Memoria Técnica - Proyecto final: Implementación de un Sistema RAG (Retrieval-Augmented Generation)

Integrantes: Andrade Dilan

Calahorrano David

Zambrano Andrés

Fecha de entrega: 13 de febrero de 2025

---

1. Introducción

Esta memoria técnica describe la implementación del proyecto Sistema RAG (Retrieval-Augmented Generation) utilizando técnicas de Scrum como metodología ágil de desarrollo. Se detallan la planificación del proyecto, los roles asignados, la ejecución de los sprints, las decisiones tomadas y la evaluación final del proceso.

El proyecto fue desarrollado en un período de tres semanas, desde el 27 de enero hasta el 13 de febrero de 2025. Se centró en la construcción de un sistema que combina técnicas de recuperación de información y generación de respuestas mediante modelos avanzados de inteligencia artificial.

2. Planificación inicial del proyecto

Para asegurar una ejecución eficiente del proyecto, se definió un plan basado en Scrum, dividiendo el desarrollo en tres sprints iterativos.

- **Sprint 1:** Preprocesamiento y Base de Datos (27 de enero - 2 de febrero)
    - **Recopilación y limpieza del corpus**: Se obtuvieron documentos relevantes, principalmente planes de trabajo y entrevistas de los candidatos presidenciales. Se aplicaron técnicas de preprocesamiento como eliminación de ruido, tokenización, lematización y stopwords removal.
    - **Generación de embeddings:** Se utilizó Sentence Transformers (all-MiniLM-L6-v2) para representar los textos en un espacio vectorial.
    - **Implementación de ChromaDB:** Se configuró ChromaDB como base de datos vectorial para almacenar y recuperar documentos de manera eficiente.

- **Sprint 2:** Implementación del Modelo de Generación (3 de febrero - 8 de febrero)
    - **Desarrollo del modelo de generación**: Se utilizó T5 (Flan-T5-Large), un modelo de transformer especializado en tareas de generación de texto.
    - **Integración con recuperación de documentos:** Se desarrolló una funcionalidad que permite que los documentos recuperados alimenten al modelo de generación, asegurando respuestas contextualizadas.

- **Creación de la API en Flask:** Se implementó una API en Flask para que los usuarios pudieran interactuar con el sistema de manera sencilla a través de consultas en lenguaje natural.

- **Sprint 3:** Evaluación y Ajustes Finales (9 de febrero - 12 de febrero)
  - **Medición de precisión de recuperación**: Se calcularon métricas como Precision@k, Recall y F1-Score para evaluar la eficacia del sistema de recuperación de documentos.
  - **Evaluación de generación de respuestas:** Se midió la calidad de las respuestas con BLEU y ROUGE-L, comparando las respuestas generadas con respuestas de referencia.
  - **Optimización del sistema:** Se ajustaron hiperparámetros del modelo de generación y recuperación para mejorar la calidad de las respuestas.

3. Sprints definidos y objetivos

Sprint	Fechas	Objetivo principal	Entregables
Sprint 1	27 de enero - 31 de enero	Procesamiento del corpus y configuración de base de datos	Corpus preprocesado, embeddings generados, ChromaDB implementado
Sprint 2	3 de febrero - 7 de febrero	Desarrollo del modelo de generación y su integración	Modelo T5 implementado, API Flask desarrollada
Sprint 3	10 de febrero - 13 de febrero	Evaluación y optimización del sistema	Métricas obtenidas, ajustes finales aplicados

4. Roles asignados en el equipo

Nombre	Rol en el equipo	Responsabilidades
Andrés Zambrano	Scrum Master/Backend	Supervisar la implementación del backend y gestionar el equipo.
Dilan Andrade	Procesamiento de datos/Embeddings	Preprocesamiento del corpus y generación de embeddings.
David Calahorrano	Frontend/Evaluación	Desarrollo de la interfaz y pruebas del sistema.

5. Documentación de decisiones tomadas

- ChromaDB se utilizó para la recuperación de documentos en lugar de SQL por su rapidez.
- Se seleccionó Sentence Transformers para generar embeddings en lugar de TF-IDF debido a su capacidad para capturar mejor el significado del texto.

- Se usó T5 (Flan-T5-Large) como modelo de generación debido a su efectividad en la producción de respuestas naturales y contextualizadas.
- Se implementó Flask como framework para la interfaz web, por su facilidad de integración con los módulos desarrollados.

6. Evaluación final del uso de scrum

Critero	Observaciones
Cumplimiento de Srpints	Se lograron completar las tareas dentro del tiempo establecido.
Coordinación del equipo	La asignación de roles permitió dividir eficientemente el trabajo.
Retos encontrados	Ajustes en la precisión de generación y recuperación.
Soluciones implementadas	Optimización de hiperparámetros y mejora en la selección de respuestas.

7. Conclusiones

- Durante el desarrollo del proyecto, se logró implementar un sistema RAG funcional que combina técnicas de recuperación de información con modelos de generación de texto, asegurando que las respuestas sean precisas y relevantes.
- La implementación de ChromaDB para la recuperación de documentos y T5 para la generación de respuestas permitió alcanzar un desempeño óptimo en términos de precisión y relevancia. Se observó que las respuestas generadas son coherentes y adecuadas, aunque podrían mejorarse con modelos más avanzados o un corpus más amplio.
- El uso de la metodología Scrum facilitó la ejecución eficiente del proyecto, organizando el trabajo en tres sprints iterativos que priorizaron las tareas más críticas. A pesar de los desafíos encontrados, como el ajuste de hiperparámetros y la optimización del rendimiento, se logró desarrollar un sistema estable y funcional en un período de tres semanas.