariales Relevantes

### 5.1 Microsoft Azure Support

Microsoft ha incorporado extensivamente modelos de NLP en su plataforma Azure para análisis automático de tickets y sugerencia de respuestas basadas en documentación técnica. Su implementación utiliza arquitecturas híbridas que combinan embeddings semánticos, sistemas de ranking multi-etapa y técnicas de respuesta generativa (Microsoft Learn, 2023). El sistema procesa automáticamente tickets entrantes, los clasifica por servicio y urgencia, y sugiere documentación relevante y soluciones potenciales basadas en casos históricos similares.

La plataforma integra múltiples fuentes de conocimiento incluyendo documentación oficial, casos de soporte históricos, y contribuciones de la comunidad developer, utilizando técnicas de fusión de rankings para optimizar la relevancia de las sugerencias.

### 5.2 Zendesk Answer Bot

Zendesk ha desarrollado “Answer Bot”, un sistema de inteligencia artificial que utiliza NLP avanzado para sugerir artículos de ayuda relevantes automáticamente cuando un usuario envía un ticket (Zendesk, 2023). El sistema ha logrado reducir en un 10-30% el volumen de tickets que requieren intervención humana directa, demostrando el impacto significativo de las tecnologías NLP en la eficiencia operacional.

Answer Bot implementa técnicas de aprendizaje continuo que mejoran sus recomendaciones basándose en el feedback implícito de los usuarios (aceptación o rechazo de sugerencias) y el feedback explícito de los agentes de soporte.

### 5.3 ServiceNow Predictive Intelligence

ServiceNow integra modelos de NLP con su módulo “Predictive Intelligence”, que clasifica y enruta tickets automáticamente utilizando modelos entrenados en datos históricos extensos (ServiceNow, 2022). El sistema también implementa funcionalidades de recomendación de artículos de la base de conocimiento y predicción de resolución, utilizando técnicas de aprendizaje automático para optimizar la asignación de recursos.

La plataforma incluye capacidades de análisis de sentimiento para priorizar tickets con mayor urgencia emocional y detectar patrones de escalación potencial.

### 5.4 Salesforce Service Cloud Einstein

La plataforma Salesforce Service Cloud ha implementado bots conversacionales que combinan NLP y búsqueda semántica para asistir tanto a clientes como a agentes en tiempo real (Salesforce, 2023). Estas herramientas son alimentadas por bases vectoriales generadas a partir de documentación técnica, casos históricos y interacciones previas, utilizando arquitecturas transformer para generar respuestas contextualizadas.

El sistema integra capacidades de procesamiento multimodal que pueden analizar no solo texto sino también imágenes y documentos adjuntos para proporcionar asistencia más comprehensiva.

## 6. Medidas de Evaluación en Recuperación de Información

### 6.1 Métricas Tradicionales de Recuperación

La evaluación rigurosa de sistemas de recuperación de información es fundamental para validar la efectividad de las soluciones propuestas. Las métricas tradicionales como Precision, Recall y F1-score continúan siendo ampliamente utilizadas, pero requieren adaptación y complementación con métricas específicas para el paradigma de recuperación semántica basado en embeddings.

**Precision** mide la proporción de documentos relevantes entre los documentos recuperados, siendo crucial cuando se busca minimizar falsos positivos. En contextos de soporte técnico, recomendar artículos irrelevantes puede generar frustración y pérdida de confianza en el sistema.

**Recall** evalúa la proporción de documentos relevantes recuperados sobre el total de documentos relevantes disponibles. Esta métrica es particularmente crítica en soporte técnico, donde omitir información relevante puede resultar en resolución inadecuada del problema del usuario.

**F1-Score** representa la media armónica entre precision y recall, proporcionando una métrica balanceada especialmente útil cuando ambos aspectos son igualmente importantes, como en este proyecto donde tanto el exceso como la omisión de información relevante afectan la experiencia del usuario.

### 6.2 Métricas de Ranking y Posición

**Mean Reciprocal Rank (MRR)** es fundamental cuando el sistema devuelve listas ordenadas de resultados y se busca evaluar qué tan pronto aparece la respuesta relevante. En soporte técnico, esta métrica es valiosa para evaluar la utilidad de los primeros resultados mostrados al agente, ya que típicamente solo se revisan los primeros 3-5 resultados.

**Normalized Discounted Cumulative Gain (nDCG)** considera tanto la relevancia de los resultados como su posición en la lista, aplicando un descuento logarítmico que penaliza resultados relevantes en posiciones inferiores. Es especialmente útil cuando múltiples respuestas son relevantes pero existe una preferencia clara por que las más útiles aparezcan primero.

### 6.3 Métricas Específicas para Sistemas RAG

**Precision@k y Recall@k** están diseñadas específicamente para evaluar la calidad de los primeros k resultados en sistemas que devuelven múltiples documentos ordenados por relevancia.

* **Precision@k** mide la proporción de resultados relevantes entre los primeros k documentos recuperados. Por ejemplo, si entre los primeros 5 artículos sugeridos, 3 son relevantes, entonces Precision@5 = 3/5 = 0.6.
* **Recall@k** evalúa cuántos documentos relevantes fueron recuperados entre los primeros k, comparado con el total de documentos relevantes disponibles. Si hay 4 documentos relevantes totales y el sistema recupera 3 dentro de los primeros 5, entonces Recall@5 = 3/4 = 0.75.

### 6.4 Métricas Avanzadas para Evaluación RAG

Las arquitecturas RAG requieren métricas especializadas que evalúen no solo la recuperación sino también la calidad de la generación y la coherencia entre ambas fases:

**Answer Relevancy** mide qué tan bien la respuesta generada aborda específicamente la pregunta formulada, evaluando la alineación semántica entre consulta y respuesta (Es et al., 2023).

**Context Precision** evalúa qué proporción del contexto recuperado es realmente relevante para responder la pregunta, identificando ruido en la fase de recuperación.

**Context Recall** mide si toda la información necesaria para responder la pregunta está presente en el contexto recuperado.

**Faithfulness** evalúa si la respuesta generada es factualmente consistente con el contexto proporcionado, detectando alucinaciones o inconsistencias.

### 6.5 Métricas de Similitud Semántica

**BERTScore** utiliza representaciones contextuales de BERT para evaluar la similitud semántica entre respuestas generadas y respuestas de referencia, proporcionando una evaluación más matizada que métricas basadas en coincidencia léxica como BLEU o ROUGE (Zhang et al., 2019).

En este proyecto se implementó BERTScore utilizando el modelo distiluse-base-multilingual-cased-v2, optimizado para evaluación de similitud semántica cross-lingual, aunque se aplicó específicamente a contenido en inglés para mantener consistencia con el corpus de documentación técnica.

### 6.6 Aplicación al Proyecto

En este proyecto se implementó un framework comprehensivo de evaluación que incluye:

* **Métricas de recuperación tradicionales**: Precision@k, Recall@k (k=1,3,5,10), MRR, nDCG
* **Métricas RAG especializadas**: Answer Relevancy, Context Precision, Context Recall, Faithfulness (implementadas via RAGAS)
* **Evaluación semántica**: BERTScore para validación de similitud semántica
* **Análisis pre/post reranking**: Comparación de métricas antes y después de aplicar CrossEncoder para cuantificar el impacto del reranking

Esta combinación permite evaluar integralmente tanto la efectividad de la recuperación como la calidad de las respuestas generadas, proporcionando insights detallados sobre el rendimiento de cada componente del pipeline RAG en el contexto específico del soporte técnico de Azure.

## Referencias del Capítulo

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P. E., … & Jégou, H. (2024). The Faiss library. *arXiv preprint arXiv:2401.08281*.

Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2023). RAGAS: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Gupta, S., & Gupta, S. K. (2020). Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121, 49-65.

He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535-547.

Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535-547.

Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., & Zhang, M. (2023). Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., … & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Malkov, Y. A., & Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4), 824-836.

Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4), 824-836.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Microsoft Learn. (2023). *Azure AI services documentation*. https://learn.microsoft.com/en-us/azure/ai-services/

PostgreSQL. (2023). *pgvector: Open-source vector similarity search for Postgres*. https://github.com/pgvector/pgvector

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv preprint arXiv:1908.10084*.

Salesforce. (2023). *Einstein for Service Cloud*. https://www.salesforce.com/products/service-cloud/features/service-cloud-einstein/

Saxena, A., Kochhar, P. S., & Lo, D. (2021). A machine learning approach to predict and categorize questions in stack overflow. *Empirical Software Engineering*, 26(4), 1-34.

ServiceNow. (2022). *Predictive Intelligence*. https://www.servicenow.com/products/predictive-intelligence.html

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. Y. (2020). MPNet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33, 16857-16867.

Wang, L., Yang, N., Huang, J., Chang, M. W., & Wang, W. (2022). Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33, 5776-5788.

Weaviate. (2023). *Weaviate: The AI-native open-source vector database*. https://weaviate.io/

Zendesk. (2023). *Answer Bot*. https://www.zendesk.com/service/answer-bot/

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). BERTScore: Evaluating text generation with BERT. *arXiv preprint arXiv:1904.09675*.