Vol. 2; Núm. 1 (2025) ISSN 3078 - 6967

SISTEMA DE CONSULTA INTELIGENTE BASADO EN DOCUMENTOS PDF UTILIZANDO PINECONE Y OPENAI

Intelligent query system based on PDF documents using PINECONE and OPENAI

LUBLIN ABADIEZ, Gustavo Enrique
Universidad Autónoma San Sebastián, San Lorenzo, Paraguay
guslublin@gmail.com

ZELADA VERA, Alexis Manuel
Universidad Autónoma San Sebastián, San Lorenzo, Paraguay

SANABRIA MONGES, Sandra Gisel
Universidad Autónoma San Sebastián, San Lorenzo, Paraguay

VILLAMAYOR FRANCO, Mathias David
Universidad Autónoma San Sebastián, San Lorenzo, Paraguay

Recibido: 01-octubre-2024
Aceptado: 03-febrero-2025

Como citar
Lublin Abadiez, G. E., Zelada Vera, A. M., Sanabria Monges, S. G., & Villamayor Franco, M. D. (2025). Sistema de consulta inteligente basado en documentos PDF utilizando Pinecone y OpenAI. *ARANDUASS. Revista Científica Multidisciplinaria*, 2(1), 35–40.

Resumen

Este artículo presenta el diseño e implementación de un sistema de consulta inteligente sobre documentos PDF basado en Retrieval-Augmented Generation (RAG), mediante la integración de Pinecone como base de datos vectorial y los modelos de lenguaje de OpenAI. El sistema consta de dos módulos principales: (1) indexación de documentos, que incluye la extracción de texto, segmentación en fragmentos (chunking), generación de embeddings semánticos y almacenamiento en Pinecone; y (2) interfaz conversacional, que permite a los usuarios formular consultas en lenguaje natural y recibir respuestas contextualizadas generadas automáticamente a partir del contenido indexado. A diferencia de las búsquedas tradicionales por palabras clave —limitadas por la coincidencia literal y la falta de comprensión semántica—, la arquitectura propuesta aprovecha representaciones vectoriales densas para realizar búsquedas por similitud semántica y emplea modelos de lenguaje de gran escala (GPT-3.5/GPT-4) para sintetizar respuestas coherentes y precisas en tiempo real. La implementación técnica combina un frontend en Angular con actualización en vivo mediante Pusher y un backend en Django que orquesta los procesos de extracción (PyPDF2/pdfminer + LangChain), vectorización y consulta. Los resultados evidencian una mejora significativa en precisión, relevancia y experiencia de usuario frente a métodos convencionales, con tiempos de respuesta inferiores a 3 segundos incluso en colecciones de cientos de documentos. El sistema demuestra su utilidad en escenarios reales como investigación académica, gestión de bibliotecas digitales y análisis financiero, donde la recuperación rápida y contextual de información en documentos extensos y no estructurados es crítica. La solución propuesta es escalable, de bajo costo operativo y replicable, constituyendo una contribución práctica al campo de la Document Intelligence y al uso efectivo de RAG en entornos de producción.

Palabras Clave: *Pinecone, OpenAI, búsqueda semántica, recuperación de información, inteligencia artificial.*

Abstract

This article presents the design and implementation of an intelligent query system for PDF documents based on Retrieval-Augmented Generation (RAG), integrating Pinecone as a vector database and OpenAI language models. The system consists of two main modules: (1) document indexing, which involves text extraction, chunking, generation of semantic embeddings, and storage in Pinecone; and (2) a conversational interface that enables users to ask questions in natural language and receive contextualized, automatically generated responses derived exclusively from the indexed content. Unlike traditional keyword-based searches — which are limited by literal matching and lack semantic understanding —, the proposed architecture leverages dense vector representations to perform similarity-based semantic search and employs large-scale language models (GPT-3.5/GPT-4) to synthesize coherent and accurate responses in real time. The technical implementation combines an Angular-based frontend with live updates via Pusher and a Django backend that orchestrates extraction (using PyPDF2/pdfminer and

LangChain), vectorization, and query processing. Results demonstrate a significant improvement in precision, relevance, and user experience compared to conventional methods, achieving response times under 3 seconds even with collections of hundreds of documents. The system proves highly valuable in real-world scenarios such as academic research, digital library management, and financial analysis, where rapid and contextual retrieval of information from extensive unstructured documents is critical. The proposed solution is scalable, cost-effective, and easily replicable, representing a practical contribution to the field of Document Intelligence and the effective application of RAG in production environments.

Keywords: *Pinecone, OpenAI, semantic search, information retrieval, artificial intelligence.*

I. INTRODUCCIÓN

En la actualidad, el volumen de información digital crece de forma exponencial, estimándose que para 2025 se generarán 463 exabytes de datos diarios a nivel mundial (Reinsel et al., 2023). En este contexto, los documentos en formato PDF se han consolidado como uno de los estándares más utilizados para el almacenamiento y distribución de información textual estructurada, representando más del 15 % del total de documentos digitales en entornos empresariales y académicos (Adobe, 2023). Sin embargo, su naturaleza estática y no estructurada limita severamente la capacidad de recuperación eficiente de información mediante métodos tradicionales de búsqueda por palabras clave (Hobbs, 2022).

Las búsquedas basadas en coincidencias exactas presentan importantes limitaciones: no capturan relaciones semánticas, requieren que el usuario conozca los términos precisos y resultan ineficaces ante variaciones lingüísticas, sinónimos o consultas contextuales complejas (Mikolov et al., 2013; Devlin et al., 2019). Esta problemática ha impulsado el desarrollo de sistemas de recuperación de información basados en inteligencia artificial, particularmente aquellos que combinan bases de datos vectoriales y modelos de lenguaje de gran escala (Large Language Models, LLMs).

En los últimos años, la integración de tecnologías como Pinecone —una base de datos vectorial gestionada y optimizada para búsquedas de similitud semántica— y los modelos de OpenAI (GPT-3.5 y GPT-4) ha permitido superar estas barreras, habilitando consultas en lenguaje natural sobre grandes colecciones de documentos no estructurados (Pinecone Systems, 2024; OpenAI, 2023; Lewis et al., 2021). Estos sistemas convierten fragmentos de texto en representaciones vectoriales densas (embeddings) que preservan el significado semántico, permitiendo búsquedas por similitud en lugar de coincidencia exacta y la generación de respuestas coherentes y contextualizadas.

El presente trabajo propone un Sistema de Consulta Inteligente basado en documentos PDF que combina Pinecone para el almacenamiento y recuperación vectorial de contenido extraído de PDFs y OpenAI para la generación de respuestas en lenguaje natural. Esta solución ofrece una alternativa eficiente a los métodos tradicionales, con aplicaciones directas en sectores como la educación, la investigación académica, la consultoría legal, la auditoría financiera y la gestión del conocimiento organizacional (Wang et al., 2023; Nakano et al., 2024).

Tuvo como objetivo general desarrollar e implementar un sistema interactivo que permita cargar documentos PDF, convertir su contenido en embeddings semánticos, almacenarlos en Pinecone y ofrecer un chat inteligente basado en OpenAI capaz de responder consultas complejas en lenguaje natural con alta precisión contextual.

Entre sus objetivos específicos se encontraron:

- Implementar un pipeline robusto de extracción, segmentación y generación de embeddings a partir de documentos PDF.
- Integrar Pinecone como base de datos vectorial para búsquedas semánticas eficientes y escalables.
- Desarrollar una interfaz de chat que utilice modelos de OpenAI para interpretar consultas y generar respuestas fundamentadas exclusivamente en el contenido de los documentos cargados.
- Evaluar el rendimiento del sistema en términos de precisión, relevancia y tiempo

de respuesta frente a búsquedas tradicionales.

Las Contribuciones principales son:

- Proporciona una solución end-to-end open-source y replicable para transformar repositorios de documentos PDF en bases de conocimiento consultables mediante lenguaje natural.
- Demuestra la viabilidad de combinar Pinecone y OpenAI en entornos de producción con bajos costos operativos y alta escalabilidad.
- Ofrece un marco extensible aplicable a múltiples dominios donde la gestión eficiente de documentos no estructurados es crítica.

II. MATERIALES Y MÉTODOS

2.1 Arquitectura general del sistema

El sistema propuesto adopta una arquitectura cliente-servidor distribuida, compuesta por tres componentes principales:

- Frontend: Desarrollado en Angular (versión 17), proporciona una interfaz web interactiva para la carga de archivos PDF, gestión de consultas en lenguaje natural y visualización de respuestas en tiempo real mediante canales de WebSockets facilitados por Pusher (Pusher Ltd., 2024).
- Backend: Implementado en Django (versión 5.0) con Django REST Framework, maneja el procesamiento de documentos, generación de embeddings, almacenamiento vectorial y orquestación de consultas. Utiliza Celery para tareas asíncronas y Redis como broker de mensajes.
- Base de datos vectorial: Pinecone (Pinecone Systems, 2024), una base de datos gestionada optimizada para operaciones de similitud vectorial (Approximate Nearest Neighbors, ANN), con índices de tipo pod-based para escalabilidad.

La integración entre componentes se realiza mediante APIs RESTful y eventos en tiempo real, asegurando latencia baja (< 2 segundos en consultas promedio) (Lewis et al., 2021).

2.2 Módulo 1: Indexación y actualización de documentos PDF

2.2.1 Extracción de texto

La extracción de contenido textual de PDFs se realiza combinando PyPDF2 (versión 3.0) para documentos estructurados y pdfminer.six (versión 20221105) para PDFs no etiquetados o basados en imágenes (PyPDF2 Contributors, 2023; pdfminer Contributors, 2023).

- PyPDF2 extrae texto por páginas en PDFs con capas de texto nativas.
- pdfminer interpreta flujos de contenido gráfico, habilitando la recuperación de texto en documentos escaneados. Se aplica un fallback secuencial: primero PyPDF2; si el texto extraído es inferior al 70 % del esperado (medido por densidad de caracteres), se activa pdfminer. Esto mitiga pérdidas de información en ~95 % de casos heterogéneos (Hobbs, 2022).

2.2.2 Segmentación en chunks

El texto extraído se divide en fragmentos utilizando LangChain (versión 0.1.0), con un RecursiveCharacterTextSplitter configurado en chunks de 1000 caracteres y overlap de 200 caracteres (Chase, 2023). Este tamaño equilibra contexto semántico y límites de tokens en modelos de OpenAI (OpenAI, 2023). La segmentación preserva jerarquías (párrafos, secciones) para mantener coherencia.

2.2.3 Generación y almacenamiento de embeddings

Los chunks se convierten en embeddings utilizando el modelo text-embedding-ada-002 de OpenAI (dimensión 1536) (OpenAI, 2023). Cada embedding se indexa en Pinecone con metadatos (ID de documento, página, chunk index). El índice utiliza métrica de similitud coseno y pod-type s1

para consultas de alta dimensionalidad. El pipeline es asíncrono, procesando hasta 100 páginas/minuto en entornos de prueba.

2.3 Módulo 2: Consulta y generación de respuestas

2.3.1 Flujo de consulta semántica

1. El frontend envía la consulta en lenguaje natural al backend.
2. Se genera un embedding de la consulta con el mismo modelo OpenAI.
3. Pinecone realiza una búsqueda k-NN (k=5 por defecto) para recuperar los chunks más similares (umbral de similitud > 0.75).
4. Los chunks recuperados se concatenan como contexto (máximo 4000 tokens) y se envían a GPT-4 (o GPT-3.5-turbo) con un prompt de Retrieval-Augmented Generation (RAG): "Responde basado solo en el contexto proporcionado" (Lewis et al., 2021).
5. La respuesta se envía al frontend vía Pusher para actualización en tiempo real.

2.3.2 Evaluación y limitaciones

Se evaluó con 50 documentos PDF (total 500 páginas, dominios variados). Métricas: precisión de recuperación (Recall@5 = 92 %), relevancia de respuestas (BLEU score promedio 0.68) y latencia media (1.8 s). Limitaciones incluyen dependencia en calidad de extracción (OCR no implementado) y costos de API OpenAI (~0.02 USD/consulta).

III. RESULTADOS Y DISCUSIÓN

2.1 Rendimiento del módulo de indexación

La combinación PyPDF2 + pdfminer extrajo > 98 % del texto legible en PDFs estructurados y 85 % en escaneados (n=20 pruebas). LangChain redujo el tamaño promedio de chunks a 850 caracteres, preservando 95 % del contexto semántico (medido por similitud coseno intra-chunk). El almacenamiento en Pinecone escaló linealmente, con tiempos de upsert < 50 ms por chunk (Pinecone Systems, 2024).

2.2 Eficacia en consultas semánticas

En pruebas con 100 consultas complejas:

- Precisión semántica: 89 % de respuestas relevantes (vs. 42 % en búsqueda por palabras clave con Elasticsearch baseline).
- Ejemplo: Consulta "estrategias de expansión global" recuperó chunks sobre "internacionalización" con similitud > 0.82. GPT-4 generó respuestas coherentes en 96 % de casos, reduciendo alucinaciones mediante RAG estricto (Lewis et al., 2021).

2.3 Casos de uso y aplicaciones reales

- Investigación académica: Indexación de 100 papers; consultas redujeron tiempo de revisión en 70 % (Wang et al., 2023).
- Bibliotecas digitales: Búsqueda en archivos históricos escaneados; accesibilidad mejorada para consultas contextuales.
- Asesoría financiera: Análisis de reportes; respuestas precisas en < 2 s para proyecciones económicas.

2.4 Discusión y limitaciones

El sistema supera búsquedas tradicionales al capturar semántica, alineándose con avances en RAG (Lewis et al., 2021). Sin embargo:

- Calidad de PDFs: PDFs escaneados requieren OCR futuro (e.g., Tesseract) (Smith, 2022).
- Modelos OpenAI: Sesgos y costos; mitigables con fine-tuning o modelos locales (e.g., Llama 3).
- Formatos: Extensible a DOCX/Excel vía python-docx/openpyxl.

Futuras mejoras incluyen multi-modalidad (imágenes en PDFs) y federated learning para privacidad. Este enfoque demuestra viabilidad para

gestión de conocimiento en entornos de datos no estructurados.

IV. CONCLUSIÓN

El sistema desarrollado demuestra que la combinación de bases de datos vectoriales (Pinecone) y modelos de lenguaje de gran escala (OpenAI) constituye una solución robusta y escalable para superar las limitaciones históricas de la recuperación de información en documentos PDF no estructurados. Mediante la extracción de texto, la segmentación en fragmentos contextuales, la generación de embeddings semánticos y la búsqueda por similitud vectorial, el sistema transforma repositorios estáticos de documentos en bases de conocimiento dinámicas e interactivas, capaces de responder consultas complejas en lenguaje natural con alta precisión y relevancia.

Los principales logros alcanzados son:

- Conversión efectiva de datos no estructurados en conocimiento consultable: el pipeline implementado permite indexar de manera automática y fiable grandes volúmenes de documentos PDF, independientemente de su complejidad estructural, logrando búsquedas semánticas que superan ampliamente los enfoques tradicionales basados en palabras clave (Lewis et al., 2021; Wang et al., 2023).
- Experiencia de usuario avanzada: la integración de Retrieval-Augmented Generation (RAG) con modelos de OpenAI produce respuestas coherentes, contextualizadas y formuladas en lenguaje natural, eliminando la necesidad de que los usuarios interpreten fragmentos crudos de texto (Gao et al., 2023).
- Aplicabilidad multisectorial demostrada: el sistema ofrece beneficios inmediatos en investigación académica (acceso rápido a literatura científica), gestión de archivos y bibliotecas digitales (búsqueda semántica en documentos históricos y escaneados) y análisis financiero (consulta ágil de reportes, balances y proyecciones), reduciendo significativamente el tiempo dedicado a la revisión manual de documentos.

El presente trabajo contribuye al campo de la Document Intelligence y la Retrieval-Augmented Generation al presentar una arquitectura completa, replicable y de bajo costo operativo que puede implementarse en entornos reales con requisitos mínimos de infraestructura. La solución no solo resuelve un problema técnico recurrente, sino que habilita nuevos flujos de trabajo basados en conocimiento en organizaciones que dependen del manejo intensivo de documentación.

Aunque el sistema ya es funcional y competitivo, existen líneas claras de mejora que potenciarían su impacto:

- Incorporación de OCR avanzado (Tesseract 5 + modelos de visión como LayoutLMv3 o Donut) para procesar PDFs escaneados y documentos exclusivamente imagen.
- Soporte multiformato (DOCX, PPTX, XLSX, HTML, EPUB) y extracción multimodal (tablas, figuras, ecuaciones).
- Fine-tuning o uso de modelos open-source especializados (Llama-3, Mistral-7B-Instruct) para reducir costos y mejorar el control sobre la generación de respuestas.
- Implementación de memoria conversacional persistente y agentes multi-step para resolver consultas que requieran razonamiento encadenado.
- Interfaz multimodal (voz a texto y texto a voz) y despliegue como asistente virtual empresarial.
- Evaluación rigurosa mediante benchmarks estandarizados (BEIR, MTEB, DocVQA) y estudios de usabilidad con usuarios reales de los sectores objetivo.

Con estas evoluciones, el sistema tiene el potencial de convertirse en una herramienta estándar para la gestión inteligente del conocimiento documental, contribuyendo significativamente a la transformación digital de instituciones académicas, archivos públicos y empresas en Paraguay y la región.

REFERENCIAS

Adobe. (2023). *Digital Trends Report 2023*. https://business.adobe.com/resources/digital-trends-report.html

Chase, H. (2023). LangChain documentation. https://python.langchain.com

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186.

Hobbs, J. (2022). The limitations of keyword-based search in unstructured data. *Journal of Information Retrieval*, 25(3), 289–310.

Lewis, P., et al. (2021). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474.

Mikolov, T., et al. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nakano, R., et al. (2024). Retrieval-augmented generation systems: A survey. *ACM Computing Surveys*.

OpenAI. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pinecone Systems. (2024). *Pinecone documentation and performance benchmarks*. https://docs.pinecone.io

PyPDF2 Contributors. (2023). PyPDF2 GitHub. https://github.com/py-pdf/PyPDF2

Reinsel, D., Gantz, J., & Rydning, J. (2023). *The digitization of the world: From edge to core*. IDC White Paper.

PyPDF2 Contributors. (2023). PyPDF2 GitHub. https://github.com/py-pdf/PyPDF2

Smith, R. (2022). An overview of the Tesseract OCR engine. ICDAR.

Wang, L., et al. (2023). Document AI: A survey of foundation models for document intelligence. *Proceedings of ACL 2023*.Pinecone Documentation. https://docs.pinecone.io