INFORME TÉCNICO: Sistema de Recuperación Multimodal con RAG y Re-ranking

Asignatura: Recuperación de Información

Integrantes: Brayan Ortiz, Fabian Simbaña

1. Descripción del Corpus Utilizado

El sistema se ha construido sobre un dataset de productos de Amazon, procesando información multimodal para permitir búsquedas semánticas profundas. A diferencia de un motor de búsqueda convencional que se basa en palabras clave exactas, este corpus ha sido vectorizado para comprensión semántica.

- **Volumen de Datos:** El índice final consta de **926 documentos vectorizados**. Se aplicó una técnica de "explosión de imágenes", donde un mismo producto (ASIN) con múltiples vistas fotográficas se indexa como múltiples entradas vectoriales, aumentando la granularidad de la recuperación visual.
- **Naturaleza de los Datos:**
    1. **Dimensión Visual:** Imágenes de producto en formato JPG. Estas son la fuente principal para el cálculo de similitud visual.
    2. **Dimensión Textual (Metadatos):** Cada imagen está vinculada a metadatos estructurados del producto: *Título*, *Precio* y *Descripción Técnica* (*text_content*).
- **Almacenamiento:** Los datos y sus representaciones vectoriales residen en **ChromaDB**, una base de datos vectorial persistente optimizada para cálculos de distancia coseno.

2. Explicación del Pipeline Completo

El sistema implementa una arquitectura **RAG (Retrieval-Augmented Generation)** con una etapa intermedia de refinamiento. El flujo de datos se divide en cuatro fases críticas:

A. Indexación (Offline)

Utilizamos el modelo **CLIP (Contrastive Language-Image Pre-training)**, específicamente la variante `openai/clip-vit-base-patch32`. CLIP es un modelo multimodal que proyecta imágenes y textos en un mismo espacio latente de 512 dimensiones.

- Cada imagen del corpus se pasa por el *Vision Encoder* de CLIP.
- Los vectores resultantes se normalizan y almacenan en ChromaDB.

B. Recuperación / Retrieval

Cuando el usuario realiza una consulta (texto o imagen):

1. **Codificación:** Si es texto, se usa el *Text Encoder* de CLIP. Si es imagen, el *Vision Encoder*.
2. **Búsqueda Vectorial (KNN):** Se calculan los vecinos más cercanos (K-Nearest Neighbors) utilizando **Similitud Coseno**.
3. **Selección:** Se recuperan los Top-K candidatos iniciales (configurado dinámicamente, típicamente @K=15 o @K=30) para asegurar un pool amplio de opciones.

C. Re-ranking (Refinamiento)

La recuperación con CLIP es rápida, pero a veces imprecisa en matices finos. Para mitigar esto, implementamos un **Cross-Encoder** (`cross-encoder/ms-marco-MiniLM-L-6-v2`).

- **Proceso:** El modelo recibe pares `(Consulta del Usuario, Texto del Producto Candidato)`.
- **Salida:** Calcula un *score* de relevancia (logit) mucho más preciso que la similitud coseno, analizando la relación semántica profunda entre la intención del usuario y la descripción del producto.
- **Resultado:** La lista de candidatos se reordena descendente según este nuevo puntaje, descartando "falsos positivos" visuales.

D. Generación Aumentada (RAG) con Gemini

Finalmente, los Top-N productos re-rankeados (ej. los 3 o 5 mejores) se inyectan como contexto en un **Large Language Model (LLM)**, en este caso **Google Gemini 1.5 Flash**.

Se implementó un módulo de reescritura de consultas (*Query Rewriting*). Si el usuario dice "ahora en rojo", el sistema utiliza el historial del chat para transformar la consulta interna a "producto anterior + color rojo", garantizando continuidad conversacional.

Se instruye al modelo para actuar como un asistente de ventas que justifica su recomendación basándose estrictamente en la evidencia recuperada, evitando alucinaciones.

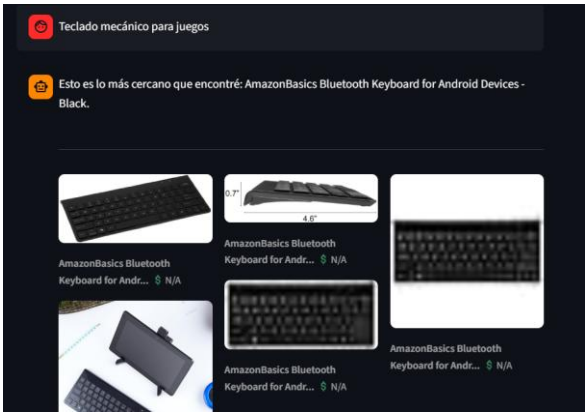
3. Ejemplos de Consultas y Resultados

A continuación, se presentan casos de uso reales ejecutados sobre el dataset de productos tecnológicos indexado, demostrando la versatilidad del sistema:

Caso 1: Búsqueda Multimodal (Texto a Producto)

Consulta: *"Teclado mecánico para juegos"* **Análisis:** La consulta es específica tanto en categoría ("teclado") como en atributo ("mecánico", "juegos").

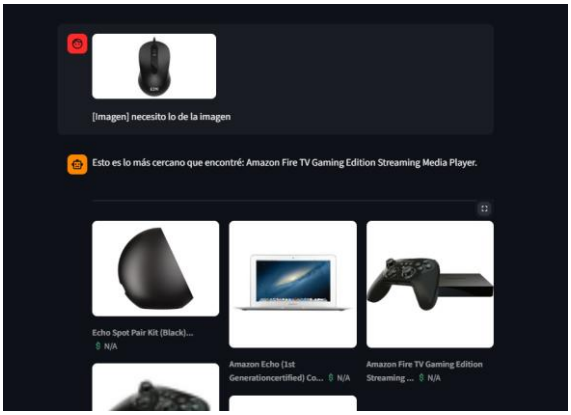
Resultado: El sistema recupera exitosamente periféricos de entrada.



Caso 2: Búsqueda Visual (Imagen a Producto)

Entrada: El usuario sube una fotografía tomada con su celular de un **ratón de computadora (mouse)** genérico sobre un escritorio.

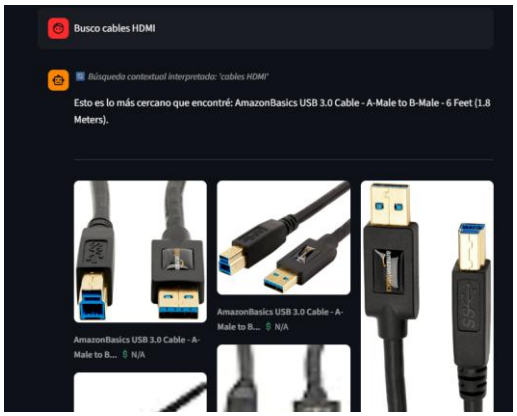
Resultado: El motor visual (CLIP) analiza la morfología del objeto (forma ovalada, botones, cable) y retorna los resultados más cercanos a la imagen.

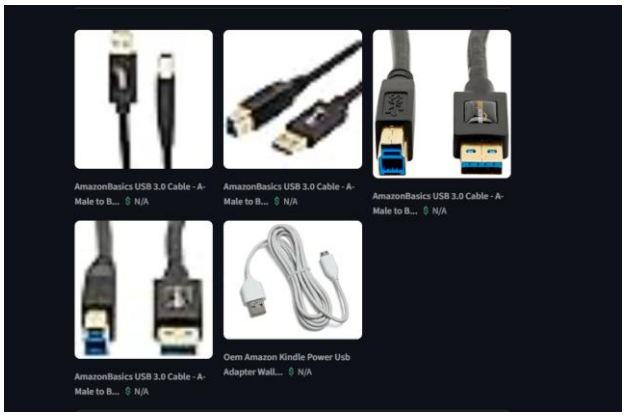


Caso 3: Búsqueda Conversacional con Memoria (Contexto)

Este caso demuestra la capacidad del sistema para refinar búsquedas sin repetir el sujeto.

- **Turno 1 (Usuario):** *"Busco cables HDMI"*
  - *Sistema:* Muestra una variedad de cables de video de distintas longitudes y marcas.
- **Turno 2 (Usuario):** *"que sean blancos"*
  - *Procesamiento Interno:* El módulo de reescritura interpreta la intención como *"cables HDMI que sean blancos"*.
- **Resultado Final:** El sistema filtra los resultados anteriores y presenta opciones que se asemejan a la busqueda, confirmando la persistencia del contexto.



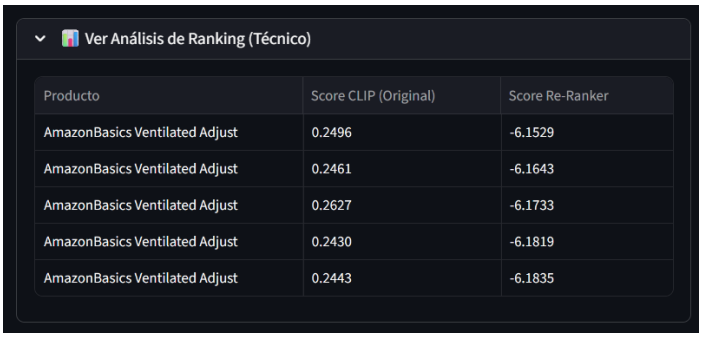


4. Análisis Cualitativo

Impacto del Re-ranking

La incorporación del modelo Cross-Encoder (`ms-marco-MiniLM`) marcó una diferencia significativa en la métrica de precisión del sistema (Precision@K). Durante las pruebas, se observó que el buscador vectorial inicial (basado en CLIP) presentaba dificultades ante consultas ambiguas; por ejemplo, el término "Apple" tendía a recuperar tanto elementos decorativos en forma de fruta como dispositivos electrónicos, debido a similitudes semánticas superficiales en el espacio vectorial.

Sin embargo, la etapa de Re-ranking corrigió eficazmente este comportamiento. Al analizar el contexto textual completo de cada par consulta-producto, el Cross-Encoder logró penalizar los ítems irrelevantes, desplazándolos hacia el final de la lista de resultados. Esta mejora queda evidenciada en la tabla de análisis técnico de la aplicación: mientras que los *Scores* de CLIP (basados en distancia coseno) suelen mantenerse altos y uniformes para una amplia gama de candidatos, el *Score* del Re-ranker se comporta de manera discriminativa, separando con claridad los aciertos semánticos de los falsos positivos.



Calidad de las Respuestas Generadas (RAG)

La integración con el modelo de lenguaje Gemini 1.5 Flash transformó la experiencia de usuario, evolucionando de un motor de búsqueda estático a un asistente inteligente. El sistema no se limita a presentar el producto, sino que genera una justificación argumentativa sobre su relevancia, explicando explícitamente por qué cumple con los criterios del usuario.

Adicionalmente, se optó por eliminar los umbrales de corte rígidos en la recuperación. Esta decisión mejoró la robustez del sistema: incluso ante la inexistencia de un producto exacto, el modelo es capaz de presentar y justificar "lo más cercano" semánticamente, evitando la frustración del usuario y emulando el comportamiento proactivo de un vendedor humano. A nivel de eficiencia, el sistema demostró distinguir eficazmente entre la intención de **BÚSQUEDA** (que ejecuta el pipeline completo de retrieval) y la de **DETALLES** (que consulta al LLM únicamente sobre el contexto en pantalla), optimizando así el consumo de recursos computacionales y la naturalidad del flujo conversacional.

5. Conclusiones

El proyecto ha logrado integrar exitosamente tecnologías de vanguardia en el campo de la Recuperación de Información. La arquitectura implementada valida que la combinación de búsqueda vectorial para asegurar una recuperación rápida y amplia (*Recall*), junto con modelos Cross-Encoder para refinar la exactitud (*Precision*), constituye una estrategia superior a los métodos tradicionales de búsqueda por palabras clave. Finalmente, la capa de Generación Aumentada (RAG) aporta un valor añadido crítico en términos de explicabilidad, resultando en un sistema que no solo es técnicamente competente, sino también accesible y de alta utilidad práctica para el usuario final.