Table of Contents

# 2. ESTADO DEL ARTE

## 2.1 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.

## 2.2 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.

### 2.2.1 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.

### 2.2.2 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.

### 2.2.3 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.

## 2.3 Bases de Conocimiento como Entrada para Recuperación de Información

### 2.3.1 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.

### 2.3.2 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).

### 2.3.3 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.

## 2.4 Comparación de Enfoques Vectoriales y Clásicos

### 2.4.1 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.

### 2.4.2 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).

### 2.4.3 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).

### 2.4.4 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.

## 2.5 Casos Empresariales Relevantes

### 2.5.1 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.

### 2.5.2 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.

### 2.5.3 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.

### 2.5.4 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.

## 2.6 Medidas de Evaluación en Recuperación de Información

### 2.6.1 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.

### 2.6.2 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.

### 2.6.3 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.

### 2.6.4 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.

### 2.6.5 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.

### 2.6.6 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.

## 2.7 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*.