

INFORME TÉCNICO

RETO 1 HACKYTHON

Optimización Inteligente de
Procesos de Licitación
en Construcción

Proyecto de Aprendizaje: Sistema de IA para análisis
automatizado de documentos

Tecnologías: Python, OpenAI, ChromaDB, Jupyter
Notebooks

Nivel Académico: Estudiantes de primeros niveles en
programación

Fecha: Agosto 2025
Versión: 1.0
Estado: Prototipo Educativo

Desarrollado por:
Equipo de Estudiantes
Hackython 2025
Proyecto de Aprendizaje

Resumen Ejecutivo

Contexto Académico

Este proyecto fue desarrollado por estudiantes de primeros niveles en programación como parte del Hackython 2025. Durante el desarrollo enfrentamos conceptos avanzados de inteligencia artificial que inicialmente resultaron desafiantes para nuestro nivel académico, pero que logramos implementar mediante investigación, experimentación y apoyo colaborativo.

Objetivo del Proyecto

Desarrollar un sistema automatizado que utilice inteligencia artificial para procesar y analizar documentos de licitación en el sector construcción, con el fin de:

- Reducir el tiempo de revisión manual de documentos
- Detectar cláusulas críticas y posibles riesgos
- Facilitar la comparación objetiva entre propuestas
- Generar reportes automáticos de análisis

Tecnologías Implementadas

El sistema utiliza un stack tecnológico que incluye:

- **Python** como lenguaje principal
- **OpenAI Embeddings** para representación vectorial de texto
- **ChromaDB** para almacenamiento y búsqueda vectorial
- **Jupyter Notebooks** para desarrollo iterativo
- **Streamlit** para interfaz web básica

1. Arquitectura del Sistema

El sistema implementa un pipeline de procesamiento modular organizado en los siguientes componentes:

1.1. Pipeline de Procesamiento

1. **Extracción de Texto** (01_extract_text.ipynb)
2. **Procesamiento y Limpieza** (02_process_text.ipynb)
3. **Consultas Vectoriales** (04_query_vector_db.ipynb)
4. **Generación de Reportes** (05_generate_report.ipynb)
5. **Detección de Problemas** (06_detect_issues.ipynb)

1.2. Herramientas Especializadas

- **Comparador Pliego vs Propuestas** (compare_pliego_vs_propuestas.ipynb)
- **Comparador de Múltiples Propuestas** (compare_proposals.ipynb)
- **Sistema de Validadores** (validators.ipynb)
- **Aplicación Web** (app.py)
- **Interfaz Principal** (app_rol1.ipynb)

2. Funcionalidades Implementadas

2.1. Procesamiento de Documentos

El sistema puede procesar documentos PDF extrayendo su contenido textual y organizándolo en fragmentos manejables para el análisis posterior.

2.2. Búsqueda Semántica

Implementación de un sistema de consultas que permite realizar búsquedas conceptuales en lugar de búsquedas por palabras clave exactas.

2.3. Comparación Automatizada

Herramientas para comparar automáticamente:

- Pliegos técnicos contra propuestas recibidas
- Múltiples propuestas entre sí
- Cumplimiento de requisitos específicos

2.4. Detección de Riesgos

Sistema básico para identificar cláusulas que podrían representar riesgos potenciales en los contratos.

2.5. Generación de Reportes

Capacidad para generar reportes automáticos con métricas básicas del procesamiento realizado.

3. Desafíos de Aprendizaje Enfrentados

Conceptos Complejos para Estudiantes Principiantes

Como estudiantes de primeros niveles, varios conceptos técnicos representaron desafíos significativos que requirieron investigación adicional y experimentación extensa.

3.1. Embeddings y Representación Vectorial

Desafío: Comprender cómo el texto se convierte en vectores numéricos y por qué esto es útil para búsquedas semánticas.

Solución de Aprendizaje: Investigación de tutoriales, experimentación con diferentes tipos de embeddings y pruebas prácticas para observar los resultados.

3.2. APIs y Autenticación

Desafío: Integración con servicios externos como OpenAI API, manejo de claves de autenticación y variables de entorno.

Solución de Aprendizaje: Estudio de documentación oficial, implementación gradual con validación de cada paso.

3.3. Bases de Datos Vectoriales

Desafío: Concepto completamente nuevo de almacenamiento y búsqueda basada en similitud vectorial.

Solución de Aprendizaje: Experimentación práctica con ChromaDB, pruebas de diferentes configuraciones y parámetros.

3.4. Fragmentación de Documentos

Desafío: Entender por qué es necesario dividir documentos en fragmentos más pequeños y cómo optimizar este proceso.

Solución de Aprendizaje: Pruebas con diferentes tamaños de fragmentos y observación de impacto en la calidad de resultados.

4. Resultados Obtenidos

4.1. Funcionalidades Logradas

Logros del Proyecto

- Sistema funcional de extracción de texto de documentos PDF
- Base de datos vectorial operativa con búsqueda semántica básica
- Herramientas de comparación automática entre documentos
- Interfaz web básica para interacción con el sistema
- Sistema de generación de reportes automatizado
- Múltiples notebooks especializados para diferentes funciones

4.2. Métricas de Rendimiento

Métrica	Valor
Documentos procesables	PDF, DOCX
Tiempo promedio de procesamiento	2-3 minutos/documento
Precisión de extracción de texto	90-95 %
Capacidad de almacenamiento	Cientos de documentos
Tipos de consulta soportados	Búsqueda semántica básica

Cuadro 1: Métricas de rendimiento del sistema

5. Limitaciones Identificadas

5.1. Limitaciones Técnicas

- **Formatos soportados limitados:** Solo PDF y DOCX nativos
- **Sin OCR:** No procesa imágenes con texto embebido
- **Análisis básico:** Detección de riesgos simplificada
- **Escalabilidad limitada:** Optimizado para volúmenes pequeños-medianos

5.2. Limitaciones de Conocimiento

Como estudiantes principiantes, identificamos áreas donde nuestro conocimiento actual es limitado:

- Optimización avanzada de algoritmos de IA
- Técnicas sofisticadas de procesamiento de lenguaje natural
- Arquitecturas de software empresariales
- Seguridad y encriptación de datos sensibles

6. Lecciones Aprendidas

6.1. Metodología de Desarrollo

- **Desarrollo iterativo:** Implementar funcionalidades básicas antes de avanzar a características complejas
- **Experimentación constante:** Probar diferentes configuraciones para encontrar las óptimas
- **Documentación continua:** Registrar cada paso del proceso de aprendizaje
- **Colaboración efectiva:** Aprovechar el conocimiento colectivo del equipo

6.2. Competencias Técnicas Desarrolladas

- Manejo intermedio de Python y sus librerías especializadas
- Comprensión básica de conceptos de inteligencia artificial
- Experiencia práctica con APIs y servicios web
- Desarrollo de interfaces de usuario básicas
- Organización y estructura de proyectos de software

7. Trabajo Futuro

7.1. Mejoras a Corto Plazo

- Implementación de OCR para procesamiento de imágenes
- Mejora de la interfaz web con más funcionalidades
- Optimización del rendimiento para documentos grandes
- Implementación de más validadores automáticos

7.2. Mejoras a Largo Plazo

- Desarrollo de un modelo de IA especializado en contratos de construcción
- Integración con sistemas empresariales (ERP, CRM)
- Implementación de análisis predictivo de riesgos
- Desarrollo de una interfaz web profesional completa

8. Conclusiones

Este proyecto representa un logro significativo en nuestro proceso de aprendizaje como estudiantes de primeros niveles en programación. A pesar de enfrentar conceptos técnicos avanzados que inicialmente estaban fuera de nuestro nivel de conocimiento, logramos implementar un sistema funcional que demuestra el potencial de la inteligencia artificial aplicada al análisis de documentos.

8.1. Impacto Educativo

El proyecto nos permitió:

- Comprender conceptos avanzados de IA de manera práctica
- Desarrollar competencias en integración de tecnologías
- Experimentar con herramientas de desarrollo profesionales
- Aprender metodologías de desarrollo de software

8.2. Valor del Prototipo

Aunque reconocemos las limitaciones de nuestro prototipo, consideramos que representa:

- Una base sólida para futuro desarrollo
- Demostración práctica de conceptos teóricos
- Experiencia valiosa en resolución de problemas complejos
- Fundamento para proyectos más avanzados

8.3. Reflexión Final

Este proyecto nos ha demostrado que, con dedicación, investigación y colaboración, es posible abordar desafíos técnicos complejos incluso desde niveles iniciales de conocimiento. La experiencia ha fortalecido nuestra confianza para enfrentar futuros proyectos de mayor complejidad y nos ha proporcionado una base sólida en tecnologías de inteligencia artificial.

9. Anexos

9.1. Estructura de Archivos del Proyecto

- `01_extract_text.ipynb`: Extracción de texto de documentos PDF
- `02_process_text.ipynb`: Procesamiento y limpieza de texto
- `04_query_vector_db.ipynb`: Sistema de consultas vectoriales
- `05_generate_report.ipynb`: Generación automática de reportes
- `06_detect_issues.ipynb`: Detección de problemas y riesgos
- `app.py`: Aplicación web con interfaz Streamlit
- `compare_pliego_vs_propuestas.ipynb`: Comparación pliego-propuestas
- `compare_proposals.ipynb`: Comparación entre múltiples propuestas
- `validators.ipynb`: Sistema de validadores automáticos

9.2. Recursos de Aprendizaje Utilizados

- Documentación oficial de OpenAI
- Tutoriales de LangChain y ChromaDB
- Recursos educativos sobre procesamiento de lenguaje natural
- Documentación de Python y librerías especializadas

- Foros de desarrollo y comunidades de programadores

*Informe elaborado por estudiantes del Hackython 2025
Proyecto de aprendizaje en Inteligencia Artificial aplicada*