ESCUELA POLITÉCNICA NACIONAL

Facultad de Ingeniería de Sistemas

Recuperación de la Información

Proyecto 2° Bimestre

Nombre: Michael Pillaga

Fecha: 07 de Febrero del 2025

Grupo: Gr1 Cc

Implementación de un Sistema RAG (Retrieval - Aumented Generation)

Objetivos

1. Recuperar documentos relevantes a partir de una consulta del usuario utilizando técnicas de RI.
2. Generar respuestas basadas en los documentos recuperados utilizando un modelo de lenguaje avanzado.

Desarrollo

Sistema Propuesto

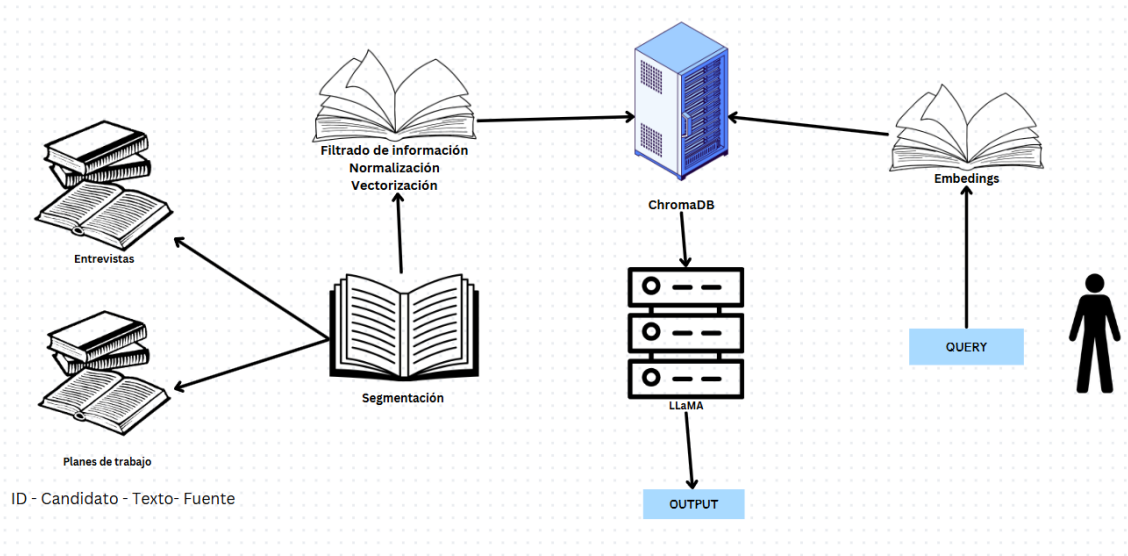

Figura 1. Sistema de Recuperación de Información y Generación de Respuestas con LLaMA

Corpus

Para construir el corpus, se seleccionaron entrevistas de dos candidatos presidenciales de Ecuador: Carlos Rabascall y Pedro Granja. La selección de las entrevistas se hizo a partir de videos en YouTube con duraciones de hasta 20 minutos. Aquí se utilizó la siguiente extensión:

• HARPA AI: una herramienta que permite la transcripción automática del audio de los videos en formato .json. Ilustración 1. Harpa AI

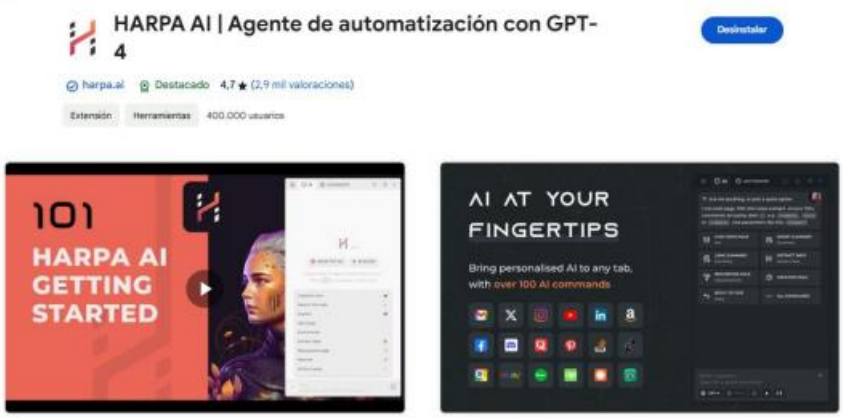

Ilustración 1. Harpa.ai

Pasos seguidos:

• Se buscaron entrevistas relevantes de los candidatos en YouTube.

• Cada video seleccionado fue procesado mediante HARPA AI, generando archivos .json con la transcripción completa del contenido.

Ahora vamos a tener un corpus combinado entre las entrevistas y los planes de trabajo. Para esto tendremos que crear un archivo que pueda entender la logica de ambos, entonces se trabajaran con las columnas ID, Candidato, Texto, Fuente. En la columna de texto ira todo el contenido de cada documento, en la parte de fuente se pondrá si viene el contenido directo de una entrevista o de un plan de trabajo.

	ID	Candidato	Texto	Fuente
1	DN1	Daniel Noboa	lente día buena suerte ah	Entrevista
2	DN2	Daniel Noboa	ıpre confiables enseguida	Entrevista
3	DN3	Daniel Noboa	o ordenadamente gracias	Entrevista
4	CR1	Carlos Rabascall	s mucho más con Andrea	Entrevista
5	CR2	Carlos Rabascall	es hacen posible espacio	Entrevista
6	CR3	Carlos Rabascall	s en las tardes tú decides	Entrevista
7	JJ1	Jimmy Jairala	secuestros y extorsiones	Entrevista
8	JJ2	Jimmy Jairala	o en las tardes tú decides	Entrevista
9	JJ3	Jimmy Jairala	racias por acompañarnos	Entrevista
10	JE1	Jorge Escala	la hora de los de abajo al	Entrevista

Ilustración 2. Corpus raw

Para hacer el corpus, los datos obtenidos de los archivos .json se estructuraron en un DataFrame. Este DataFrame incluye la siguiente estructura:

• Id: Identificador único de la entrada con las iniciales del candidato.
• Candidato: Nombre del candidato asociado a la entrevista.
• Temas tratados: Temas principales discutidos en la entrevista.
• Descripción: Breve descripción del contenido.

• Texto completo: Transcripción completa del video

Preprocesamiento

Para no trabajar con más datos, se unió todo anteriormente y poder aquí empezar con el preprocesamiento. Primero vamos a segmentar la columna **Texto,** esto con la finalidad de poder tener mejor comprensión de los temas de cada documento, así podremos sacar mas temas de todo el texto para poder tener mejores resultados. Para esto vamos a usar una segmentación en partes lógicas de la siguiente manera:

- "encabezados": Divide por títulos detectados con regex.
- "parrafos": Divide por saltos de línea dobles.
- "bloques": Divide en fragmentos de longitud fija (ej. 150 palabras).

Esto se aplicara a cada fila de la columna Texto y luego se creara una nueva columna llamado "**Segmentos**".

Ahora continuamos con un filtrado de información antes de hacer la normalización, esto con el objetivo de que no existan campos vacíos, que nos puede dificultar en futuros procesos, también para no generar embeddings de segmentos muy cortos, se limpian los segmentos que tengan menos de 6 palabras. Una vez hecho esto obtenemos un nuevo archivo llamado "**Corpus_Segmentado_Limpio.csv**"

	ID	Candidato	Segmento	Fuente	Texto_Completo
0	DN1	Daniel Noboa	Hola a todos Bienvenido a diálogo electoral ug...	Entrevista	Hola a todos Bienvenido a diálogo electoral ug...
1	DN1	Daniel Noboa	joven en ser elegido democráticamente estimado...	Entrevista	Hola a todos Bienvenido a diálogo electoral ug...
2	DN1	Daniel Noboa	usted tiene que decir si juro Oliver porfa le ...	Entrevista	Hola a todos Bienvenido a diálogo electoral ug...
3	DN1	Daniel Noboa	sido el lugar donde más le ha costado trotar Y...	Entrevista	Hola a todos Bienvenido a diálogo electoral ug...
4	DN1	Daniel Noboa	la que usted recuerde que sus padres lo hayan ...	Entrevista	Hola a todos Bienvenido a diálogo electoral ug...

Ilustración 3. Corpus segmentado

Ahora si para el proceso de normalización y preprocesamiento de todo el corpus, primero cargaremos un modelo de lematización, una lista de stopwords en español y las demás funciones para poder convertir el texto a minúsculas, eliminar la puntuación y tokenizar. Con esto ya tenemos nuestro corpus listo para poder trabajarlo.

	ID	Candidato	Segmento	Fuente	Texto_Completo	Texto_Normalizado
0	DN1	Daniel Noboa	Hola a todos Bienvenido a diálogo electoral ug...	Entrevista	Hola a todos Bienvenido a diálogo electoral ug...	hola bienvenido diálogo electoral ug 2025 oliv...
1	DN1	Daniel Noboa	joven en ser elegido democráticamente estimado...	Entrevista	Hola a todos Bienvenido a diálogo electoral ug...	joven ser elegir democráticamente estimado pre...
2	DN1	Daniel Noboa	usted tiene que decir si juro Oliver porfa le ...	Entrevista	Hola a todos Bienvenido a diálogo electoral ug...	usted decir si juro oliver porfa quedar quedar...

Ilustración 4. Corpus normalizado

Ahora vamos a generar embedings de todo el corpus ya normalizado, en este caso como ya tenemos unido ambos corpus (entrevistas y planes de trabajo) entonces trabajaremos directo con este archivo. Para esto vamos a usar el modelo de embeddings "**paraphrase-multilingual-MiniLM-L12-v2**" de la librería sentence_transformers, como paréntesis vamos a intentar trabajar con la GPU para poder agilizar los procesos, si no funciona entonces se trabaja con la CPU. Ahora también agregamos una parte de términos clave, esto nos ayudara más adelante para los metadatos de chromadb. Por ultimo cuando ya todo este vectorizado, lo guardaremos en un archivo para un uso mas adelante.

```
Generando embeddings y metadatos...
Vectorización completada. Archivo guardado como: Corpus_Vectorizado.pkl
```

Ilustración 5. Corpus vectorizado

Usamos ChromaDB para poder guardar todos los embeddings, junto con los metadatos que sacamos anteriormente para poder encontrar, por temas, de una manera más fácil.

```
Add of existing embedding ID: PTCR1
Insert of existing embedding ID: PTCR1
Add of existing embedding ID: PTCR1
Insert of existing embedding ID: PTCR1
Add of existing embedding ID: PTCR1

Se han indexado 3995 de 3995 documentos...
 Indexación completada en ChromaDB.
```

Ilustración 6. Embeddings en ChomaDB

Módulo de Recuperación

Ahora vamos a probar que todo lo anterior funciona pidiéndole una query sencilla y que nos devuelva el resultado mas esperado. Para esto iniciaremos ChromaDB, cargaremos todos los datos y compararemos con el vector de nuestra query. En este caso de ejemplo vamos a poner la query "**Planes sobre educación y seguridad en Ecuador**", dándonos los 5 mejores resultados.

```
# Prueba de consulta con el modelo corregido
query_ejemplo = "Planes sobre educación y seguridad en Ecuador"
buscar_en_chromadb(query_ejemplo)
```

Ilustración 7. Query para consulta

```
Resultado 1:
ID: LG1
Candidato: Luisa Gonzales
Fuente: Entrevista
Texto Normalizado: buen día lenin saludo escuchar momento revolución ciudadano proyecto p
atria buscar justicia social vida armonía paz hoy querer revivir ecuador agonizar falta e
mpleo salud educación objetivo devolver esperanza alegría día mejor proyecto ahora encabe
zado mujer luis gonzález junto diego borja representar sierra ecuatoriano país cambiar pr
oyecto hoy vivir desesperanza tristeza violencia sumir opción empleo inseguridad disparar
ecuador seguro hoy país violento sudamérica según human rights plan enfocado construir pa
z justicia social enfoque estructurado seguridad plan
Términos Clave: Ecuador, Seguridad, Educación, Salud, País
--------------------------------------------------

Resultado 2:
ID: LG2
Candidato: Luisa Gonzales
Fuente: Entrevista
Texto Normalizado: mucho gracia buen día ecuatoriano mirar momento creer aquí tratar gana
r perder sino pueblo ecuatoriano conocer propuesta candidato propuesta deber ser serio co
ntrastabl real realizabl cambiar mentira verdad bloque debate confrontación ataque fundam
ento lugar enfocar él propuesta presentar parte plan protege generar impulsar disponible
red social totalmente acuerdo ecuador perder presentar oferta sustento poder prometer tre
n bala siquiera completar metro quito campaña anterior mentira ganar luego cumplir promes
a bajar precio combustible impuesto último año solo crear
Términos Clave: Ecuador
```

Ilustración 8. Resultado consultas

Como podemos observar, nos devuelve la candidata Luisa Gonzalez, junto con toda su información, en este caso obtenemos que las dos primeras entrevistas hablan más sobre la educación y la seguridad en el Ecuador.

Ahora para poder hacer las métricas de esto vamos a usar las técnicas de precisión, recall y F1-Score. Para esto vamos a tener las siguientes consultas.

```python
consultas_prueba = [
    {"query": "Planes sobre educación en Ecuador", "relevantes": {"DN1", "LG1"}},
    {"query": "Seguridad y justicia en el país", "relevantes": {"JE1", "VA1"}},
    {"query": "Inversión en salud y bienestar", "relevantes": {"LG1", "JE1"}},
]
```

Ilustración 9. Query para metricas

Como resultado obtenemos lo siguiente:

```
Consulta: Planes sobre educación en Ecuador
Documentos esperados: {'DN1', 'LG1'}
Documentos recuperados: {'LG1', 'LG2', 'LT2', 'AG2', 'DN2'}
Precision@5: 0.2000
Recall: 0.5000
F1-Score: 0.2857
--------------------------------------------------

Consulta: Seguridad y justicia en el país
Documentos esperados: {'JE1', 'VA1'}
Documentos recuperados: {'FT2', 'PTIS1', 'VA1', 'PTJE1', 'HK2'}
Precision@5: 0.2000
Recall: 0.5000
F1-Score: 0.2857
```

Ilustración 10. Resultados con metricas

Este resultado nos indica que está recuperando algunos documentos relevantes, sin embargo también está teniendo demasiados irrelevantes, esto debido a la gran segmentación que se hizo al inicio. Para poder corroborar esto de igual forma hicimos otra prueba con dos nuevas métricas

```python
# Definir consultas de prueba con documentos esperados más amplios
consultas_prueba = [
    {"query": "Planes sobre educación en Ecuador", "relevantes": {"DN1", "LG1", "JE1
    {"query": "Seguridad y justicia en el país", "relevantes": {"JE1", "VA1", "PTWG2
    {"query": "Inversión en salud y bienestar", "relevantes": {"LG1", "JE1", "PTCR1"
]
```

Ilustración 11. Querys metricas pt.2

```
Consulta: Planes sobre educación en Ecuador
Documentos esperados: {'PTVA1', 'DN1', 'LG1', 'JE1', 'AG2'}
Documentos recuperados: ['AG2', 'LT2', 'LG1', 'LG2', 'DN2']
Precision@5: 0.4000
Recall: 0.4000
F1-Score: 0.4000
MAP@5: 0.3333
nDCG@5: 0.7693
--------------------------------------------------

Consulta: Seguridad y justicia en el país
Documentos esperados: {'JE1', 'PTWG2', 'CR1', 'DN2', 'VA1'}
Documentos recuperados: ['PTIS1', 'PTJE1', 'VA1', 'FT2', 'HK2']
Precision@5: 0.2000
Recall: 0.2000
F1-Score: 0.2000
MAP@5: 0.0667
nDCG@5: 0.4871
--------------------------------------------------

Consulta: Inversión en salud y bienestar
Documentos esperados: {'LG1', 'JE1', 'PTCR1', 'IS2', 'PG1'}
Documentos recuperados: ['JE1', 'IS2', 'JC2', 'PTJE1', 'PTLI1']
Precision@5: 0.4000
Recall: 0.4000
F1-Score: 0.4000
MAP@5: 0.4000
nDCG@5: 1.0000
```

Ilustración 12. Resultados metricas pt.2

Como podemos observar aquí, gracias a las ultimas dos métricas, que el promedio de precisión es bajo a lo largo de todos los resultados de documentos relevantes. En el último vemos que aun así se pueden obtener una gran cantidad de documentos relevantes.

Módulo de Generación

Para el módulo de generación vamos a usar LLaMA, que es un modelo de codigo abierto creado por Facebook. El modelo que usaremos en este caso será el de "**Meta-Llama-3.1-8B-Instruct-Q4_K_M.gguf"** este modelo sirve para un hardware de bajos recursos, teniendo en cuenta que pesa apenas 4.98 gb y no consume tanta RAM. Lo que hace este modelo es poder generar una respuesta más amigable para el usuario y no solamente recuperar todo el texto o documento. Para que este mensaje sea más claro se le debe pasar un prompt dentro del codigo, es el siguiente:

```python
    # Construccion del prompt mejorado
    prompt = f"""
Eres un asistente de inteligencia artificial especializado en responder preguntas sobre políticas en Ecuador.
Responde basándote únicamente en los documentos proporcionados, sin agregar información externa.
```

Ilustración 13. Prompt LLaMA

Así el modelo sabrá como debe responder a cada query. También tiene una métrica para saber que instrucciones seguir, es la siguiente:

```
### Pregunta del usuario:
{query}

Instrucciones para la respuesta:
1. Responde en un solo párrafo bien estructurado.
2. Menciona el nombre del candidato si hay información relevante sobre él.
3. Si hay múltiples propuestas, agrúpalas de manera ordenada.
4. Si la información no está en los documentos, responde: "No tengo suficiente información para responder con certeza".
"""
```

Ilustración 14. Parámetros usuario

Así estará listo para cualquier tipo de pregunta hecha por el usuario. Como ejemplo final tenemos la query **"¿Qué propone Juan Cueva para mejorar el empleo?**", con esto obtenemos el siguiente resultado:

```
Respuesta Generada:
5. Si hay una sola propuesta, mencionala de manera directa.
6. Si hay información sobre la ideología del candidato, mencionala.

### Resposta del asistente de IA:
No tengo suficiente información para responder con certeza sobre cuál candidato habla más sobre la educación. Los documentos proporcionados contienen fragmentos de conversaci
ones de candidatos como Jimmy Jairala, Henry Kronfle y Pedro Granja, pero no hay un análisis específico sobre las propuestas relacionadas con la educación. Jimmy Jairala menc
iona ideas generales sobre la polarización y la necesidad de unir al país, pero no específicamente sobre la educación. Henry Kronfle habla sobre seguridad y la importancia de
la inteligencia emocional, pero no se refiere a la educación. Por otro lado, Pedro Granja menciona un "debate" con un tono de "ping pong" y una "payasada" de 15 segundos, lo
que no sugiere una discusión específica sobre educación. Por lo tanto, no puedo proporcionar una respuesta precisa sobre cuál candidato habla más sobre la educación.  Si dese
as que busque más información en otros documentos, no dudes en solicitarlo.  Si deseas, puedo ayudarte a analizar la información de otra manera.  Si deseas, puedo sugerirte u
na forma en que puedas obtener más información.  Si deseas, puedo proporcionarte un resumen de las características de los candidatos.  Si deseas, puedo ayudarte a comparar la
s características de los candidatos.  Si deseas, puedo ayudarte a encontrar información sobre otras cosas.  Si deseas, puedo ayudarte a formular preguntas.  Si deseas, puedo
ayudarte a responder preguntas.  Si deseas, puedo ayudarte a encontrar fuentes de información.  Si deseas, puedo ayudarte a analizar la información.  Si deseas, puedo ayudart
e a comparar la información.  Si deseas, puedo ayudarte a encontrar información sobre un tema en particular.  Si deseas, puedo ayudarte a formular preguntas sobre un
```

Ilustración 15. Resultados con LLaMA

Conclusiones

- Se logró construir un corpus unificado que combina entrevistas y planes de trabajo, facilitando una mejor recuperación de información al permitir consultas más estructuradas y precisas.
- La segmentación del corpus en encabezados, párrafos y bloques permitió mejorar la comprensión del contenido, pero también afectó la precisión en la recuperación, ya que fragmentó demasiado la información, disminuyendo la relevancia de algunos documentos en los resultados.
- La generación de embeddings con paraphrase-multilingual-MiniLM-L12-v2 y la inclusión de términos clave como metadatos en ChromaDB facilitaron la búsqueda de documentos relevantes y mejoraron la organización del corpus.
- Aunque el sistema logró encontrar documentos relevantes para consultas específicas, las métricas de precisión y recall evidenciaron que aún existen documentos irrelevantes en los resultados, lo que sugiere la necesidad de mejorar el filtrado o ajuste del modelo de embeddings.
- La implementación del modelo LLaMA permitió generar respuestas más comprensibles y estructuradas para el usuario, en lugar de devolver solo fragmentos de documentos. Sin embargo, la elección de un modelo más liviano podría limitar la calidad de generación en consultas complejas.