PROTOTIPO DE SISTEMA BASADO EN LA IA DE DEEPSEEK PARA GESTIONAR LAS BASES DE DATOS DE PROPIEDAD INTELECTUAL DE LOS GRUPOS DE INVESTIGACIÓN DE LA E3T

DUVAN ANDRES RODRIGUEZ SUAREZ
DAVID SANTIAGO MOSQUERA QUITIAN

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2026

PROTOTIPO DE SISTEMA BASADO EN LA IA DE DEEPSEEK PARA GESTIONAR LAS BASES DE DATOS DE PROPIEDAD INTELECTUAL DE LOS GRUPOS DE INVESTIGACIÓN DE LA E3T

DUVAN ANDRES RODRIGUEZ SUAREZ
DAVID SANTIAGO MOSQUERA QUITIAN

Trabajo de grado para optar al título de Ingeniería Electrónica

Director
Homero Ortega Boada
Doctor en Ingeniería

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2026

DEDICATORIA

Agradezco a Dios por estar siempre presente, no solo durante esta etapa, sino a lo largo de toda mi vida.

Dedico este trabajo, en primer lugar, a mis padres, quienes han sido mi mayor apoyo y guía a lo largo de todo este camino académico y personal. Su amor incondicional, su confianza y su constante acompañamiento me dieron la fortaleza necesaria para seguir adelante. Su esfuerzo, paciencia y sacrificio han sido fundamentales para que hoy pueda alcanzar esta meta. A mis hermanos, gracias por su comprensión, su apoyo y por estar siempre presentes.

A mis nonos, a quienes debo un agradecimiento muy especial, por su presencia constante, su cariño y sus palabras llenas de sabiduría. Su ejemplo de vida, su apoyo incondicional y su fe en mí han sido una fuente permanente de motivación y fortaleza en cada etapa de mi formación. Su amor ha sido un refugio y un impulso para no rendirme.

A toda mi familia, por creer en mí, acompañarme y sostenerme con su apoyo en cada paso de este proceso. Su respaldo ha sido esencial para mantener la constancia, la disciplina y el compromiso necesarios para culminar este importante logro.

A mis compañeros de universidad, con quienes compartí aprendizajes, retos y experiencias que marcaron profundamente mi crecimiento académico y personal. Gracias por el compañerismo, el apoyo mutuo y los momentos que hicieron de este proceso una etapa enriquecedora. A mis amigos de infancia, por su amistad genuina, su apoyo constante y por recordarme siempre mis raíces.

A Muñeca, mi mascota y compañera fiel, que incluso en silencio estuvo siempre a mi lado durante las largas noches de estudio, regalándome su amor y compañía, y que hoy vive para siempre en mi corazón.

Finalmente, a los profesores que me acompañaron durante este proceso, por su

guía, paciencia y valiosas correcciones. Sus aportes, enseñanzas y exigencia académica contribuyeron significativamente a mi formación profesional y al desarrollo de este trabajo, dejando una huella importante que llevaré conmigo más allá de esta etapa.

Duvan Andres Rodriguez Suarez

Agradezco primeramente a Dios, arquitecto de mi vida, por darme salud y las oportunidades para alcanzar esta meta. Gracias por ser mi guía en los momentos de duda, por protegerme y por mantenerme firme en el camino hacia mis sueños.

A mi madre, mi mayor ejemplo de resiliencia y sacrificio. Gracias por haber asumido con una fortaleza inquebrantable el reto de criarnos a mis hermanos y a mí; tu dedicación es la base de mis logros. Me enseñaste que el trabajo duro no es solo un deber, sino una virtud, y que hacer las cosas bien es el único camino al éxito.

A mi pareja, Katalina, quien fue mi roca y mi refugio durante todo este trayecto académico. Gracias por tu paciencia infinita, por enseñarme que saber escuchar es tan importante como saber hablar, y por transformarme en un hombre capaz de amar con generosidad y madurez.

A mis hijas de cuatro patas, mis compañeras silenciosas de desvelos. Gracias por su amor incondicional y puro, ese que no pide nada a cambio y que lograba renovar mis energías con solo un movimiento de cola tras las jornadas más agotadoras.

David Santiago Mosquera Quitian

AGRADECIMIENTOS

Deseamos manifestar nuestra gratitud a la Universidad Industrial de Santander y a la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones (E3T), por ser el espacio donde forjamos nuestra visión profesional. Agradecemos a cada uno de los docentes que, con su rigor y conocimiento, nos brindaron las herramientas necesarias para abordar este reto tecnológico con éxito.

Un reconocimiento especial merece el profesor Homero Ortega Boada, director de esta investigación. Su experticia y su guía constante fueron el motor que permitió aterrizar conceptos complejos de Inteligencia Artificial en una solución funcional y estructurada. Gracias por su compromiso, por la libertad creativa que nos otorgó y por sus valiosas correcciones que elevaron la calidad de este trabajo.

Asimismo, agradecemos a nuestros amigos y compañeros de estudio. Las discusiones técnicas y el apoyo mutuo durante estos años de carrera fueron piezas clave para fortalecer nuestro criterio y superar los obstáculos del desarrollo.

De manera muy personal, expresamos nuestro amor y gratitud a nuestras familias. Este proyecto es el resultado de su esfuerzo, sacrificio y fe incondicional en nosotros.

A nuestros padres y parejas, gracias por ser nuestro soporte emocional y por darnos la motivación diaria para culminar nuestra etapa como estudiantes de Ingeniería Electrónica.

A todos los que, de forma directa o indirecta, aportaron a la realización de este prototipo, nuestro más sincero agradecimiento.

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

ANEXOS

Los anexos están disponibles en el Repositorio Institucional

GLOSARIO

API: conjunto de reglas y especificaciones que permiten que diferentes aplicaciones de software se comuniquen entre sí, facilitando el intercambio de datos y la integración de funcionalidades entre sistemas.

CHUNKING: técnica utilizada en el procesamiento de lenguaje natural que consiste en dividir textos extensos en segmentos más pequeños, con el fin de facilitar su análisis y procesamiento.

EMBEDDING: representación numérica de elementos como texto, imágenes o audio en un espacio vectorial, donde la posición de cada elemento refleja relaciones semánticas útiles para algoritmos de aprendizaje automático.

LLM: tipo de modelo de aprendizaje profundo entrenado con grandes volúmenes de datos textuales, capaz de comprender, generar y manipular lenguaje natural para realizar diversas tareas lingüísticas.

NOSQL: enfoque de gestión de bases de datos que permite almacenar y consultar información sin seguir estrictamente el modelo relacional tradicional, facilitando el manejo de grandes volúmenes de datos y estructuras flexibles.

PLN: área de la inteligencia artificial que se enfoca en el desarrollo de técnicas y modelos que permiten a los sistemas computacionales analizar, comprender y generar lenguaje humano.

RAG: enfoque que combina modelos de lenguaje con mecanismos de recuperación de información, permitiendo que el sistema consulte fuentes externas de conocimiento antes de generar una respuesta.

SQL: lenguaje de programación utilizado para definir, consultar y manipular datos almacenados en bases de datos relacionales, organizadas en tablas compuestas por filas y columnas.

RESUMEN

TÍTULO PROTOTIPO DE SISTEMA BASADO EN LA IA DE DEEPSEEK PARA GESTIONAR LAS BASES DE DATOS DE PROPIEDAD INTELECTUAL DE LOS GRUPOS DE INVESTIGACIÓN DE LA E3T [*]

AUTOR: DUVAN ANDRES RODRIGUEZ SUAREZ, DAVID SANTIAGO MOSQUERA QUITIAN [**]

PALABRAS CLAVE: CHATBOT, INTELIGENCIA ARTIFICIAL, NLP

DESCRIPCIÓN: El presente trabajo de grado surge del interés por evaluar el potencial de la inteligencia artificial como herramienta para mejorar la gestión de información dentro de los grupos de investigación de la E3T, tomando como caso de estudio la base de datos del grupo CPS. Actualmente, la administración y consulta de la producción intelectual del grupo incluyendo proyectos, artículos, reportes técnicos y demás documentos— resulta compleja debido a la diversidad de formatos y la necesidad de integrar información dispersa en diferentes sistemas. Con el fin de abordar esta problemática, se propone el desarrollo de un prototipo de agente inteligente capaz de interpretar consultas formuladas en lenguaje natural y transformarlas en consultas SQL y no SQL mediante el uso de modelos de lenguaje de gran escala (LLMs), técnicas de procesamiento del lenguaje natural y mecanismos de búsqueda semántica. El sistema incorpora además una base de datos vectorial que permite la recuperación aumentada de información (RAG), facilitando el acceso contextualizado a contenido estructurado y no estructurado del grupo. Aunque el prototipo no busca ser una solución definitiva, sí constituye una prueba de concepto que demuestra la aplicabilidad real de la IA en la organización y consulta del conocimiento académico, evidenciando cómo estas tecnologías pueden reducir la carga administrativa, mejorar la disponibilidad de la información y sentar las bases para el desarrollo futuro de herramientas más avanzadas dentro de la E3T.

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Ingeniería Electrónica. Director: Homero Ortega Boada. Doctor en Ingeniería. Codirector: Álvaro Enrique Patiño. Representante Directo de TesAmerica.

ABSTRACT

TITLE: PROTOTYPE OF AN AI-BASED SYSTEM USING DEEPSEEK TO MANAGE THE INTELLECTUAL PROPERTY DATABASES OF THE E3T [*]

AUTOR: DUVAN ANDRES RODRIGUEZ SUAREZ, DAVID SANTIAGO MOSQUERA QUITIAN [**]

Keywords: CHATBOT, ARTIFICIAL INTELLIGENCE, NLP

Description: This undergraduate thesis arises from the interest in evaluating the potential of artificial intelligence as a tool to improve information management within the research groups of the E3T, using the CPS group's database as a case study. Currently, the administration and consultation of the group's intellectual output—including projects, articles, technical reports, and other documents—are challenging due to the diversity of formats and the need to integrate information dispersed across multiple systems. To address this issue, the development of an intelligent agent prototype is proposed. This agent is capable of interpreting queries formulated in natural language and transforming them into SQL and non-SQL queries through the use of large language models (LLMs), natural language processing techniques, and semantic search mechanisms. The system also incorporates a vector database that enables Retrieval-Augmented Generation (RAG), facilitating contextualized access to both structured and unstructured content from the group. Although the prototype is not intended to be a definitive solution, it represents a proof of concept that demonstrates the practical applicability of AI in the organization and querying of academic knowledge. Furthermore, it highlights how these technologies can reduce administrative workload, improve information accessibility, and lay the foundation for the future development of more advanced tools within the E3T.

---

[*] Degree work

[**] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Ingeniería Electrónica. Advisor: Homero Ortega Boada. Doctor in Engineering. Co-advisor: Álvaro Enrique Patiño. Direct Representative of TesAmerica.

INTRODUCCIÓN

En los últimos años, la inteligencia artificial ha dejado de ser un concepto reservado para especialistas y se ha convertido en una tecnología presente en la vida cotidiana. Su impacto es tal que diversos autores coinciden en que nos encontramos ante una transformación comparable a grandes revoluciones tecnológicas previas, como la sociedad industrial, debido a su capacidad de modificar procesos productivos, educativos y organizacionales[1]. La inteligencia artificial generativa, en particular, ha acelerado este cambio al permitir que usuarios sin conocimientos técnicos avanzados interactúen con sistemas capaces de razonar, sintetizar información y responder consultas complejas[2]. Herramientas como ChatGPT, Gemini, Claude o DeepSeek han marcado un punto de inflexión que redefine procesos laborales, educativos y administrativos, abriendo interrogantes sobre el impacto y las oportunidades de estas tecnologías en distintos sectores.

En este contexto de cambio, las instituciones académicas enfrentan un desafío doble: adaptarse al ritmo del desarrollo tecnológico y gestionar de manera eficiente la creciente cantidad de información que producen sus grupos de investigación. La E3T, al igual que muchas unidades académicas, cuenta con una amplia diversidad de documentos técnicos, artículos y registros institucionales que requieren un manejo organizado. Sin embargo, los sistemas tradicionales de almacenamiento y consulta basados en palabras clave presentan limitaciones significativas, ya que no permiten interpretar adecuadamente el significado de las consultas ni realizar búsquedas semánticas profundas, lo que incrementa el tiempo y el esfuerzo necesarios

---

[1] Daniel Jurafsky y James H. Martin. *Speech and Language Processing*. 3.ª ed. Pearson, 2023.

[2] IBM. *What is artificial intelligence?* s.f. URL: https://www.ibm.com/think/topics/artificial-intelligence (visitado 15-01-2025).

para recuperar información relevante[3].

En particular, se seleccionó al grupo de investigación CPS como caso de estudio para explorar cómo la inteligencia artificial puede asistir en la integración, organización y consulta ágil de información científica y administrativa. Este grupo, al igual que muchos otros en la E3T, cuenta con documentos distribuidos en diversos repositorios y formatos, lo cual dificulta la explotación eficiente del conocimiento. Esta problemática es común en entornos académicos donde, a pesar de la generación constante de información valiosa, se carece de herramientas que permitan consultarla de manera inteligente y contextualizada. Frente a este escenario, los avances en procesamiento del lenguaje natural, modelos de lenguaje de gran escala y bases de datos vectoriales ofrecen nuevas posibilidades para transformar la gestión del conocimiento académico[4].

Motivado por este panorama, el presente trabajo de grado propone el desarrollo de un prototipo de agente inteligente orientado a la gestión de la base de datos del grupo CPS y diseñado para interpretar solicitudes formuladas en lenguaje natural. Este agente combina técnicas de inteligencia artificial generativa y mecanismos de recuperación aumentada de información (RAG) para transformar preguntas complejas en consultas SQL o búsquedas vectoriales, según el tipo de información solicitada[5]. Para ello, se construyó una base de datos vectorial capaz de almacenar representaciones embebidas de documentos, permitiendo identificar fragmentos conceptualmente relacionados incluso cuando las palabras utilizadas difieren de la consulta

---

[3] Christopher D. Manning, Prabhakar Raghavan e Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[4] Nils Reimers e Iryna Gurevych. «Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks». En: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 2019.

[5] Patrick Lewis et al. «Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks». En: *Advances in Neural Information Processing Systems* 33 (2020), págs. 9459-9474.

original.

1. OBJETIVOS

1.1. OBJETIVO GENERAL

Desarrollar un prototipo basado en inteligencia artificial generativa para demostrar la viabilidad de los modelos de lenguaje de gran escala (LLMs) en la mejora de la gestión de la producción intelectual de los grupos de investigación de la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones (E3T), facilitando la organización, estructuración y consulta semántica de la información.

1.2. OBJETIVOS ESPECÍFICOS

Desarrollar un prototipo funcional basado en inteligencia artificial generativa y modelos de lenguaje de gran escala (LLMs) para gestionar y organizar la producción intelectual del grupo de investigación seleccionado, permitiendo la consulta semántica y la recuperación eficiente de información relevante.

Demostrar la viabilidad de la solución en el contexto del grupo de investigación seleccionado, evaluando su capacidad para mejorar la gestión de la información, reduciendo la carga administrativa y facilitando el acceso a datos clave para la toma de decisiones.

Integrar el prototipo con las bases de datos existentes del grupo de investigación, optimizando el acceso a la información sin realizar modificaciones estructurales en los sistemas actuales de gestión de datos.

Evaluar el impacto de la solución en la eficiencia operativa del grupo de investigación, considerando aspectos como la reducción de tiempo en la gestión de la producción intelectual y la mejora en la calidad de la información disponible para los investigadores y administradores.

2. MARCO TEÓRICO

2.1. INTELIGENCIA ARTIFICIAL Y PLN

La inteligencia artificial (IA) se ha consolidado como una de las tecnologías más influyentes en la transformación digital de las últimas décadas. Su propósito fundamental es desarrollar sistemas capaces de realizar tareas que tradicionalmente requieren habilidades humanas, como el razonamiento, el aprendizaje, la inferencia y la resolución de problemas. Entre sus ramas más destacadas se encuentran el aprendizaje automático (machine learning), el aprendizaje profundo (deep learning) y los sistemas basados en conocimiento, los cuales permiten crear modelos capaces de identificar patrones, interpretar datos y tomar decisiones autónomas en entornos cada vez más complejos[6].

El procesamiento del lenguaje natural (PLN) es un subcampo de la informática y la inteligencia artificial que emplea técnicas de machine learning y aprendizaje profundo para permitir que las computadoras comprendan, interpreten y generen lenguaje humano. El PLN combina enfoques de la lingüística computacional, basados en reglas, con modelos estadísticos y algoritmos de aprendizaje automático, facilitando la interacción entre los sistemas computacionales y el lenguaje natural en forma de texto o voz[1].

En el procesamiento de documentos, las herramientas de PLN permiten clasificar información, extraer datos relevantes y generar resúmenes de manera automática, reduciendo significativamente el tiempo y los errores asociados a los procesos ma-

---

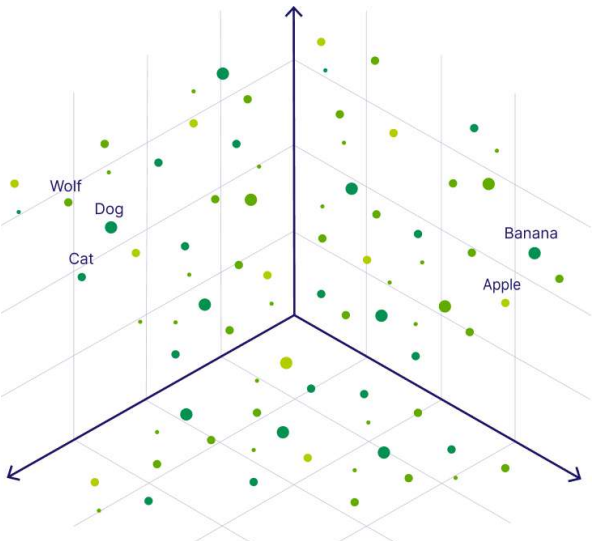
[6] Stuart J. Russell y Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4.ª ed. Hoboken, NJ: Pearson, 2021.

nuales de gestión de información[7].

La búsqueda de información también se ve beneficiada por el uso de técnicas de PLN, ya que estas permiten comprender la intención detrás de las consultas realizadas por los usuarios y ofrecer resultados más precisos y contextualmente relevantes. A diferencia de los enfoques tradicionales basados únicamente en coincidencias de palabras clave, los sistemas impulsados por PLN analizan el significado semántico de palabras y frases, mejorando la experiencia del usuario en la recuperación de documentos y datos empresariales, incluso cuando las consultas son ambiguas o complejas[3].

2.2. BASES DE DATOS VECTORIALES Y EMBEDDINGS

Figura 1. Visualización de embeddings vectoriales.



Fuente: Tomada de https://weaviate.io/blog/vector-embeddings-explained.

---

[7] IBM. *What is natural language processing (NLP)?* s.f. URL: https://www.ibm.com/think/topics/natural-language-processing (visitado 15-01-2025).

El crecimiento acelerado de la información digital ha generado un problema central en el ámbito del big data: ¿cómo almacenar grandes volúmenes de información no estructurada como textos extensos, imágenes o audio y, al mismo tiempo, permitir su recuperación eficiente cuando se requiere? Este desafío no solo implica guardar datos, sino también extraer información relevante de ellos de manera rápida y precisa, especialmente en aplicaciones modernas de inteligencia artificial.

Las bases de datos tradicionales resultan adecuadas para información estructurada, pero presentan limitaciones importantes al momento de recuperar datos no estructurados basándose en su significado. En estos sistemas, la búsqueda suele depender de coincidencias exactas de términos, lo que dificulta capturar relaciones semánticas entre conceptos.

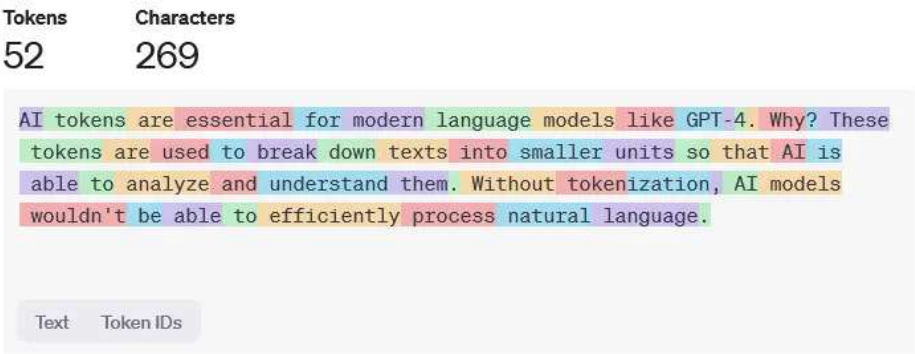
Para abordar este problema, la información se transforma en *embeddings*, representaciones vectoriales que ubican cada objeto dentro de un espacio multidimensional. En este espacio, la cercanía entre vectores indica similitud semántica. La Figura 1 ilustra este principio: conceptos relacionados como *Dog*, *Cat* y *Wolf* aparecen agrupados en una misma región, mientras que términos no relacionados, como *Apple* y *Banana*, se localizan en zonas distintas del espacio vectorial.

Las bases de datos vectoriales aprovechan esta organización para almacenar e indexar los embeddings, permitiendo realizar búsquedas basadas en similitud en lugar de coincidencias exactas. Gracias a este enfoque, es posible recuperar información relevante incluso cuando los términos de la consulta no aparecen explícitamente en los documentos.

Uno de los logros de este proyecto es haber logrado implementar una base de datos vectorial propia, aunque usando servicios existentes para ello, a diferencia de proyectos anteriores desarrollados por el grupo de investigación RadioGIS. Pero los logros van más allá porque se ha logrado combinar el uso de bases de datos vectoriales con bases de datos estruturales y otros aportes que se describen más

adelante

Figura 2. Representación visual del proceso de tokenización.



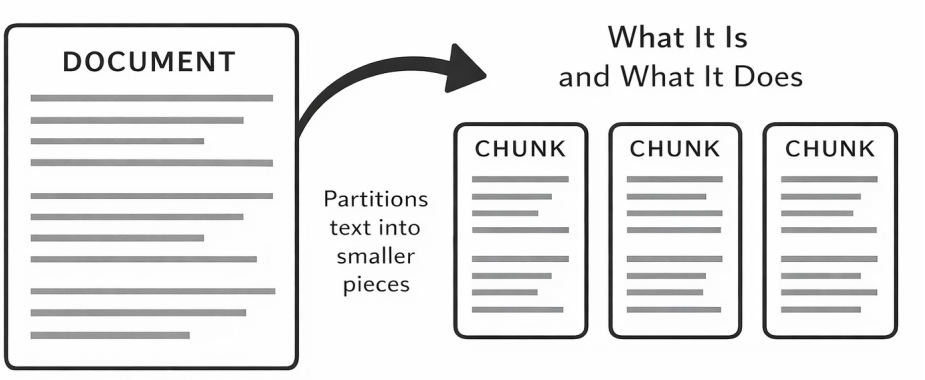
Fuente: Tomada de https://www.ionos.co.uk/digitalguide/websites/web-development/ai-tokens/.

La figura muestra el proceso de *tokenización* de un fragmento de texto, mediante el cual un modelo de lenguaje divide la entrada en unidades discretas denominadas *tokens*. Cada color representa un token individual, que puede corresponder a palabras completas, subpalabras o símbolos. Asimismo, se evidencia la diferencia entre el número de caracteres y el número de tokens, aspecto relevante para el procesamiento interno, la eficiencia computacional y las limitaciones de contexto en los modelos de lenguaje modernos.

2.2.1. Segmentación de documentos y chunking El procesamiento de documentos extensos en sistemas de búsqueda semántica presenta una limitación fundamental: los modelos de lenguaje y de generación de embeddings tienen restric-

ciones en la cantidad de tokens que pueden procesar simultáneamente. Esto impide representar documentos largos de forma íntegra sin afectar el rendimiento o la calidad de la información procesada.

Figura 3. Representación gráfica del proceso de chunking en Retrieval-Augmented Generation.



Fuente: Tomada de https://www.linkedin.com/pulse/qu

Para abordar esta limitación, se emplean técnicas de segmentación de documentos, cuyo objetivo es dividir el contenido en unidades más pequeñas y manejables. Este proceso resulta esencial en sistemas de recuperación de información, ya que permite consultar secciones específicas de un documento de manera más precisa.
En este contexto, el *chunking* consiste en fragmentar un documento en porciones más pequeñas, denominadas *chunks*, que son procesadas de forma independiente para generar sus respectivos embeddings y almacenarlas en una base de datos vectorial. Cada embedding preserva el significado local del fragmento al que representa.
La segmentación puede realizarse mediante criterios simples, como un número fijo de tokens o párrafos, o mediante enfoques más avanzados basados en segmentación semántica, los cuales buscan mantener la coherencia temática de cada fragmento.

Un esquema de chunking adecuado mejora la calidad de la recuperación de información, ya que genera embeddings más precisos y facilita la respuesta a consultas específicas dentro de documentos extensos, convirtiéndose en un componente clave de los sistemas de búsqueda semántica.[5][8][3]

2.2.2. Pgvector Una vez generados los embeddings a partir de información no estructurada y segmentada, surge el desafío de almacenar y consultar eficientemente grandes volúmenes de vectores de alta dimensión. Las bases de datos relacionales tradicionales no están optimizadas para realizar operaciones de similitud vectorial, lo que limita su uso directo en sistemas de búsqueda semántica.

Para abordar este problema, el proyecto utiliza *pgvector*, una extensión de PostgreSQL que permite almacenar vectores de alta dimensión y ejecutar consultas de similitud dentro de una base de datos relacional. Esta funcionalidad resulta fundamental para implementar mecanismos de búsqueda semántica sobre los fragmentos de texto procesados.

Pgvector soporta distintas métricas de comparación, como la similitud del coseno, el producto interno y la distancia euclidiana, las cuales permiten medir la cercanía semántica entre embeddings y recuperar información relevante incluso sin coincidencias exactas de términos.

En el sistema desarrollado, pgvector se integra de forma nativa a través de Supabase, facilitando el almacenamiento de los embeddings asociados a los *chunks* y la ejecución de consultas híbridas que combinan filtros estructurados con búsquedas basadas en significado.

De este modo, pgvector permite unificar el manejo de datos estructurados y no estructurados en una sola infraestructura, constituyéndose como un componente clave

---

[8] Marti A. Hearst. «TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages». En: *Computational Linguistics* 23.1 (1997), págs. 33-64.

para la implementación de un sistema de recuperación de información eficiente y orientado a aplicaciones de inteligencia artificial.

2.3. RECUPERACIÓN SEMÁNTICA Y SIMILITUD

Uno de los principales problemas en los sistemas tradicionales de recuperación de información es su dependencia de la coincidencia literal de palabras clave. Este enfoque resulta limitado cuando los documentos y las consultas utilizan vocabulario distinto para expresar ideas similares, lo que puede provocar la omisión de información relevante a pesar de su cercanía conceptual. Esta limitación se vuelve especialmente crítica en escenarios con grandes volúmenes de información no estructurada.

Para abordar este problema surge la *recuperación semántica*, un enfoque de búsqueda que prioriza el significado del contenido por encima de la coincidencia exacta de términos. En este paradigma, tanto los documentos como las consultas realizadas por los usuarios son transformados en representaciones vectoriales mediante modelos de embeddings, lo que permite capturar relaciones semánticas entre textos incluso cuando estos emplean palabras diferentes para describir conceptos similares[39].

Una vez representados como vectores en un espacio de alta dimensión, el proceso de recuperación de información se basa en la comparación entre la consulta y los documentos almacenados. Esta comparación se realiza empleando métricas de similitud vectorial, las cuales permiten cuantificar el grado de cercanía semántica entre dos representaciones. Entre las métricas más utilizadas se encuentran la similitud del coseno, el producto interno (*inner product*) y la distancia euclidiana.

---

[9] Peter D. Turney y Patrick Pantel. «From Frequency to Meaning: Vector Space Models of Semantics». En: *Journal of Artificial Intelligence Research* 37 (2010), págs. 141-188.

Estas métricas evalúan la relación entre los vectores considerando su orientación y distancia dentro del espacio vectorial, de modo que una mayor similitud o una menor distancia indica una mayor relación semántica entre los textos representados. Gracias a este enfoque, los sistemas de recuperación semántica pueden ofrecer resultados más relevantes y precisos, superando las limitaciones de los métodos tradicionales basados únicamente en coincidencias léxicas[10].

2.4. BASES DE DATOS ESTRUCTURADAS VS NO ESTRUCTURADAS

Los datos en la gestión de la información se clasifican en estructurados y no estructurados, los cuales requieren métodos distintos de almacenamiento y consulta. Los datos estructurados se organizan en formatos definidos, como tablas, lo que facilita su consulta mediante SQL; en este proyecto, corresponden a la información extraída del sitio web del grupo de investigación y almacenada en Supabase. Por su parte, los datos no estructurados no siguen un formato fijo y se encuentran en documentos extensos, por lo que requieren técnicas de procesamiento de lenguaje natural; estos fueron almacenados en Google Drive y procesados mediante segmentación y embeddings para su indexación en una base vectorial. La diferencia entre ambos tipos de datos justifica una solución híbrida que combina consultas SQL y búsquedas semánticas, permitiendo una recuperación de información más eficiente y complementaria.

2.5. GRANDES MODELOS DE LENGUAJE LLM

El crecimiento exponencial de la información digital y la complejidad de las consultas realizadas por los usuarios han evidenciado las limitaciones de los sistemas tradicionales de procesamiento del lenguaje natural, los cuales suelen requerir modelos

---

[10] Charu C. Aggarwal. *Data Mining: The Textbook*. Springer, 2015.

especializados para cada tarea específica. Este enfoque dificulta la escalabilidad y la adaptación de los sistemas a distintos escenarios, especialmente en aplicaciones que demandan comprensión contextual profunda y generación de respuestas coherentes.

Como respuesta a este problema surgen los Grandes Modelos de Lenguaje (*Large Language Models*, LLM), una clase de sistemas de inteligencia artificial diseñados para comprender, generar y razonar sobre el lenguaje natural de manera generalizada. Estos modelos se entrenan utilizando enormes volúmenes de texto, lo que les permite aprender patrones lingüísticos, sintácticos y semánticos a gran escala.

Los LLM se basan principalmente en arquitecturas de tipo *transformer*, las cuales permiten procesar secuencias de texto de forma paralela y capturar relaciones contextuales complejas entre palabras y frases. Gracias al mecanismo de atención, estos modelos pueden identificar dependencias relevantes a largo alcance dentro del texto, superando las limitaciones de arquitecturas secuenciales tradicionales.

Una de las principales características de los LLM es su capacidad para desempeñar múltiples tareas sin necesidad de ajustes específicos para cada una de ellas. Entre estas tareas se incluyen la respuesta a preguntas, la generación y el resumen de textos, la traducción automática, la generación de código, la clasificación de información y la asistencia en sistemas de búsqueda semántica. Esta versatilidad se debe a su entrenamiento sobre corpora masivos y al uso de modelos con miles de millones de parámetros, lo que les permite generalizar con alta precisión a distintos contextos y dominios de aplicación.

En el contexto de los sistemas de recuperación de información y de interacción hombre–máquina, los LLM representan un componente clave para la construcción de interfaces inteligentes capaces de ofrecer respuestas contextualizadas, coherentes y alineadas con las necesidades del usuario.

2.6. DEEPSEEK

La incorporación de grandes modelos de lenguaje en aplicaciones reales implica desafíos relacionados con el costo computacional, la latencia y la accesibilidad a modelos de alto rendimiento, especialmente en proyectos académicos y de investigación. En este contexto, la API de *DeepSeek* se presenta como una alternativa para acceder a modelos avanzados de procesamiento del lenguaje natural y generación de texto.

Los modelos ofrecidos por DeepSeek proporcionan un equilibrio entre capacidad de razonamiento, velocidad de respuesta y eficiencia computacional, lo que facilita su integración en sistemas interactivos. Entre sus principales ventajas se destacan su buen desempeño en tareas de generación de respuestas coherentes y su menor latencia frente a modelos de mayor tamaño, características relevantes en aplicaciones como chatbots o sistemas de consulta académica.

Sin embargo, al igual que otros grandes modelos de lenguaje, DeepSeek presenta limitaciones asociadas a su conocimiento estático y al riesgo de generar respuestas imprecisas en contextos altamente especializados. Estas restricciones hacen necesario complementar su uso con mecanismos de recuperación de información externa.

A pesar de ello, la API de DeepSeek resulta adecuada para el presente proyecto, ya que permite aprovechar las capacidades de los modelos de lenguaje manteniendo un balance entre rendimiento, costo y facilidad de integración, especialmente cuando se combina con técnicas de recuperación aumentada de información.

2.7. n8n

El desarrollo de aplicaciones basadas en inteligencia artificial y recuperación de información implica la integración de múltiples servicios en la nube, como modelos

de lenguaje, bases de datos y APIs externas. La implementación manual de estas integraciones suele ser compleja y demandar un esfuerzo significativo en términos de programación, mantenimiento y escalabilidad.

En este contexto, surge la necesidad de herramientas que permitan orquestar y automatizar flujos de trabajo de manera eficiente. n8n se presenta como una plataforma de código abierto orientada a la automatización de procesos y la integración de servicios, facilitando la construcción de flujos de trabajo complejos en entornos de cómputo en la nube.

La plataforma permite diseñar flujos mediante una interfaz visual basada en nodos, donde cada nodo representa una acción específica, como consultas a bases de datos, consumo de APIs o procesamiento de información. Este enfoque reduce la cantidad de código requerido y ofrece un control claro sobre la lógica y el flujo de datos del sistema.

Entre sus principales ventajas se destacan la flexibilidad, la amplia capacidad de integración y la posibilidad de autoalojamiento, lo que proporciona un mayor control sobre la infraestructura y los datos. Asimismo, su arquitectura modular favorece la escalabilidad y la adaptación a nuevos requerimientos, siendo especialmente útil en proyectos de investigación y prototipos de inteligencia artificial.

En este proyecto, n8n se emplea como herramienta de orquestación para coordinar la interacción entre los modelos de lenguaje, la base de datos vectorial y los servicios externos, permitiendo desarrollar una solución flexible, escalable y mantenible sin incrementar innecesariamente la complejidad del sistema.

2.8. Gemini Developer API

La *Gemini Developer API* fue utilizada en este proyecto para la generación de representaciones vectoriales (*embeddings*) a partir de texto, las cuales permiten el procesamiento y la recuperación semántica de información no estructurada. Esta

API proporciona acceso a modelos de lenguaje desarrollados por Google, optimizados para tareas de comprensión y representación del lenguaje natural.
En particular, se empleó el modelo `models/text-embedding-004`, el cual está diseñado para transformar fragmentos de texto en vectores numéricos de alta dimensionalidad que capturan relaciones semánticas entre los contenidos. Estos embeddings fueron posteriormente almacenados en una base de datos vectorial implementada en Supabase, donde son utilizados para realizar búsquedas por similitud y recuperación eficiente de información relevante.

2.9. Supabase

El desarrollo de sistemas de recuperación semántica y aplicaciones basadas en grandes modelos de lenguaje requiere una infraestructura capaz de gestionar datos estructurados, información no estructurada y operaciones de similitud vectorial. En este contexto, la selección de la plataforma de almacenamiento resulta un aspecto fundamental en el diseño del sistema.
Si bien existen bases de datos vectoriales especializadas como Pinecone o Milvus, estas suelen implicar el uso de servicios externos propietarios, costos asociados y una separación entre la base de datos relacional y la base de datos vectorial. En este proyecto se optó por utilizar *Supabase*, una plataforma de backend basada en PostgreSQL que integra de forma nativa la extensión *pgvector*, permitiendo el almacenamiento y consulta de embeddings dentro de una infraestructura unificada.
El uso de Supabase facilita la gestión conjunta de datos estructurados, representaciones vectoriales y metadatos, lo que posibilita la ejecución de consultas híbridas que combinan filtros tradicionales con búsquedas por similitud semántica. Este enfoque simplifica la arquitectura del sistema y reduce su complejidad operativa.
Desde el punto de vista económico y de control de la información, Supabase ofrece planes gratuitos y opciones de autoalojamiento, lo que lo hace especialmente

adecuado para proyectos académicos y de investigación. Además, no existía una exigencia técnica que obligara al uso de una base de datos vectorial dedicada para la integración con el modelo de lenguaje DeepSeek, ya que este actúa como componente de generación, mientras que la recuperación depende de la infraestructura de almacenamiento.

Aunque en escenarios de muy gran escala Supabase puede presentar limitaciones frente a soluciones vectoriales especializadas, los requerimientos y el volumen de datos del presente proyecto se encuentran dentro de las capacidades ofrecidas por PostgreSQL con *pgvector*.

2.10. Hostinger

El desarrollo del sistema no solo requirió una infraestructura adecuada para el almacenamiento y recuperación de la información, sino también un entorno que permitiera su despliegue y acceso mediante una interfaz gráfica. En este contexto, se utilizó el servicio de alojamiento web proporcionado por *Hostinger* para publicar la aplicación y permitir su acceso desde cualquier ubicación.

Hostinger ofrece un entorno de hosting que facilita el despliegue de aplicaciones web, permitiendo alojar tanto el frontend del sistema como los componentes necesarios para su comunicación con los servicios de backend. Esta característica resultó fundamental para exponer el sistema como una aplicación accesible a través de un navegador, eliminando la necesidad de configuraciones locales por parte de los usuarios.

3. DISEÑO DE SOLUCIÓN

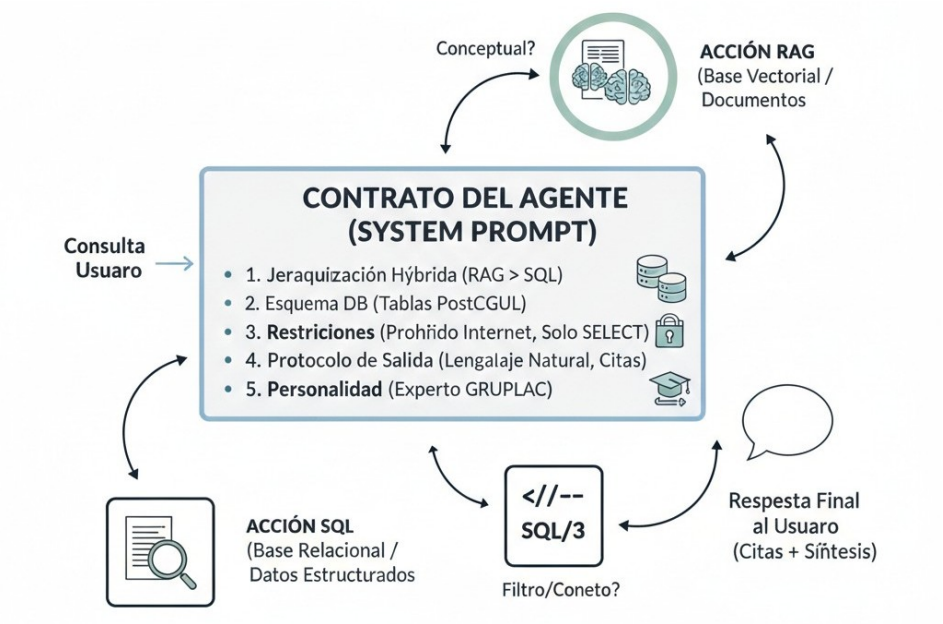
3.1. DESCRIPCIÓN GENERAL DEL AGENTE

El prototipo desarrollado permite consultar la información del grupo de investigación a través de un chatbot que actúa como interfaz principal. El usuario ingresa una pregunta y el sistema la procesa para determinar si corresponde a una consulta SQL o a una búsqueda de tipo semántico.

Si la consulta es clasificada como SQL, el sistema accede directamente a las tablas de la base de datos; si no lo es, realiza la búsqueda en la base vectorial. Con base en esta identificación, el prototipo recupera la información requerida y genera una respuesta clara para el usuario, integrando ambos tipos de consulta en un mismo flujo de interacción.

Figura 4. Arquitectura de procesamiento de consultas del chatbot basado en IA.

Figura 5. Arquitectura general del sistema de recuperación y generación de respuestas



3.1.1. Configuración de las instrucciones del agente Para asegurar un comportamiento coherente y controlado del chatbot, se implementó un *Contrato del Agente*, definido como un *System Prompt* que establece las reglas operativas que el modelo debe seguir antes de procesar las consultas del usuario. Este contrato actúa como un mecanismo de control que reduce alucinaciones y garantiza el uso exclusivo de las fuentes de información autorizadas.

Las instrucciones definen una lógica híbrida de consulta, priorizando la recuperación aumentada (RAG) para solicitudes conceptuales y limitando el uso de consultas SQL a operaciones de lectura sobre datos estructurados. Asimismo, se incluye el esquema de la base de datos para permitir la generación de consultas válidas y se imponen restricciones de seguridad que bloquean el acceso a fuentes externas y cualquier modificación de datos.

Finalmente, el contrato establece un protocolo de salida que exige respuestas en lenguaje natural y la citación de fuentes cuando la información proviene de la base vectorial, permitiendo que el sistema funcione como un asistente de consulta híbrido confiable y especializado.

3.2. PROCESAMIENTO Y PREPARACIÓN DE LOS DOCUMENTOS

Para iniciar el desarrollo del prototipo fue necesario definir el proceso mediante el cual se gestionaron los documentos no estructurados del grupo de investigación. Estos documentos —como trabajos de grado, artículos y publicaciones— debían ser transformados en representaciones vectoriales antes de ser integrados al sistema. En primera instancia, se creó una carpeta en Google Drive destinada exclusivamente al almacenamiento de estos documentos. Este repositorio funcionó como fuente centralizada desde la cual se tomarían los textos para su posterior procesamiento. A continuación, se desarrolló un flujo de trabajo en Google Colab que permitía automatizar las etapas necesarias para preparar los documentos. Desde este entorno se estableció la conexión con Google Drive y se implementó un script encargado de realizar las siguientes tareas:

- Carga de los documentos desde Drive.
- Segmentación (Chunking), para dividir cada archivo en fragmentos manejables.
- Generación de embeddings.
- Envío de los datos a Supabase.

3.3. DISEÑO DE LA BASE VECTORIAL

Una vez definidos y probados los mecanismos de procesamiento en Colab, se procedió a la construcción de la base de datos vectorial en Supabase. Esto incluyó

la creación de las tablas necesarias, la integración de la extensión pgvector y la configuración del tipo de dato vectorial para almacenar los embeddings generados previamente.
Con esto se estableció la infraestructura que permitiría realizar búsquedas por similitud y complementar las consultas del sistema más allá de lo que la información estructurada podía ofrecer.

3.4. DISEÑO DE LA BASE RELACIONAL

3.4.1. Metodología de Extracción y Transformación de Datos (ETL) La creación y el llenado de las tablas en Supabase no fue simplemente un proceso de carga masiva, sino que se llevó a cabo a través de un flujo de ETL (Extract, Transform, Load) diseñado para asegurar que la información científica mantuviera su integridad.

Web Scraping y Captura de Datos Crudos. La primera etapa implicó la extracción del código fuente HTML del portal GrupLAC. Utilizando nodos de solicitud HTTP y técnicas de procesamiento de texto, se identificaron y capturaron secciones específicas de interés, como “Producción bibliográfica” o “Trabajos dirigidos”. En este punto, la información estaba en un estado desestructurado y lleno de ruido, con etiquetas HTML mezcladas con el contenido textual de los registros.

Segmentación por Filas mediante Lógica de Código. Para convertir el gran bloque de HTML en datos más manejables, se utilizó un nodo de código con lógica en JavaScript.

- Segmentación (Parsing): Se aplicaron Expresiones Regulares (RegEx) avanzadas para identificar los patrones de las filas (<tr>...</tr>) dentro de las tablas de origen.

- Iteración Secuencial: Una vez que los datos fueron segmentados, se pasaron a un nodo Loop Over Items. Esta configuración es crucial, ya que permite procesar la información fila por fila, evitando problemas de memoria y asegurando que cada registro se maneje de manera individual antes de ser insertado en la base de datos.

Normalización y Limpieza mediante AI Agent. El paso más innovador de este flujo es la intervención de un AI Agent para la limpieza de los datos:

- Eliminación de Ruido: El Agente recibe el fragmento de HTML crudo de una sola fila y, a través de un Prompt Personalizado, extrae únicamente la información relevante (títulos, años, autores, categorías), desechando etiquetas y caracteres especiales innecesarios.
- Estructuración en JSON: La inteligencia artificial no solo se encarga de limpiar el texto, sino que también lo normaliza, proporcionando un objeto JSON con claves estandarizadas que se alinean perfectamente con las columnas de las tablas que hemos creado en Supabase.

Ingestión en Supabase. Una vez que los datos están limpios y estructurados, se envían al nodo PostgreSQL. Dado que la IA ya ha realizado la normalización, el nodo simplemente lleva a cabo una operación de inserción, mapeando los valores del JSON a sus columnas correspondientes en Supabase. Este enfoque asegura que cada tabla contenga información precisa, tipificada y libre de los errores comunes que suelen aparecer en el raspado web tradicional.

3.5. IDENTIFICACIÓN Y EXTRACCIÓN DE LA BASE DE DATOS

La segunda fase del proyecto consistió en determinar la fuente de datos más adecuada para alimentar el sistema, evitando la necesidad de construir manualmente

una base de datos desde cero. Para ello, se realizó un proceso de exploración y análisis de las plataformas disponibles, con el objetivo de identificar un repositorio confiable que centralizará la información necesaria para el trabajo.

Durante esta revisión se encontró que la información relevante ya estaba publicada en un recurso web oficial, el cual contenía datos estructurados sobre grupos de investigación, productos académicos y documentos relacionados. Una vez identificada esta fuente, se evaluó la posibilidad de extraer los datos directamente desde el sitio web.

El análisis incluyó verificar el formato en el que la página exponía la información, revisar si contaba con endpoints o APIs accesibles y determinar si la estructura del HTML permitía aplicar técnicas de web scraping. A partir de esta evaluación, se concluyó que era factible obtener los datos directamente desde la página mediante extracción programada, lo que evitó la necesidad de reconstruir o crear manualmente una base de datos local.

3.6. DIVISIÓN ESTRUCTURAL Y NO ESTRUCTURAL

Una vez obtenida la información desde las diferentes fuentes, se procedió a organizarla en dos grandes categorías con el fin de facilitar su procesamiento y su integración dentro del sistema: base de datos estructural y base de datos no estructural.

En primer lugar, se definió como base de datos estructural toda la información obtenida directamente desde la página web oficial identificada en la etapa anterior. Este conjunto de datos se caracteriza por contar con una estructura clara y consistente, lo que permitió extraer de forma organizada elementos como nombres de grupos de investigación, productos académicos, investigaciones registradas y metadatos asociados. Esta información proviene de un entorno controlado, con formato estable, lo que facilita su conversión posterior a texto procesable.

En contraste, la base de datos no estructural está conformada por documentos de

naturaleza heterogénea, principalmente trabajos de grado, artículos y otras publicaciones de texto libre. Estos documentos no comparten un formato único y presentan variaciones en su estructura interna, extensión y estilo de redacción. Para gestionar este conjunto, se creó una carpeta en Google Drive destinada exclusivamente al almacenamiento de estos archivos, con el fin de centralizar su acceso y procesamiento.

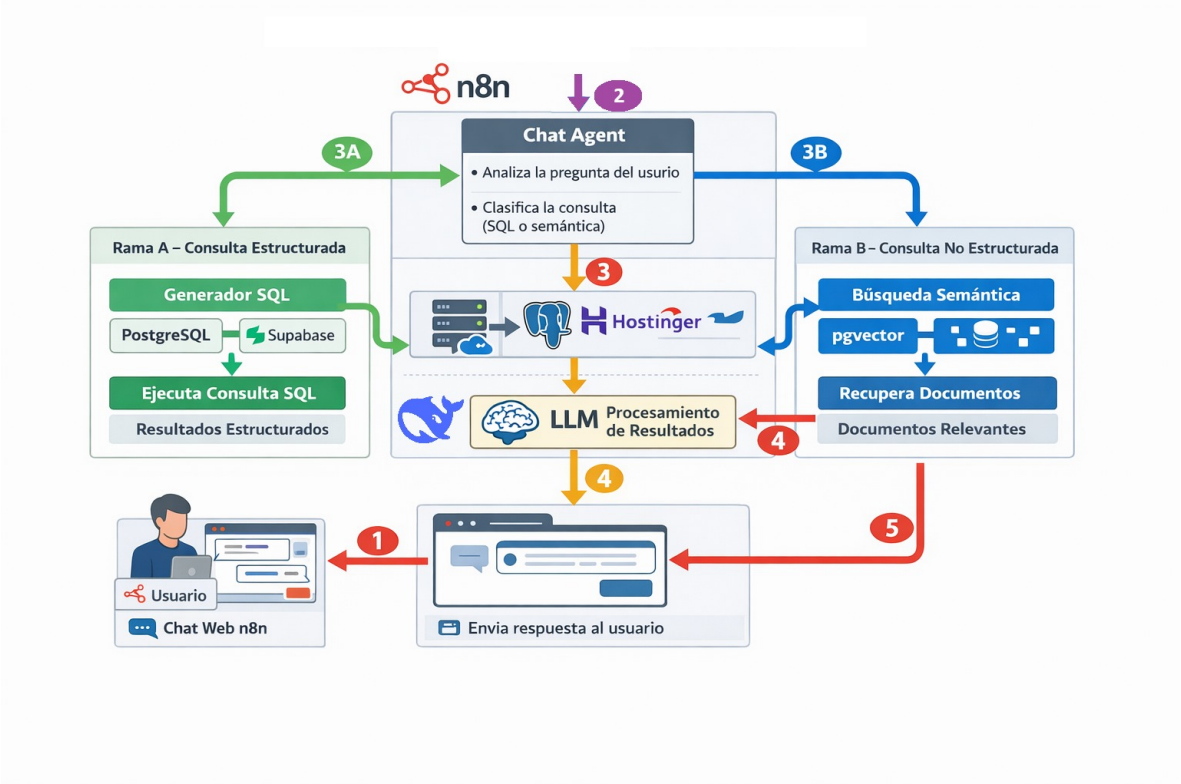
La diferenciación entre ambas bases de datos permitió aplicar técnicas específicas según el tipo de contenido: mientras la información estructural se integró mediante extracción automatizada desde la web, la información no estructurada fue sometida a procesos de chunkerización, generación de embeddings y carga en la base vectorial. Esta separación fue clave para garantizar que cada tipo de dato fuera tratado con la metodología más adecuada para su naturaleza.

3.7. ARQUITECTURA GENERAL DEL SISTEMA

Una vez conformada la base de datos y separada en los dos componentes principales estructural y no estructural se avanzó en el diseño del agente encargado de interpretar las consultas realizadas por los usuarios. En esta etapa, el objetivo principal fue definir las características y reglas necesarias para que el sistema pudiera distinguir entre consultas de tipo SQL y consultas de tipo no estructurado, permitiendo así seleccionar el mecanismo de búsqueda adecuado.

Para ello, se analizaron los patrones más comunes en las preguntas que podrían realizar los usuarios, teniendo en cuenta el tipo de información disponible en cada una de las dos bases. Se establecieron criterios para que el agente identificara comandos, términos o estructuras asociadas a consultas SQL, como solicitudes de filtrado, conteo, selección de columnas o condiciones específicas. De esta manera, cuando una consulta presentaba características propias de una instrucción estructurada, el sistema la dirigía hacia la base de datos relacional.

Figura 6. Arquitectura general del sistema de recuperación y generación de respuestas



Por otra parte, cuando el usuario formulaba preguntas abiertas, descriptivas o contextuales, el agente debía clasificarlas como consultas no estructuradas. En estos casos, el sistema empleaba la base vectorial, que contenía los embeddings generados a partir de los documentos no estructurados, permitiendo una búsqueda semántica más flexible y cercana al lenguaje natural.

3.8. DESARROLLO DE LA INTERFAZ WEB

La etapa final del proyecto consistió en el diseño y construcción de la página web desde la cual se integrará y operará el chatbot. El objetivo de esta fase fue crear un entorno accesible y práctico que permitiera a los usuarios interactuar directamente con el sistema desarrollado.

Para ello, se definió una interfaz sencilla y funcional donde el usuario pudiera escribir sus consultas y recibir respuestas de manera inmediata. La página web se configuró para comunicarse con el backend del prototipo, enviando cada consulta al agente encargado de clasificar y procesar, y posteriormente mostrando la respuesta correspondiente al usuario.

3.9. INTEGRACIÓN DE PLATAFORMAS

Para el desarrollo del proyecto se empleó un conjunto de herramientas que permitieron gestionar documentos, procesar información, almacenar datos y construir la lógica del sistema. A continuación, se describen las principales plataformas utilizadas:

- Google Drive: Se utilizó como repositorio para almacenar los documentos no estructurados, incluyendo trabajos de grado, artículos y publicaciones. Desde esta ubicación se tomaron los archivos que serían procesados y convertidos en embeddings.

- Google Colab: En Colab se ejecutó el código encargado de cargar los documentos desde Google Drive, realizar la segmentación o chunking, generar los embeddings y enviarlos a la base de datos vectorial en Supabase. Este entorno facilitó la automatización del procesamiento y su integración con otras herramientas.

- Supabase: Actuó como la plataforma principal de almacenamiento. Allí se gestionaron tanto la base de datos estructural como la base vectorial. Supabase permitió organizar la información, almacenar los embeddings y acceder a ellos mediante consultas semánticas y SQL.

- n8n: Se empleó para construir el backend del sistema mediante flujos automatizados. En esta plataforma se integraron las consultas del chatbot, la clasificación entre SQL y noSQL, y las interacciones con Supabase y el modelo de lenguaje.

- API de DeepSeek: Proporcionó el acceso al modelo de lenguaje utilizado en el sistema. A través de esta API fue posible seleccionar el modelo, enviarle las consultas procesadas y recibir las respuestas que posteriormente se entregarán al usuario.

- Easy panel: Se utilizó como plataforma de administración y despliegue de servicios, permitiendo gestionar de forma centralizada las aplicaciones alojadas en el VPS.

- Hostinger: Se empleó como proveedor del VPS, proporcionando la infraestructura en la nube necesaria para alojar y ejecutar los servicios del proyecto.

4. VALIDACIÓN DE LA SOLUCIÓN

Para la validación del sistema se formularon preguntas de tipo SQL y noSQL, con el fin de evaluar la capacidad del chatbot para recuperar y generar respuestas de manera precisa y coherente. Las preguntas de tipo SQL se utilizaron para validar la búsqueda de información almacenada en bases de datos estructuradas, mientras que las preguntas no SQL se diseñaron para evaluar la recuperación de información representada mediante embeddings.

4.1. VALIDACIÓN DEL SISTEMA

Durante el proceso de evaluación, cada pregunta realizada y la respuesta generada por el chatbot fueron registradas. Posteriormente, se estableció de forma manual la respuesta considerada correcta para cada consulta, la cual sirvió como referencia. Con base en esta comparación, las respuestas del chatbot fueron calificadas como correctas o incorrectas, permitiendo así medir el desempeño del sistema en ambos tipos de consultas.

Los resultados de la Tabla 1 indican que la mayoría de las respuestas generadas por el chatbot fueron clasificadas como correctas. Las respuestas que obtuvieron una clasificación incorrecta corresponden principalmente a consultas de tipo SQL relacionadas con fechas. Esto se debe a que, en la tabla donde se almacena dicha información, las fechas no se encuentran completamente especificadas, lo que genera ambigüedades al momento de realizar comparaciones temporales.

Este comportamiento se evidencia en la consulta “¿Cuál es el trabajo de grado más reciente dirigido por Ana Ramírez Silva?”. Tanto la respuesta generada por el chatbot como la establecida manualmente como respuesta correcta presentan la misma fecha: “desde 1 de 2024 hasta enero”. Ante esta igualdad temporal, se consideró

Tabla 1. Resultados de la validación de la solución.

Pregunta	Respuesta del bot	Respuesta correcta	Valoracion
Nombre de cinco integrantes del grupo	Carlos Augusto Fajardo Ariza, Said David Pertuz .	Carlos Augusto Fajardo Ariza, Said David P	correcta
Nombre de los dos integrantes mas antiguo	Julio Cibel Caro Torres, Oscar Gualdron Gonzalez	Julio Cibel Caro Torres, Oscar Gualdron Go	correcta
¿En cuantos articulos publicados figura sai	Said Pertuz figura como autor en 26 artículos pub	26 articulos publicados	correcta
¿En cuantos eventos cientificos ha particip	El grupo ha participado en 101 eventos científicos	El grupo ha participado en 101 eventos	correcta
¿Cual fue el primer evento cinetifico y el ult	Primer evento cinetifico: no tiene informacion espe	Primer evento cientifico: imposio : X Simpos	incorrecta
¿Cuantos trabajos de grado ha sido tutora	Ana Ramirez Silva ha sido tutora o cotutora de 69	Ana Ramirez ha sido turora o cotutora en 6	correcta
¿Cual es el trabajo de grado de pregrado n	Diseño de una aplicación móvil para la visualizaci	Diseño e implementación de módulos para	incorrecta
¿Cuantos trabajos de grado ha dirigido el g	El grupo ha dirigido un total de 404 trabajos de gr	El grupo ha dirigido 404 trabajos de grado	correcta
¿Cuantos trabajos de grado tienen valorac	Hay 365 trabajos de grado que tienen valoración	365 trabajos de grado tienen valoracion apr	correcta
¿Cuantos trabajos de grado estan aprobad	Trabajos de grado aprobados por nivel académico	Trabajos aprobados de pregrado 276, Maes	correcta
¿Quien es el autor de DISEÑO DE UNA TE	Basándome en la información recuperada de los	Andrés Felipe Ramírez Silva	correcta
¿Cual es el titulo del trabajo de grado de Ju	Título del trabajo de grado: "Medición del desemp	Basándome en la información recuperada d	correcta
¿Quien es el director del trabajo de grado	MIE. Dorfell Leonardo Parra Prada	MIE. Dorfell Leonardo Parra Prada	correcta
¿Cual es el objetivo general del trabajo de	Diseñar y adaptar módulos de SeisComP para el	Diseñar y adaptar módulos de SeisComP pa	correcta
Extrae el resumen existente del documento	La exploración sísmica produce gran cantidad de	La exploración sísmica produce gran cantid	correcta
Dame el nombre de 3 trabajos de grado qu	"Efectos de la estandarización de imágenes mam	"Efectos de la estandarización de imágenes	correcta

como el trabajo de grado más reciente “Diseño e implementación de módulos para Seiscomp para el estudio de sismos en Santander usando la Red Sismológica REDNE”, dado que fue el último registro almacenado en la base de datos. Bajo este criterio, la respuesta del chatbot podría considerarse válida; sin embargo, fue clasificada como incorrecta debido a las limitaciones en la estructuración de la información de fechas.

La Tabla 2 presenta los tiempos de respuesta, medidos en segundos, obtenidos por siete participantes y por el chatbot para un conjunto de quince consultas realizadas sobre la misma fuente de información. Para cada consulta se registró el tiempo individual de cada usuario, el tiempo promedio y el tiempo correspondiente al chatbot.

De manera general, los resultados evidencian que el chatbot presenta tiempos de respuesta significativamente menores en comparación con los usuarios. Mientras que el tiempo promedio total de los participantes fue de 1.118,47 segundos, el tiempo total empleado por el chatbot fue de 316,58 segundos, lo que representa una reducción aproximada del 71,7 % en el tiempo de búsqueda.

Al analizar los resultados consulta por consulta, se observa que en la mayoría de los

Tabla 2. Comparación de tiempos de respuesta entre participantes humanos y el chatbot.

	Consulta	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6	Usuario 7	Promedio	Chatbot
1	¿Quienes son lo	340,99	12,26	75,95	32,44	24,38	59,43	28,47	81,99	12,31
2	¿Cual es el nom	254,5	69,95	186,4	400,74	103,48	156,11	56,14	175,33	6,96
3	¿Que tipo de vin	23,83	10,07	49,89	36,69	36,93	85,24	30,5	39,02	34,18
4	¿Que tipo de val	136,08	27,78	55,94	32,41	30,78	21,48	36	48,64	13,57
5	¿Cuantos articul	104,62	108,76	138,78	212,93	195	115,6	86,27	137,36	7,47
6	¿En cuantos trab	205,7	197,13	185,94	247,95	259,05	100,27	79,39	182,20	7,02
7	¿Cuantos event	70,09	13,77	38,69	67,48	34,35	22,72	46,62	41,96	8,11
8	¿Cuál fue el prim	7,81	47,66	19,08	43,94	75,33	29,96	94,33	45,44	17,55
9	¿Cuál fue el últin	109,91	28,24	67,58	32,15	34,45	73,47	84,67	61,50	13,4
10	¿Cual es el traba	64,98	19,4	51,9	36,31	60,37	26,94	43,99	43,41	14,59
11	¿Que rol desemp	89,89	42,05	11,03	36,03	71,87	7,39	38,95	42,46	57,23
12	¿Quien fue el au	49,41	6,18	19,08	40,48	24,6	19,6	33,1	27,49	13,33
13	Nombre tres trab	93,74	67,87	84,08	94,82	127,75	41,22	81,15	84,38	18,43
14	¿Cuál es el título	28,59	14,31	48,62	20,02	22,86	34,84	35,89	29,30	40,91
15	¿Qué trabajos d	38,58	82,23	20,83	111,05	74,55	22,59	196,08	77,99	51,52
Tiempo total consulta		1.618,72	747,66	1.053,79	1.445,44	1.175,30	816,86	971,55	1.118,47	316,58

casos el chatbot responde en menos de 15 segundos, incluso en consultas donde los tiempos de los participantes superan ampliamente el minuto. Por ejemplo, en la Consulta 1, el tiempo promedio fue de 81,99 segundos, mientras que el chatbot resolvió la misma consulta en 12,31 segundos. De forma similar, en la Consulta 2, el promedio alcanzó 175,33 segundos, frente a los 6,96 segundos del chatbot.

No obstante, se identifican algunas excepciones. En la Consulta 11, el chatbot registró un tiempo de 57,23 segundos, superior al promedio de 42,46 segundos, lo que indica que, aunque el chatbot es generalmente más rápido, su desempeño puede verse afectado por la complejidad semántica de ciertas preguntas o por la forma en que la información se encuentra representada internamente.

Los tiempos registrados por los participantes presentan una alta variabilidad, tanto entre diferentes usuarios como entre distintas consultas. Esto se refleja en diferencias significativas entre los valores mínimos y máximos para una misma consulta. Por ejemplo, en la Consulta 2, los tiempos de los participantes oscilan entre 56,14 segundos y 400,74 segundos, lo que evidencia la influencia de factores como la experiencia del usuario, la familiaridad con la plataforma y la estrategia de búsqueda

utilizada.

Los resultados demuestran que el uso del chatbot permite optimizar significativamente el tiempo de recuperación de información frente a la búsqueda manual realizada por usuarios. Esta diferencia se hace más evidente en consultas que requieren navegar por múltiples secciones de la página web o interpretar información distribuida en distintos apartados. Sin embargo, también se observa que el rendimiento del chatbot puede verse afectado en consultas específicas, lo que resalta la importancia de una adecuada estructuración de la información y del modelo de recuperación empleado.

Tabla 3. Evaluación de la precisión de las respuestas de los participantes y el chatbot.

Consulta	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6	Usuario 7	Chatbot
1	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
2	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
3	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
4	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
5	incorrecta	correcta	correcta	correcta	incorrecta	incorrecta	incorrecta	incorrecta
6	correcta	correcta	incorrecta	correcta	incorrecta	incorrecta	correcta	correcta
7	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
8	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
9	correcta	correcta	correcta	correcta	correcta	parcial	correcta	parcial
10	correcta	correcta	correcta	correcta	correcta	correcta	correcta	incorrecta
11	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
12	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
13	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
14	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta
15	correcta	correcta	correcta	correcta	correcta	correcta	correcta	correcta

La Tabla 3 presenta la evaluación de la precisión de las respuestas obtenidas por los participantes humanos y por el chatbot para un conjunto de quince consultas. Las respuestas fueron clasificadas como correctas, incorrectas o parciales, según su correspondencia con la respuesta establecida como referencia.

Se observa un alto nivel de precisión tanto en los usuarios humanos como en el

chatbot. La mayoría de las consultas fueron respondidas correctamente por todos los participantes, lo que indica que la información era accesible y comprensible a través de la fuente consultada.

Las respuestas clasificadas como incorrectas o parciales se concentran en un número reducido de consultas, principalmente en las consultas 5, 6, 9 y 10. En estos casos, los errores se presentan tanto en usuarios humanos como en el chatbot, lo que sugiere que las dificultades están asociadas a la ambigüedad de la información o a la forma en que esta se encuentra estructurada, más que a una limitación específica del sistema.

5. CONCLUSIONES

En relación con el objetivo de diseñar e implementar un agente inteligente capaz de interpretar consultas en lenguaje natural, los resultados obtenidos demuestran que el prototipo desarrollado cumple satisfactoriamente con esta finalidad. El sistema logró analizar las solicitudes de los usuarios y clasificarlas correctamente como consultas de tipo SQL o NoSQL, permitiendo su direccionamiento hacia la fuente de información correspondiente. Este comportamiento confirma que la integración de modelos de lenguaje de gran escala con mecanismos de razonamiento semántico constituye una solución viable para facilitar el acceso a información académica compleja sin requerir conocimientos técnicos por parte del usuario final.

Respecto al objetivo de integrar información estructurada y no estructurada mediante una arquitectura híbrida, se evidenció que la combinación de bases de datos relacionales y bases de datos vectoriales permite una recuperación de información más completa y flexible. La utilización de consultas SQL para datos estructurados, junto con búsquedas semánticas sobre documentos procesados mediante embeddings, posibilitó el acceso contextualizado a distintos tipos de contenido del grupo de investigación CPS. Esto valida la pertinencia del enfoque híbrido propuesto y demuestra su capacidad para gestionar de manera eficiente información heterogénea distribuida en múltiples formatos y repositorios.

En cuanto al objetivo de evaluar el desempeño del sistema en términos de eficiencia y precisión, los resultados experimentales evidencian una mejora significativa frente a los métodos tradicionales de búsqueda manual. La reducción del 71,7 % en el tiempo total de consulta, comparando el desempeño del chatbot con el promedio de los participantes, confirma que el agente inteligente optimiza sustancialmente los procesos de recuperación de información. Asimismo, la alta proporción de respuestas clasificadas como correctas demuestra que el sistema no solo es más rápido,

sino también confiable para apoyar tareas de consulta académica y administrativa. En relación con el objetivo de analizar las limitaciones del sistema, se identificó que los casos de respuestas incorrectas estuvieron principalmente asociados a deficiencias en la estructura y actualización de los datos, especialmente en consultas relacionadas con información temporal. Este hallazgo permite concluir que el desempeño del agente inteligente depende en gran medida de la calidad de las fuentes de datos disponibles, y resalta la necesidad de implementar estrategias de normalización y mantenimiento de la información para maximizar la precisión del sistema. En conjunto, estas conclusiones confirman que el prototipo desarrollado cumple los objetivos propuestos y establece una base sólida para futuras mejoras y ampliaciones del sistema.

6. RECOMENDACIONES

A partir de la experiencia adquirida durante el diseño, implementación y validación del sistema conversacional, se identifican diversas oportunidades de mejora que pueden orientar trabajos futuros en este ámbito. En primer lugar, se recomienda fortalecer los mecanismos de organización y actualización de la información almacenada en las bases de datos estructuradas, especialmente en campos críticos como fechas y metadatos. Una gestión más rigurosa de estos elementos permitiría reducir ambigüedades en las consultas de tipo SQL y mejorar la precisión de las respuestas generadas por el sistema.

Adicionalmente, se sugiere profundizar en la optimización del proceso de clasificación de consultas entre SQL y NoSQL, con el fin de manejar de manera más eficiente consultas complejas o ambiguas. La incorporación de técnicas más avanzadas de análisis semántico podría contribuir a una mejor asignación de las consultas al tipo de base de datos correspondiente, mejorando así tanto el desempeño como la coherencia de las respuestas del chatbot.

En cuanto al proceso de evaluación, resulta recomendable automatizar parcialmente las tareas de validación del sistema, incorporando herramientas que permitan registrar, analizar y comparar de manera sistemática los tiempos de respuesta y la precisión de las respuestas. Esto no solo reduciría la carga de trabajo manual, sino que también facilitaría la repetición de los experimentos y la obtención de métricas más consistentes y objetivas.

Finalmente, se recomienda continuar evaluando y actualizando los modelos de lenguaje y las tecnologías empleadas, considerando la rápida evolución del campo de la inteligencia artificial. La revisión periódica de nuevas arquitecturas, técnicas de recuperación de información y modelos emergentes permitiría mejorar la escalabilidad, eficiencia y robustez del sistema, garantizando su vigencia y aplicabilidad en

escenarios reales de consulta.

BIBLIOGRAFÍA

Aggarwal, Charu C. *Data Mining: The Textbook*. Springer, 2015 (vid. pág. 25).

Hearst, Marti A. «TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages». En: *Computational Linguistics* 23.1 (1997), págs. 33-64 (vid. pág. 23).

IBM. *What are vector databases?* s.f. URL: https://www.ibm.com/think/topics/vector-database (visitado 15-01-2025).

— *What is an embedding?* s.f. URL: https://www.ibm.com/think/topics/embedding (visitado 15-01-2025).

— *What is artificial intelligence?* s.f. URL: https://www.ibm.com/think/topics/artificial-intelligence (visitado 15-01-2025) (vid. pág. 14).

— *What is natural language processing (NLP)?* s.f. URL: https://www.ibm.com/think/topics/natural-language-processing (visitado 15-01-2025) (vid. pág. 19).

Jurafsky, Daniel y James H. Martin. *Speech and Language Processing*. 3.ª ed. Pearson, 2023 (vid. págs. 14, 18).

Lewis, Patrick et al. «Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks». En: *Advances in Neural Information Processing Systems* 33 (2020), págs. 9459-9474 (vid. págs. 15, 23).

Manning, Christopher D., Prabhakar Raghavan e Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008 (vid. págs. 15, 19, 23, 24).

Mikolov, Tomas et al. «Efficient Estimation of Word Representations in Vector Space». En: *arXiv preprint arXiv:1301.3781* (2013).

Reimers, Nils e Iryna Gurevych. «Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks». En: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 2019 (vid. pág. 15).

Russell, Stuart J. y Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4.ª ed. Hoboken, NJ: Pearson, 2021 (vid. pág. 18).

Turney, Peter D. y Patrick Pantel. «From Frequency to Meaning: Vector Space Models of Semantics». En: *Journal of Artificial Intelligence Research* 37 (2010), págs. 141-188 (vid. pág. 24).