



ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA DE SISTEMAS
COMPUTACIÓN
RECUPERACIÓN DE LA INFORMACIÓN**

IMPLEMENTACIÓN DE UN SISTEMA RAG – DOCUMENTO DE MEMORIA TÉCNICA

Docente: PhD. Iván Carrera

Integrantes: Diego Arias

Sebastián Guamán

Anthony Vargas

Fecha de entrega: 13 de febrero del 2025

Implementación de un Sistema RAG – Documento de Memoria Técnica

Metodología SCRUM

Contenido

Introducción	3
Planificación del Proyecto	4
Definición del Product Backlog	4
Criterios de Éxito del Proyecto.....	5
Roles y Responsabilidades.....	5
SPRINTS.....	6
Evaluación de Resultados	8
Sprint Retrospective	8
Burndown Chart	12
Conclusiones	13
Recomendaciones	14

Ilustraciones

Ilustración 1. Corpus Crudo.....	8
Ilustración 2. Corpus Preprocesado	9
Ilustración 3. Generación de Embeddings	9
Ilustración 4. Función LLM.....	10
Ilustración 5. Resultados LLM.....	10
Ilustración 6. Interfaz Web desde la PC	11
Ilustración 7. Interfaz Web desde el Móvil.....	11
Ilustración 8. Burndown Chart	13

Tablas

Tabla 1. Product Backlog	4
Tabla 2. Criterios de Éxito	5
Tabla 3. Tareas Sprint 1	6
Tabla 4. Tareas Sprint 2	7
Tabla 5. Tareas Sprint 3	7
Tabla 6. Trabajo Estimado y Real	13

Introducción

Este proyecto consiste en diseñar e implementar un sistema RAG (Retrieval-Augmented Generation). Se busca combinar técnicas de Recuperación de Información (RI) con modelos de generación de texto, permitiendo que un modelo genere respuestas a partir de documentos relevantes recuperados desde un corpus.

En el contexto político actual ecuatoriano, surge la necesidad de mantener informada a la población acerca de los planes de trabajo de los 16 candidatos presidenciales, y al entrever la falta de interés popular por leer cada uno de los planes de trabajo o informarse activamente fuera de los medios de comunicación tradicionales, se dispone la creación de un sistema RAG que permita recuperar información contextualizada, resumida y verás de los diferentes partidos políticos.

Es responsabilidad nuestra generar soluciones tecnológicas y eficientes que alienten el interés por la participación ciudadana en los temas de gobierno sin dejarse llevar por discursos populistas, de esta manera, el uso de un sistema RAG es el primer paso para automatizar el flujo de información sin corresponder a sesgos ni cámaras de eco.

Objetivos

1. Diseñar un sistema RAG que recupere documentos relevantes a partir de una consulta del usuario utilizando técnicas de RI.
2. El sistema RAG debe generar respuestas basadas en los documentos recuperados utilizando un modelo de lenguaje avanzado.

Planificación del Proyecto

Definición del Product Backlog

ID	Historia de Usuario	Descripción	Prioridad
US01	Configuración del entorno	Configurar el entorno en Python con las librerías necesarias.	Alta
US02	Carga del corpus	Obtener y estructurar los planes de trabajo y entrevistas de los candidatos.	Alta
US03	Preprocesamiento del corpus	Limpiar, tokenizar y normalizar los datos textuales.	Alta
US04	Generación de embeddings	Convertir el corpus en vectores para facilitar la búsqueda.	Alta
US05	Implementación del módulo de recuperación	Diseñar el motor de búsqueda basado en embeddings.	Alta
US06	Implementación del módulo de generación	Integrar un modelo de lenguaje para generar respuestas.	Alta
US07	Evaluación del sistema	Probar la recuperación y generación de respuestas con métricas definidas.	Media
US08	Desarrollo de la interfaz	Implementar una interfaz básica para consultas.	Media
US09	Documentación técnica	Redacción de la memoria técnica del proyecto.	Media
US10	Pruebas finales y ajustes	Testeo con datos reales y optimización.	Media
US11	Presentación del proyecto	Preparar la exposición y entrega de informes.	Baja

Tabla 1. Product Backlog

Criterios de Éxito del Proyecto

Criterio	Consideraciones		
	Consideración 1	Consideración 2	Consideración 3
Configuración del entorno	El entorno de desarrollo debe estar correctamente configurado en Python con las librerías necesarias	La ejecución del código no debe generar errores en un ambiente controlado	N/A
Corpus preprocesado y vectorizado	Los datos deben estar estructurados correctamente (planes de trabajo y entrevistas).	El preprocesamiento debe eliminar ruido (stopwords, caracteres especiales, errores de transcripción).	Los embeddings deben estar generados y almacenados eficientemente.
Módulo de recuperación	Debe recuperar los documentos más relevantes para una consulta con alta precisión.	Evaluación mediante Precision@k, Recall y F1-Score.	La recuperación debe ejecutarse en menos de 2 segundos en pruebas con corpus completo.
Módulo de generación	Las respuestas generadas deben ser coherentes, relevantes y precisas según la consulta y documentos recuperados.	Se debe evaluar mediante comparaciones con respuestas de referencia.	N/A
Interfaz Funcional	Debe permitir que los usuarios ingresen consultas y reciban respuestas sin errores.	La UI debe ser intuitiva y clara, incluso en una versión básica (CLI o interfaz web simple).	N/A
Evaluación del sistema	Se deben registrar métricas de rendimiento y calidad del modelo.	Comparación con métodos tradicionales de recuperación de información.	N/A
Documentación	La memoria técnica debe estar completa y bien estructurada.	El código debe estar documentado con comentarios y README detallado.	La presentación debe explicar el flujo de trabajo, implementación y resultados.

Tabla 2. Criterios de Éxito

Roles y Responsabilidades

- 1. **Product Owner:** PhD. Iván Carrera
- 2. **SCRUM Master:** Anthony Vargas
- 3. **SCRUM Tester:** Sergio Guamán
- 4. **SCRUM Developer:** Diego Arias

SPRINTS

Teniendo en cuenta el plazo de entrega del proyecto, el cual se ha establecido de 21 días empezando desde el 24 de enero de 2025 hasta el 13 de febrero del mismo año, se decidió establecer 3 Sprints, cada uno con una duración de 7 días.

Y el desarrollo de cada uno de los Sprints se definió de la siguiente manera:

- 1. **Sprint 1:** Semana 01 – 24 al 30 de enero
 - **Objetivo:** Configurar el entorno, obtener y preprocesar el corpus.

Descripción del Trabajo:

ID	Historia de Usuario	Descripción	Trabajo
US01	Configuración del entorno	Instalar y configurar Python, librerías y entorno de trabajo.	10
US02	Carga del corpus	Descargar, estructurar y validar los datos de los planes de trabajo y entrevistas.	25
US03	Preprocesamiento del corpus	Limpiar, tokenizar y normalizar los datos.	20
US04	Generación de embeddings	Convertir el corpus en vectores y almacenarlos.	25
US05	Documentación inicial	Registrar decisiones sobre herramientas y estructura del proyecto.	10

Tabla 3. Tareas Sprint 1

Suma total de trabajo para el Sprint: 90

Entrega Esperada: Corpus preprocesado y vectorizado. Entorno listo para pruebas.

2. **Sprint 2:** Semana 02 – 31 de enero al 06 de febrero

- **Objetivo:** Implementar los módulos de recuperación y generación.

Descripción del Trabajo:

ID	Historia de Usuario	Descripción	Trabajo
US06	Implementación del módulo de recuperación	Diseñar el motor de búsqueda basado en embeddings.	30
US07	Implementación del módulo de generación	Integrar un modelo de lenguaje para generar respuestas.	30
US08	Evaluación inicial del sistema	Probar la recuperación y generación con consultas de prueba.	20
US09	Optimización del preprocesamiento	Ajustar embeddings y limpiar errores en la transcripción.	20

Tabla 4. Tareas Sprint 2

Suma total de trabajo para el Sprint: 100

Entrega Esperada: Motor de recuperación funcional y primeras pruebas del modelo generador.

3. **Sprint 3:** Semana 03 – 07 al 13 de febrero

- **Objetivo:** Evaluar, documentar y preparar la presentación.

Descripción del Trabajo:

ID	Historia de Usuario	Descripción	Trabajo
US10	Refinamiento y pruebas finales	Medir rendimiento con métricas Precision@k, Recall y F1-Score.	20
US11	Desarrollo de interfaz	Implementar una UI básica para consultas.	15
US12	Documentación y memoria técnica	Redactar informe detallado del proyecto.	15
US13	Preparación de la presentación	Resumir el trabajo para la exposición final.	10

Tabla 5. Tareas Sprint 3

Suma total de trabajo para el Sprint: 60

Entrega Esperada: Informe técnico finalizado, interfaz funcional y presentación lista.

Considerando el trabajo total de cada uno de los Sprints se puede determinar un total de **250 puntos de trabajo** que abarcarán a todo el proyecto.

Evaluación de Resultados

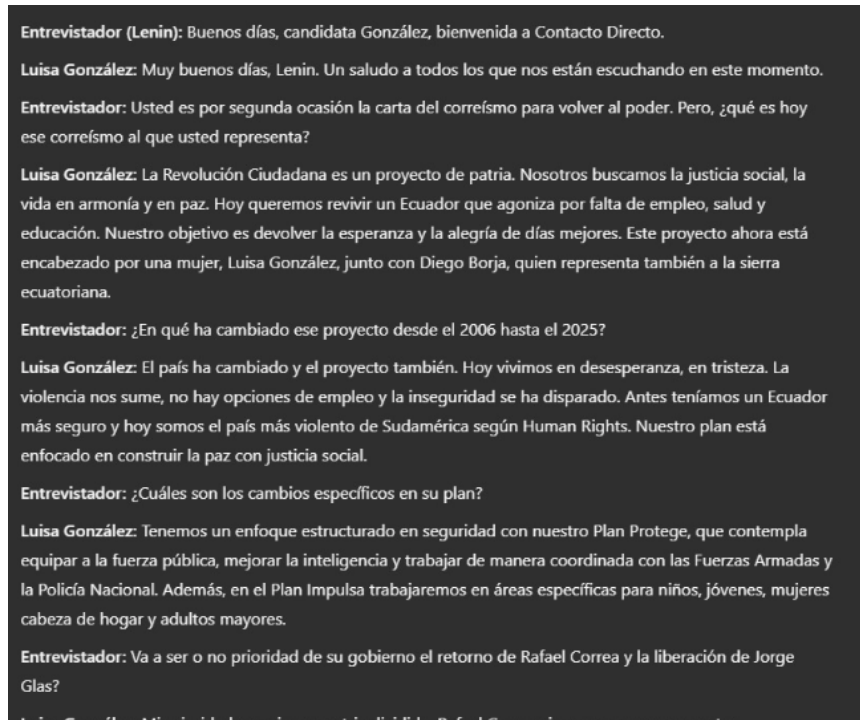
Para tener una perspectiva correcta sobre el flujo de trabajo del proyecto y elaborar una mejor conclusión, se especificará en este apartado los comentarios realizados por el SCRUM Team con relación al Sprint Retrospective, deuda técnica y el gráfico de trabajo restante o Burndown Chart.

Sprint Retrospective

El Sprint Retrospective son los comentarios acerca del trabajo realizado al final de cada Sprint, estos vienen acompañados con la descripción de la deuda técnica a trabajar en el siguiente Sprint.

Se describe a continuación el detalle por cada uno de los Sprints:

1. **Sprint 01:** Configurar el entorno, obtener y preprocesar el corpus.
 - **Evidencia:** Extracción y procesamiento del Corpus.



Entrevistador (Lenin): Buenos días, candidata González, bienvenida a Contacto Directo.

Luisa González: Muy buenos días, Lenin. Un saludo a todos los que nos están escuchando en este momento.

Entrevistador: Usted es por segunda ocasión la carta del correísmo para volver al poder. Pero, ¿qué es hoy ese correísmo al que usted representa?

Luisa González: La Revolución Ciudadana es un proyecto de patria. Nosotros buscamos la justicia social, la vida en armonía y en paz. Hoy queremos revivir un Ecuador que agoniza por falta de empleo, salud y educación. Nuestro objetivo es devolver la esperanza y la alegría de días mejores. Este proyecto ahora está encabezado por una mujer, Luisa González, junto con Diego Borja, quien representa también a la sierra ecuatoriana.

Entrevistador: ¿En qué ha cambiado ese proyecto desde el 2006 hasta el 2025?

Luisa González: El país ha cambiado y el proyecto también. Hoy vivimos en desesperanza, en tristeza. La violencia nos sume, no hay opciones de empleo y la inseguridad se ha disparado. Antes teníamos un Ecuador más seguro y hoy somos el país más violento de Sudamérica según Human Rights. Nuestro plan está enfocado en construir la paz con justicia social.

Entrevistador: ¿Cuáles son los cambios específicos en su plan?

Luisa González: Tenemos un enfoque estructurado en seguridad con nuestro Plan Protege, que contempla equipar a la fuerza pública, mejorar la inteligencia y trabajar de manera coordinada con las Fuerzas Armadas y la Policía Nacional. Además, en el Plan Impulsa trabajaremos en áreas específicas para niños, jóvenes, mujeres cabeza de hogar y adultos mayores.

Entrevistador: Va a ser o no prioridad de su gobierno el retorno de Rafael Correa y la liberación de Jorge Glas?

Luisa González: Mi prioridad es unir una patria dividida. Rafael Correa sigue su proceso en cortes

Ilustración 1. Corpus Crudo


```

# Leer el CSV
file_path = "../Data/Entrevista_LuisaGonzales_JuanCueva.csv"
data = pd.read_csv(file_path, sep=",") # Asegurate de usar el separador adecuado (tabulación en este caso)

```

	ID	Candidato	Temas	Descripción	Entrevista
0	LG1	Luisa Gonzales	justicia social propuesta corrupción	entrevista relevante entender propuesta visión...	buen día lenin saludo escuchar momento revoluc...
1	LG2	Luisa Gonzales	crisis eléctrico relación internacional acción	entrevista ofrecer visión claro prioridad estr...	mucho gracia buen día ecuatoriano mirar moment...
2	LG3	Luisa Gonzales	venezuela rafael correo persecución	gonzález defender independencia capacidad lide...	querido fernando necesitar consultar ninguno a...
3	JC1	Juan Cueva	propiedad intelectual panorama ecuatoriano	abordo tema dave relacionado protección gesti...	propiedad intelectual derecho cualquiera autor...
4	JC2	Juan Cueva	proyecto político experiencia corrupción	entrevista juan iván cueva candidato presidenc...	mucho gracia lenin invitación honor aquí prese...
5	JC3	Juan Cueva	colaboración sector desafío protección caso éx...	entrevista juan iván cueva compartir experienc...	estrategia considerar efectivo ver protección ...

Ilustración 2. Corpus Preprocesado

- **Retrospectiva:** Las tareas que retrasaron el flujo de trabajo del proyecto tiene que ver con el tratamiento del Corpus (US02, US03) y esto debido a que al tratarse de un proyecto con diferentes equipos y grupos se encargó la responsabilidad a cada grupo por extraer el corpus y configurarlo con un formato en específico, para esto se establecieron los candidatos que les correspondería a cada grupo. Sin embargo, la entrega del corpus por parte de todos los grupos fue tardía, lo que conllevó a retrasar la tarea de la generación de embeddings (US04), además de que el formato en algunos de los grupos no era el acordado.

Para mitigar este efecto, el SCRUM Team optó por trabajar únicamente con un corpus limitado de dos candidatos y a la par ir consiguiendo el resto del corpus mientras se elaboran las funciones para la generación de embeddings, el sistema de RI tradicional y experimentar con LLMs.

- **Deuda Técnica:** Al centrarnos en aspectos técnicos, descuidamos las tareas de documentación.
 - **Tareas pendientes:** US05

2. Sprint 2: Implementar los módulos de recuperación y generación.

- **Evidencia:** Generación de los Embeddings y experimentación con el LLM.

```

query = "corrupción" # la consulta en texto
query_embedding = model.encode([query]).tolist() # Generar el embedding de la consulta

results = collection.query(
    query_embeddings=query_embedding,
    n_results=3 # Devuelve los 3 documentos más similares
)

print("Resultados de la consulta:", results)

```

Resultados de la consulta: {'ids': [['LG3', 'LG2', 'JC2']], 'embeddings': None, 'documents': [['querido fernando necesitar consultar ninguno asesor manejar d...

Ilustración 3. Generación de Embeddings

```

# Obtener el contexto de los documentos relevantes
context = "\n".join(relevant_docs['content'])

# Definir el prompt que incluirá la consulta y el contexto
prompt_template = f'''SYSTEM: Eres una asistente personal que busca ayudar lo mejor posible.

USER: {query}

CONTEXT: {context}

ASSISTANT:
'''

# Generar la respuesta con Llama
response = lcgp_llm(prompt=prompt_template, max_tokens=256, temperature=0.3, top_p=0.95, repeat_penalty=1.2, top_k=150, echo=True)

# Imprimir la respuesta generada
print(response["choices"][0]["text"])

```

Ilustración 4. Función LLM

Ilustración 5. Resultados LLM

- **Retrospectiva:** En este Sprint se experimentó con varios LLMs ya que el problema que se tenía es que la respuesta otorgada por estos variaba mucho entre el inglés y el español y en algunos casos sin completar correctamente las palabras, debido a eso usamos el modelo “TheBloke/Llama-2-13B-chat-GGML” que en si viniese a ser una versión optimizada de Llama 2 en formato GGML. Este modelo genera respuestas basadas en los documentos recuperados lo que le otorga un contexto ideal. Sin embargo, el problema con el uso de este modelo es que es computacionalmente costoso y debido a nuestras limitaciones de Hardware las respuestas pueden tardar en generarse varios minutos.
- **Deuda Técnica:** Debido a limitaciones relacionadas con el choque de horarios seguíamos centrándonos en la elaboración de la parte técnica, descuidando el proceso de documentación, además de que aún no se ha logrado optimizar el preprocesamiento.
 - **Tareas Pendientes:** US05, US09

3. Sprint 3: Evaluar, documentar y preparar la presentación.

- **Evidencia:**

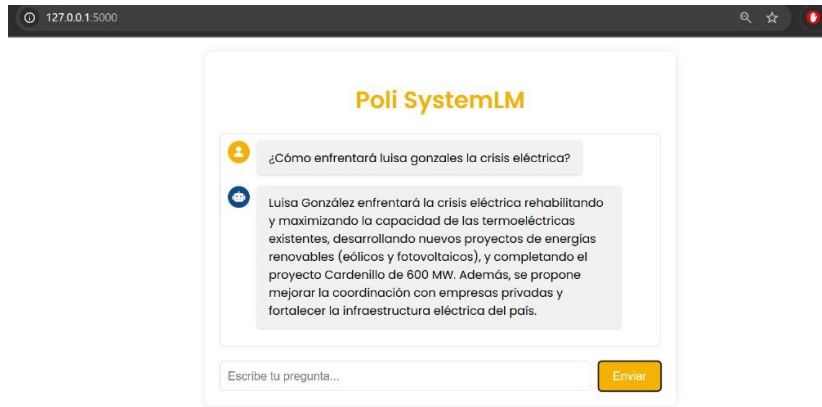


Ilustración 6. Interfaz Web desde la PC

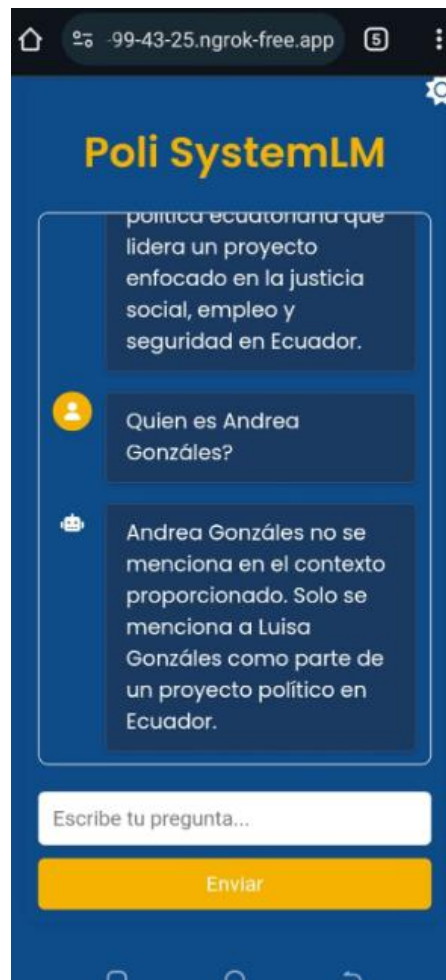


Ilustración 7. Interfaz Web desde el Móvil

- **Retrospectiva:** En este último Sprint se pudo solventar la deuda técnica de los anteriores dos, nos centramos más en la parte de optimizar el rendimiento (US09) puesto que, el modelo de lenguaje anteriormente utilizado generaba respuestas correctas, pero luego de casi 11 minutos de ejecución, lo que resultaba inviable y se solucionó al hacer uso de un modelo de lenguaje de suscripción como lo es GPT y a través de este realizamos las pruebas para garantizar resultados correctos.
Por lo mismo, la elaboración de la interfaz se realizó alrededor de este nuevo modelo y se adecuó para su uso en ordenadores y en dispositivos móviles
- **Deuda Técnica:** A pesar de haber concluido con el sistema RAG, este se encuentra únicamente desplegado en un ambiente local con su respectiva interfaz web, sin embargo, se realizaron intentos por desplegar el aplicativo usando VERCEL, pero algunas de las dependencias usadas para este proyecto son muy pesadas y la versión usada de esta herramienta no admite archivos de más de 2 GB, por lo que el despliegue de este aplicativo puede ser considerado como un trabajo pendiente, además de que la evaluación con técnicas como Precisión, Recall y F1-Score se realizaron de manera superficial.
 - **Tareas Pendientes:** US10

Burndown Chart

La elaboración de la gráfica de trabajo restante se realizó usando la herramienta de Excel en donde partíamos de una tabla en la que se establecían fechas y el tiempo de trabajo acordado para realizar por día.

Fecha	Esperado	Real
24 de Enero	250	250
25 de Enero	230	250
26 de Enero	210	245
27 de Enero	190	230
28 de Enero	170	230
29 de Enero	150	220
30 de Enero	130	200
31 de Enero	110	200
01 de Febrero	90	190
02 de Febrero	80	170
03 de Febrero	70	150
04 de Febrero	60	130
05 de Febrero	50	100
06 de Febrero	45	70
07 de Febrero	35	50
08 de Febrero	20	40

09 de Febrero	15	35
10 de Febrero	10	20
11 de Febrero	5	10
12 de Febrero	3	5
13 de Febrero	0	0

Tabla 6. Trabajo Estimado y Real

Y como resultado tenemos la siguiente gráfica que evidencia nuestro trabajo:

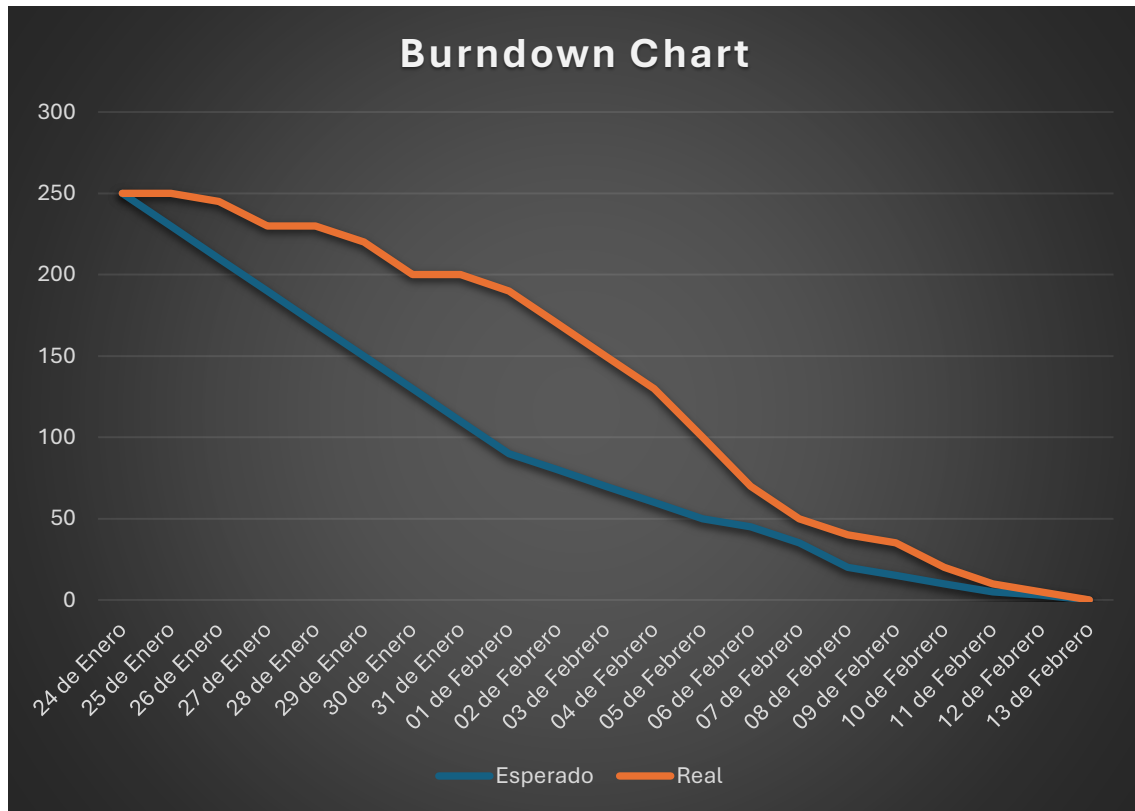


Ilustración 8. Burndown Chart

Conclusiones

1. La elaboración de un sistema RAG nos inmiscuyó en un proceso de conocimiento extenso, ya que, si bien conocíamos sobre la elaboración de sistemas de recuperación de la información tradicionales, no imaginábamos la ventaja de usar la información recuperada como contexto para un modelo de lenguaje que pueda generar respuestas resumidas y bien informadas que proporcionen a los usuarios la satisfacción de estar conversando con un sistema que entiende sus solicitudes.
2. Otorgar respuestas completas y bien informadas a preguntas específicas es muy importante en todo contexto, más aún si este amerita la depreciación de sesgos ideológicos y estancamientos en las cámaras de eco. Es por eso que en el contexto político que atraviesa el país este sistema RAG puede proporcionar una guía para que los ciudadanos se mantengan informados acerca de las candidaturas políticas presentes.

Recomendaciones

1. Para generar pruebas al sistema de recuperación de información tradicional recomendamos que en la etapa de procesamiento se obtenga un corpus amplio con un formato bien definido donde se resuma la identidad del candidato presidencial, los temas más relacionados con el mismo (se puede incluir aquí su tendencia ideológica) y que, a parte del plan de trabajo, se incluyan entrevistas que hayan tenido en medios de izquierda, derecha y centro.
2. El uso de herramientas de IA puede simplificar el proceso de extracción del corpus, pues el proceso de extracción de texto de las entrevistas y videos puede resultar laborioso y ese tiempo puede ser usado para ser más ambicioso en otras etapas del proyecto. En nuestro caso usamos la extensión HUMATA.IA para la extracción del texto de los videos en un formato JSON y luego usamos CHATGPT para diferenciar a los actores dentro de la entrevista y extraer la información correspondiente solo al candidato político.
3. Para eliminar aún más el sesgo ideológico y tener una postura más objetiva en la elaboración de este tipo de sistemas RAG se recomienda agregar al corpus los hechos históricos a favor y en contra de cada candidato, además de sus nexos con medios de comunicación, partidos políticos y empresas.
4. En la búsqueda de un LLM optimizado a nuestras necesidades descubrimos que si el corpus es manejado en inglés (tanto en su sistema de RI y el LLM) el modelo de lenguaje arroja respuestas decentes, esto también puede depender si se aplicó stemming o lematización en el proceso de limpieza del corpus.
5. Una buena capacidad de hardware puede ayudar enormemente en la optimización del tiempo en la que se generan las respuestas para algunos modelos, ya que estos suelen consumir recursos de forma directa, más aún los de la GPU, por lo que tener un buen equipo para realizar estas pruebas y despliegues sería adecuado para este tipo de sistemas RAG que busca integrar bastante información para muchos usuarios que buscan respuestas rápidas.