ESCUELA POLITÉCNICA NACIONAL

Recuperación de Información (ICCD753)

Memoria técnica

Elaborado por:
Carlos Córdova
Galo Tarapues
Hernán Sánchez

Docente:
Prof. Iván Carrera

Contenido

Introducción

Contexto del Proyecto

El presente proyecto se desarrolló como proyecto final de la materia de Recuperación de la Información en la Escuela Politécnica Nacional. La iniciativa nace de la necesidad de aplicar y consolidar los conocimientos adquiridos durante el curso, poniendo en práctica conceptos teóricos y metodológicos a través de la implementación de un sistema RAG (Retrieval-Augmented Generation). Este proyecto se distingue por su enfoque en la metodología ágil Scrum, la cual ha sido aplicada en proyectos anteriores con otros temas, permitiendo que el equipo cuente con experiencia previa en la planificación, coordinación y ejecución de desarrollos complejos. Cabe mencionar que, debido a la identificación de nuevas oportunidades de mejora durante el proceso, la fecha de finalización se extendió del 7 al 12 de febrero, lo que permitió afinar aspectos críticos para cumplir plenamente con los requerimientos establecidos.

Objetivos

El objetivo principal de este proyecto es llevar a cabo la implementación exitosa de un sistema RAG, integrando técnicas de Recuperación de Información y generación de texto, a partir de la aplicación de conocimientos teóricos y prácticos adquiridos en clase. De forma más específica, se busca:

- Aplicar de manera efectiva la metodología Scrum para la gestión y ejecución del proyecto, optimizando la coordinación del equipo y asegurando la entrega de cada sprint.
- Garantizar que cada fase del proceso (planificación, ejecución, seguimiento y ajustes) se desarrolle conforme a los objetivos planteados, solucionando el proyecto de la mejor manera posible.
- Cumplir con todos los requerimientos establecidos, asegurando que el sistema responda adecuadamente a las necesidades de consulta y generación de respuestas basadas en el corpus seleccionado.

Alcance

La presente Memoria Técnica abarca la documentación detallada de todo el proceso de gestión del proyecto, desde su inicio hasta la finalización. En este documento se describen los aspectos relacionados con la planificación, el seguimiento de los sprints y la implementación de la metodología Scrum, sin adentrarse en los detalles técnicos de la implementación del sistema RAG, los cuales serán desarrollados en el Informe Técnico..

Organización del Equipo

Para asegurar una gestión ágil y eficiente del proyecto, se adoptó la metodología Scrum, la cual establece roles y responsabilidades que permiten un seguimiento y coordinación óptimos a lo largo del desarrollo. En este proyecto, conformado por Carlos Cordova, Hernán Sánchez y Galo Tarapués, se definieron los siguientes roles:

Scrum Master – Hernán Sánchez
Hernán asumió el rol de Scrum Master, siendo el facilitador principal del proceso Scrum. Entre sus responsabilidades destacan:

- Coordinar y moderar las reuniones diarias (Daily Scrums) para mantener una comunicación fluida.
- Identificar y resolver impedimentos que puedan afectar el progreso del equipo.
- Asegurar la correcta implementación de las prácticas ágiles, adaptando los procesos según los desafíos que surjan.
- Promover la colaboración y el intercambio de ideas entre los miembros del equipo.

Product Owner – Profesor
El rol de Product Owner fue asumido por el profesor, quien actúa como representante de los stakeholders y es el responsable de definir y priorizar el Product Backlog. Sus funciones principales incluyen:

- Proveer una visión clara y estratégica del producto final.
- Establecer y comunicar los requerimientos y objetivos del proyecto.
- Evaluar y aceptar los entregables, asegurándose de que se cumplan las expectativas establecidas.

Equipo de Desarrollo – Carlos Cordova, Hernán Sánchez y Galo Tarapués
El equipo de desarrollo, integrado por los tres miembros, es el núcleo responsable de la implementación técnica del sistema RAG. Las funciones y responsabilidades compartidas incluyen:

- Programación y codificación de los módulos del sistema.
- Realización de pruebas, integración de funcionalidades y aseguramiento de la calidad del software.
- Participación activa en la planificación de sprints, colaborando en la definición de tareas y la solución de problemas.
- Comunicación constante con el Scrum Master para reportar avances y gestionar bloqueos, contribuyendo a la mejora continua del proceso.

Planificación inicial

Visión General del Producto

El objetivo es desarrollar un sistema RAG que integre de manera eficiente técnicas de recuperación de información y generación de texto. Este sistema permitirá consultar planes de trabajo y entrevistas de los candidatos a la presidencia, ofreciendo respuestas generadas a partir de un corpus previamente procesado. Se busca que el producto final cuente con una interfaz amigable que conecte un backend robusto, desarrollado en

Python, con módulos especializados en preprocesamiento, recuperación y generación de respuestas.

Herramientas y Tecnologías Seleccionadas

Para garantizar una implementación escalable y eficiente, se optó por un conjunto de herramientas y tecnologías que facilitan la integración de los distintos módulos del sistema:

1. Lenguaje de Programación

- **Python**: Utilizado para el desarrollo del sistema, incluyendo el procesamiento del corpus, la recuperación de información y la generación de respuestas.

2. Preprocesamiento de Texto

- **NLTK (Natural Language Toolkit)**: Empleado para la tokenización de texto, normalización y eliminación de ruido en los documentos.
- **Expresiones Regulares (re)**: Aplicadas para limpiar caracteres especiales y estandarizar los datos textuales antes de su procesamiento.

3. Representación Semántica y Embeddings

- **Gensim (Word2Vec)**: Utilizado para entrenar embeddings de palabras y representar semánticamente los textos.
- **TF-IDF (Scikit-learn)**: Implementado para modelar la importancia de las palabras en los documentos, permitiendo una mejor recuperación de información basada en relevancia.

4. Recuperación de Información

- **Scikit-learn (Similaridad del Coseno)**: Implementado para calcular la similitud entre documentos y consultas, permitiendo recuperar la información más relevante del corpus.

5. Transcripción de Entrevistas

- **Whisper (de OpenAI)**: Utilizado para convertir los audios de las entrevistas en texto, facilitando su integración en el corpus textual.

6. Generación de Texto

- **GPT-3.5 Turbo (OpenAI API)**: Aplicado en el módulo de generación de respuestas, asegurando que las respuestas sean coherentes y relevantes para la consulta del usuario.

7. Evaluación del Sistema

- **Scikit-learn (Métricas de Evaluación: Precision@k, Recall, y F1-score)**: Empleado para medir la efectividad del sistema de recuperación de información y validar la calidad de los resultados generados.

Implementación de Scrum en el Desarrollo del Proyecto

Sprint:

Se estructuró el proyecto en tres sprints, adaptando el cronograma inicial para incorporar una extensión que permitió abordar de forma precisa los problemas surgidos. A continuación, se detalla la planificación de cada sprint:

Sprint 1: Fundamentos del Sistema

- **Periodo:** 27 de enero – 31 de enero (5 días)
- **Objetivos:**
    - Establecer la infraestructura base del sistema.
    - Desarrollar el módulo de preprocesamiento del corpus (limpieza, tokenización y generación de embeddings).
    - Configurar la base de datos y el índice FAISS para la recuperación de información.
- **Tareas Asignadas:**
    - **Carlos Córdova:** Implementar el módulo de preprocesamiento, normalizando los datos y generando los embeddings necesarios.
    - **Hernán Sánchez:** Configurar el backend y crear la API para la consulta de documentos.
    - **Galo Tarapués:** Diseñar el prototipo inicial de la interfaz de usuario e integrar el frontend con el backend básico.
- **Ceremonias:**
    - **Sprint Planning:** 27 de enero, 18:00 hrs.
    - **Daily Standups:** Diariamente a las 18:00 hrs (del 28 al 31 de enero).
    - **Sprint Review:** 31 de enero, 08:00 hrs.
    - **Sprint Retrospective:** 31 de enero, 10:00 hrs.

Sprint 2: Desarrollo y Optimización Inicial

- **Periodo:** 1 de febrero – 7 de febrero (7 días)
- **Objetivos:**
    - Desarrollar e integrar el módulo de recuperación, optimizar la consulta para mejorar la relevancia de los resultados.
    - Implementar el módulo de generación de respuestas utilizando un modelo transformer.

    - Integrar y validar la comunicación entre el frontend y los nuevos módulos implementados.

- **Tareas Asignadas:**
    - **Carlos Córdova:** Ajustar y optimizar la generación de embeddings y perfeccionar las consultas.
    - **Hernán Sánchez:** Desarrollar el módulo de generación de respuestas y realizar pruebas unitarias para evaluar su precisión.
    - **Galo Tarapués:** Mejorar la integración del frontend, incorporando elementos que faciliten la visualización de los resultados y la interacción del usuario.
- **Problemas y Complicaciones:**
    - Se identificó una sobrecarga en el servidor durante la generación de embeddings, lo que provocó retrasos en las consultas y requirió una redistribución de tareas y ajustes temporales en el módulo de preprocesamiento.
- **Ceremonias:**
    - **Sprint Planning:** 1 de febrero, 18:00 hrs.
    - **Daily Standups:** Diariamente a las 18:00 hrs (del 2 al 7 de febrero).
    - **Sprint Review:** 7 de febrero, 17:00 hrs.
    - **Sprint Retrospective:** 7 de febrero, 18:00 hrs.

Sprint 3: Refinamiento Final y Ajustes Críticos

- **Periodo:** 8 de febrero – 12 de febrero (5 días)
- **Objetivos:**
    - Solucionar las inconsistencias detectadas en el módulo de generación de respuestas y optimizar la integración entre módulos.
    - Realizar pruebas de integración final y ajustar parámetros críticos para mejorar la precisión y eficiencia del sistema.
    - Refinar la interfaz de usuario para lograr una experiencia de uso más intuitiva y fluida.
- **Tareas Asignadas:**
    - **Carlos Córdova:** Continuar optimizando la consulta en FAISS y gestionar el manejo de errores en el módulo de recuperación.
    - **Hernán Sánchez:** Afinar el modelo de generación de respuestas incorporando el feedback recibido, mejorando los parámetros y la coherencia de las respuestas.

- **Galo Tarapués:** Realizar mejoras en el frontend, optimizando la presentación de resultados y ejecutando pruebas de usabilidad con usuarios finales.

- **Problemas y Complicaciones:**
  - Se detectaron inconsistencias en la integración del módulo de generación, lo que requirió revisar la arquitectura del backend y realizar ajustes en el flujo de datos para asegurar una comunicación eficiente entre los módulos.

- **Ceremonias:**
  - **Sprint Planning:** 8 de febrero, 18:00 hrs.
  - **Daily Standups:** Diariamente a las 18:00 hrs (del 9 al 12 de febrero).
  - **Sprint Review:** 12 de febrero, 17:00 hrs.
  - **Sprint Retrospective:** 12 de febrero, 18:00 hrs.

US:

Código: US001-01	Consulta de Información	Prioridad: Muy Alta	Estimación: 5
Como usuario del sistema RAG, **quiero** ingresar una query de consulta, **para** obtener respuestas relevantes basadas en el corpus de información de los candidatos.			
Criterios de aceptación:			
Escenario: Consulta Exitosa Dado que soy un usuario que ingresa una query bien formulada, Cuando ingreso la consulta en el sistema, Entonces el sistema debe procesar la consulta, recuperar y presentar los documentos y respuestas generadas de manera precisa. • La query se procesa convirtiéndose en embeddings de forma correcta. • El sistema recupera documentos relevantes y genera respuestas basadas en ellos. • La información se muestra en un formato claro, ordenado y legible. Escenario: Consulta sin Resultados Dado que la query ingresada no coincide con ningún documento en el corpus, Cuando ingreso la consulta, Entonces el sistema debe notificar que no se encontraron resultados y ofrecer sugerencias para mejorar la consulta. • Se muestra un mensaje indicando "No se encontraron resultados".			

Artefactos y Eventos

Product Backlog

- Módulo de preprocesamiento: limpieza, tokenización y generación de embeddings.
- Configuración e integración del índice FAISS para la recuperación de documentos.
- Desarrollo del módulo de generación de respuestas con modelos transformer.
- Implementación e integración de la API para conectar backend y frontend.
- Diseño y desarrollo de la interfaz de usuario.
- Optimización del rendimiento y gestión de errores.
- Pruebas y validación de cada funcionalidad.

Sprint Backlog

Sprint 1: Fundamentos del Sistema

- **Periodo:** 27 de enero – 31 de enero (5 días)
- **Tareas Asignadas:**
    - *Carlos Córdova:*
        - Implementar el módulo de preprocesamiento, incluyendo limpieza y tokenización del corpus.
        - Generar los embeddings utilizando SentenceTransformers.
    - *Hernán Sánchez:*
        - Configurar el backend y desarrollar la API inicial para la consulta.
    - *Galo Tarapués:*
        - Diseñar el prototipo de la interfaz de usuario e integrar de forma básica el frontend con el backend.

Sprint 2: Desarrollo y Optimización Inicial

- **Periodo:** 1 de febrero – 7 de febrero (7 días)
- **Tareas Asignadas:**
    - *Carlos Córdova:*
        - Ajustar la calidad de los embeddings y optimizar las consultas a FAISS.
    - *Hernán Sánchez:*

        - Desarrollar y refinar el módulo de generación de respuestas, realizando pruebas unitarias para validar su precisión.
    - *Galo Tarapués:*
        - Mejorar la integración del frontend, incorporando validaciones y optimizando la presentación de resultados.
- **Complicación Enfrentada:**
    - Durante este sprint, se detectó una sobrecarga en el servidor al procesar los embeddings, lo que requirió reconfigurar algunos parámetros y redistribuir tareas para mantener la eficiencia.

Sprint 3: Refinamiento Final y Ajustes Críticos

- **Periodo:** 8 de febrero – 12 de febrero (5 días)
- **Tareas Asignadas:**
    - *Carlos Córdova:*
        - Gestionar y optimizar el manejo de errores en el módulo de recuperación y realizar ajustes finales en la integración.
    - *Hernán Sánchez:*
        - Afinar el modelo de generación de respuestas basándose en el feedback recibido, mejorando la coherencia y precisión de las salidas.
    - *Galo Tarapués:*
        - Refinar el diseño del frontend, optimizando la experiencia del usuario y realizando pruebas finales de integración.
- **Complicación Enfrentada:**
    - Se detectaron inconsistencias en la integración entre el módulo de generación y la API, lo que obligó a revisar y ajustar la arquitectura del backend para asegurar una comunicación fluida.

Detalles de la Daily Scrum

- **Frecuencia y Horario:**
    - Se realizaron reuniones diarias cada mañana a las 18:00 hrs durante cada sprint.
- **Formato de la Reunión:**
    - Cada miembro respondió a:
        - ¿Qué tareas completé el día anterior?
        - ¿Qué tareas planeo realizar hoy?
        - ¿Existen impedimentos o bloqueos en mi avance?

- **Objetivo:**
    - Estas reuniones permitieron identificar rápidamente problemas, ajustar la distribución de tareas y mantener un seguimiento constante del progreso, facilitando la comunicación y colaboración entre los integrantes del equipo.

Sprint Review

Sprint 1: Fundamentos del Sistema

- Se implementó el módulo de preprocesamiento, logrando limpiar y tokenizar el corpus de manera efectiva.
- Se generaron los embeddings utilizando SentenceTransformers, y se configuró exitosamente la base de datos junto al índice FAISS.
- Se presentó un prototipo inicial del backend y la integración básica con el frontend, demostrando la funcionalidad esencial del sistema.
- Se identificaron oportunidades para mejorar la coordinación entre el backend y el frontend, especialmente en la comunicación de datos.

Sprint 2: Desarrollo y Optimización Inicial

- Se completó la integración del módulo de recuperación utilizando FAISS, con una mejora en la relevancia de los documentos recuperados.
- Se implementó el módulo de generación de respuestas, logrando una primera versión funcional del sistema de generación basado en transformers.
- Se evidenció una reducción del 20% en los tiempos de respuesta tras optimizar las consultas, aunque se detectaron problemas de integración entre algunos módulos y sobrecarga temporal en el servidor.

Sprint 3: Refinamiento Final y Ajustes Críticos

- Se resolvieron las inconsistencias entre el módulo de generación y la API, logrando una integración estable entre todos los componentes del sistema.
- Se implementaron ajustes críticos en el backend que mejoraron la comunicación entre módulos y se optimizó la presentación de resultados en el frontend.
- Se logró una mayor precisión en las respuestas generadas, y el sistema mostró un rendimiento robusto y coherente con los objetivos planteados.

Sprint Planning

Sprint 1: Fundamentos del Sistema

- Se definieron las tareas esenciales para establecer la infraestructura base del sistema.
- Se asignaron responsabilidades claras: Carlos se encargó del preprocesamiento y generación de embeddings, Hernán de la configuración del backend y Galo del diseño del prototipo del frontend.
- Se establecieron objetivos medibles y se acordó realizar reuniones diarias para monitorear el avance.

Sprint 2: Desarrollo y Optimización Inicial

- Se priorizaron tareas orientadas a optimizar el rendimiento del servidor y la integración del módulo de generación de respuestas.
- Se ajustaron los ítems del Product Backlog para abordar la sobrecarga detectada y se definieron tareas específicas para perfeccionar la comunicación entre el backend y el frontend.
- Se repartieron nuevas responsabilidades: Carlos trabajó en la optimización de embeddings y consultas, Hernán en la mejora del modelo de generación, y Galo en la integración y validación del frontend.

Sprint 3: Refinamiento Final y Ajustes Críticos

- Se definieron tareas orientadas a corregir errores críticos y a mejorar la experiencia del usuario final.
- Se priorizó la revisión del flujo de datos entre módulos y se establecieron objetivos claros para las pruebas finales e integración completa.
- Se asignaron tareas específicas para la optimización de la interfaz, la gestión de errores y la integración final del sistema.

Documentación de Decisiones y Ajustes en el Proceso

Decisiones Clave

1. **Obtención y Preprocesamiento del Corpus**
    - **Decisión Inicial:**
        - Se propuso utilizar SentenceTransformers para la generación de embeddings, complementado con un proceso de limpieza, tokenización y eliminación de stopwords.

- **Cambio Realizado:**
  - Se optimizó la limpieza de texto mediante expresiones regulares.
  - Se reemplazó SentenceTransformers por un modelo basado en Word2Vec (Gensim), lo que permitió mayor control y eficiencia.
  - Se simplificó la tokenización utilizando NLTK.
- **Justificación:**
  - El cambio permitió reducir los tiempos de ejecución y optimizar el almacenamiento, manteniendo la precisión en la recuperación de documentos.

2. **Módulo de Recuperación**
   - **Decisión Inicial:**
     - Implementación de un sistema basado en FAISS para realizar búsquedas mediante similitud de embeddings.
   - **Cambio Realizado:**
     - Se optó por utilizar TF-IDF junto con la métrica de similaridad del coseno, eliminando FAISS.
     - Se optimizó el proceso de consulta para reducir la latencia.
   - **Justificación:**
     - Dado que el corpus no era excesivamente grande, el enfoque con TF-IDF ofreció un rendimiento comparable a un menor costo computacional y menor consumo de memoria.
3. **Transcripción de Entrevistas**
   - **Decisión Inicial:**
     - Uso de Google Speech-to-Text para la transcripción de audio.
   - **Cambio Realizado:**
     - Se adoptó Whisper (de OpenAI) por su mayor precisión en la transcripción y su robusto manejo de errores.
     - Se agregó una fase de normalización del texto transcrito para corregir errores comunes.
   - **Justificación:**
     - Whisper demostró mayor exactitud en audios con ruido y acentos variados, mejorando la calidad del corpus.
4. **Generación de Respuestas**
   - **Decisión Inicial:**

        - Se planificó utilizar un modelo Transformers personalizado para la generación de respuestas.
    - **Cambio Realizado:**
        - Se sustituyó el modelo original por GPT-3.5 Turbo mediante la API de OpenAI.
        - Se implementó una estrategia de mejora en los prompts para obtener respuestas más precisas.
    - **Justificación:**
        - La adopción de GPT-3.5 Turbo permitió respuestas más coherentes y relevantes, reduciendo la necesidad de ajustes manuales en el modelo.
5. **Evaluación del Sistema**
    - **Decisión Inicial:**
        - La evaluación se centró en métricas de recuperación de información (Precision@k, Recall y F1-score).
    - **Cambio Realizado:**
        - Se incorporaron métricas adicionales (BLEU Score y ROUGE Score) para evaluar la calidad de las respuestas generadas.
        - Se complementó la evaluación automatizada con pruebas manuales para validar la coherencia.
    - **Justificación:**
        - Estas modificaciones permitieron un análisis más completo del desempeño del sistema, abordando tanto la recuperación de documentos como la calidad en la generación de respuestas.
6. **Implementación Técnica e Interfaz**
    - **Decisión Inicial:**
        - Desarrollo del frontend con HTML, CSS y JavaScript
    - **Cambio Realizado:**
        - Se mejoró la presentación de las respuestas con una estructura de visualización optimizada.
        - Se afinó la comunicación entre el frontend y backend para reducir la carga en las solicitudes.
    - **Justificación:**
        - Estas mejoras facilitaron una experiencia de usuario más intuitiva y una interpretación más clara de los resultados.

Gestión de Imprevistos y Adaptaciones

- **Desafío en la Generación de Embeddings:**
    - **Problema:** La utilización inicial de SentenceTransformers generaba un alto consumo de recursos y tiempos de procesamiento elevados.
    - **Solución:** Se migró a Word2Vec, lo que redujo la carga computacional y aceleró la generación y almacenamiento de embeddings.
    - **Impacto:** Este ajuste permitió mantener un ritmo de desarrollo constante y mejorar la eficiencia general del sistema.
- **Sobrecarga en el Servidor:**
    - **Problema:** Durante el sprint de optimización, se detectó que las consultas a FAISS (posteriormente TF-IDF) sobrecargaban el servidor.
    - **Solución:** Se optimizó el proceso de consulta y se ajustaron los parámetros del sistema para distribuir mejor la carga.
    - **Impacto:** Se logró una reducción en la latencia y una mayor estabilidad en el rendimiento, permitiendo una experiencia de usuario más fluida.
- **Inconsistencias en la Integración de Módulos:**
    - **Problema:** Se presentaron inconsistencias entre el módulo de generación de respuestas y la API del backend, afectando la comunicación entre módulos.
    - **Solución:** Se revisó la arquitectura del backend y se realizaron ajustes en el flujo de datos para asegurar una integración coherente y robusta.
    - **Impacto:** La resolución de este problema permitió que la generación de respuestas fuera más precisa y que la integración entre módulos se consolidara, cumpliendo con los objetivos del sprint final.
- **Optimización de la Transcripción:**
    - **Problema:** La precisión en la transcripción inicial utilizando Google Speech-to-Text no era la esperada en condiciones de audio subóptimas.
    - **Solución:** La adopción de Whisper y la normalización del texto transcrito mejoraron significativamente la calidad del corpus.
    - **Impacto:** Esto garantizó que la información procesada fuera de alta calidad, incrementando la precisión en las fases posteriores de recuperación y generación de respuestas.