Proyecto 2

Mapeando el Caos: Indexación y Organización de Datos No Estructurados para Datos Multimedia.

1- Enunciado

En grupos de hasta cinco integrantes, deberán desarrollar una aplicación de base de datos multimedia, como por ejemplo: un sistema de recomendación de productos para e-commerce, un sistema de recomendación musical, o un sistema de detección de plagio en audio, entre otros. Esta aplicación debe integrar de manera eficiente técnicas de indexación y búsqueda por similitud, articulándose con el desarrollo realizado en la primera entrega, con el fin de construir una base de datos multimodal funcional.

El objetivo del proyecto es comprender y aplicar algoritmos de búsqueda y recuperación de información basados en contenido. El desarrollo se divide en dos componentes principales:

a) La construcción óptima de un índice invertido para búsqueda y recuperación de información en documentos de texto, y
b) La implementación eficiente de un índice para recuperación de imágenes o audio, utilizando descriptores locales.

Para validar el funcionamiento del sistema, se deberán emplear datasets reales, preferentemente obtenidos de plataformas como Kaggle o GitHub. El proyecto deberá demostrar eficiencia en las operaciones, una estructura de código clara y organizada, así como incluir una breve documentación técnica que describa el diseño, las decisiones adoptadas y los resultados obtenidos.

2- Construcción del Índice Invertido Textual (Full-Text Search)

Implementar el índice invertido usando el modelo de recuperación por ranking para consultas de texto libre. Considere los siguientes pasos generales:

a) Preprocesamiento:
Se usarán los campos textuales concatenados para formar un solo texto por cada fila de la tabla.

- Tokenization
- Filtrar Stopwords
- *Eliminar signos innecesarios*
- Reducción de palabras *(Stemming)*

b) Construcción del Índice

- Estructurar el índice invertido para guardar los pesos TF-IDF.
- Calcular una sola vez la norma de cada documento y guardarlo para reutilizarlo al momento de aplicar la similitud de coseno.
- **Construcción del índice en memoria secundaria para grandes colecciones de datos.**
  - SPIMI: Single-pass in-memory indexing

- Debe explicar en el informe el funcionamiento del algoritmo gráficamente y como fue implementado.

c) Consulta

- La consulta es una frase en lenguaje natural.
- El scoring se obtiene aplicando la similitud de coseno sobre el índice invertido en **memoria secundaria**.
  - Evitar cargar todo el índice en la RAM.
- La función de recuperación debe retornar el top-k de documentos que se aproximen a la consulta.

3- Indexación de descriptores locales (Multimedia Database)

a) Extracción de características

Se usará librerías existentes para extraer un vector característico de cada objeto multimedia (imagen o audio). De preferencia, los objetos multimedia deben estar asociados a los datos tabulares del apartado anterior.

- Tutorial para extraer características en imágenes:
  - SIFT Algorithm
  - Inception v3 CNN
  - **resnet50**

- Tutorial para extraer características en audio:
  - Se puede usar MFCC (Mel-Frequency Cepstral Coefficients) o espectrogramas
  - MFCC Feature Extraction from Audio

b) Construcción del diccionario visual o acústico (Codebook)

- Se recopilan los descriptores locales extraídos de múltiples objetos (imágenes o audios) para conformar un único conjunto de datos.
- Se aplica el algoritmo K-Means para agrupar los descriptores en clusters, donde cada cluster representa un “visual Word” (en el caso de imágenes) o un “acoustic word” (en el caso de audio).
- El centroide de cada cluster se considera un codeword, y el conjunto de estos codewords conforma el diccionario visual o acústico.

c) KNN Secuencial

- Representa cada objeto (imagen o audio) mediante un histograma de visual words o acoustic words, utilizando el diccionario previamente construido.
- Para el objeto de consulta, genera su correspondiente histograma de visual/acoustic words.
- Aplica técnicas de ponderación como TF-IDF para asignar un peso a cada palabra (visual o acústica), según su importancia relativa en el conjunto de datos.
- Calcula la similitud de coseno entre el histograma del objeto de consulta y los histogramas almacenados.
- Recupera los K objetos más similares en función de la similitud obtenida. Debe estar optimizando con una cola de prioridad (heap) para mantener de forma eficiente los K resultados más relevantes durante la búsqueda.

d) KNN con Indexación Invertida

- Para cada objeto (imagen o audio), asigna sus descriptores locales a los visual words o acoustic words más cercanos, utilizando el diccionario previamente generado.
- Construye una estructura de índice invertido, donde cada palabra (visual o acústica) referencia los objetos en los que aparece.
- Aplica técnicas de ponderación como TF-IDF para estimar la relevancia de cada palabra en relación con el objeto y el conjunto de datos.
- El algoritmo de búsqueda debe seguir un enfoque análogo al utilizado en los índices invertidos textuales, adaptado al dominio visual o acústico, permitiendo una recuperación eficiente de los objetos más relevantes.

4- FrontEnd

a) Para búsqueda en texto:

Se debe desarrollar una aplicación que permita interactuar con las principales operaciones del índice invertido textual. La interfaz debe incluir las siguientes funcionalidades:

- Entrada de consultas textuales mediante una sintaxis tipo SQL, o bien una sintaxis personalizada definida por el grupo.
  - **Ejemplo** de consulta con sintaxis tipo SQL:

```
SELECT title, artist, lyric
FROM Audio
WHERE lyric @@ 'amor en tiempos de guerra'
LIMIT 10;
```

- Opción para especificar la cantidad de documentos a recuperar (**Top-K**).
- Presentación de resultados de manera clara y amigable para el usuario:
  - Mostrar los campos solicitados (por ejemplo, título, artista, fragmento de letra).
  - Indicar el **tiempo de respuesta** de la consulta.

b) Para base de datos multimedia (imágenes o audio)

Se debe construir una interfaz que permita realizar consultas mediante un objeto multimedia (imagen o audio) y visualizar los resultados más similares. Las funcionalidades mínimas son:

- Opción para cargar el archivo multimedia de consulta (por ejemplo, imagen) y ejecutar la búsqueda basada en contenido.
- Posibilidad de utilizar una sintaxis tipo SQL o definida por el grupo para consultas multimodales.
  - **Ejemplo** (consulta extendida):

```
SELECT id, title
FROM Multimedia
WHERE image_sim <-> 'D:\imagenes\query.jpg'
LIMIT 10;
```

- Presentación interactiva de los resultados de la búsqueda, incluyendo:
  - Visualización de los objetos similares (imágenes o audios con sus respectivos metadatos).
  - Asociación con resultados textuales relacionados (por ejemplo, título o descripción).

- Indicación del **tiempo de ejecución** de la consulta.

5- Experimento y Evaluación de Desempeño:

a) Búsqueda en texto:

Evalúe el rendimiento de su implementación del índice invertido comparándolo con los resultados obtenidos utilizando PostgreSQL. Para ello:

- Diseñe consultas equivalentes en ambos sistemas y mida el **tiempo de respuesta** y la **calidad de los resultados** (¿Se obtiene los mismo resultados?).
- Investigue y explique cómo PostgreSQL realiza la recuperación de información textual:
  - El tipo de índice que utiliza (por ejemplo, GIN o GiST).
  - La función de similitud o ranking que emplea (ts_rank, ts_rank_cd, entre otras).
  - Cómo gestiona el procesamiento de consultas en campos tsvector y tsquery.

	MyIndex	PostgreSQL
N = 1000		
N = 2000		
N = 4000		
N = 8000		
N = 16000		
N = 32000		
N = 64000		
…		

b) Búsqueda en bases de datos multimedia (imágenes o audio)

Ejecute y compare los algoritmos **KNN secuencial** y **KNN con indexación invertida** sobre una colección de objetos multimedia de tamaño N, evaluando su eficiencia en función del tiempo de ejecución.

Además, compare los resultados con una implementación en **PostgreSQL** utilizando extensiones diseñadas para búsquedas vectoriales. Considere los siguientes aspectos:

- La **eficiencia** en consultas de similitud sobre vectores de características (por ejemplo, descriptores de imagen o audio).
- El impacto de la **dimensionalidad** en el rendimiento de los índices (considerar la **maldición de la dimensionalidad**).

Herramientas recomendadas para la comparación:

1. **pgVector**: Extensión de PostgreSQL para búsqueda por similitud sobre vectores (open-source).
2. **Faiss**: Librería desarrollada por Facebook para búsqueda eficiente de vectores, con soporte para CPU/GPU y algoritmos como **HNSW**.

	KNN-Secuencial	KNN-Indexado	KNN- PostgreSQL
N = 1000			
N = 2000			
N = 4000			
N = 8000			
N = 16000			
N = 32000			
N = 64000			

* Mantener el valor de K = 8.

4. Datasets

- Fashion Product Images Dataset (kaggle.com)
- Audio features and lyrics of Spotify songs Dataset (kaggle.com)
- FMA: A Dataset For Music Analysis (github.com)

5. Entregable

- Los alumnos formaran grupos de máximo de cinco integrantes.
- El proyecto estará alojado enteramente en GitHub.
- Trabajar de forma colaborativa, se considerará para su nota individual.
  - Incluir en el informe un cuadro de actividades por integrante en Project Boards.
- En el Canvas subir solo el enlace público del proyecto.
- La fecha límite de entrega es el _____.

6. Informe del proyecto y Rúbrica de evaluación

El proyecto debe documentarse adecuadamente mediante un informe técnico conciso y bien redactado, acompañado por un repositorio en GitHub que contenga el código, ejemplos de uso y documentación básica.

Requisitos Generales

- Inclusión de un archivo **README** o una **Wiki** en GitHub con instrucciones claras de ejecución.
- Cuidado en la **ortografía**, **redacción técnica** y **consistencia de los párrafos**.
- El informe no debe ser extenso, pero sí debe cubrir **todos los aspectos relevantes de la implementación**.

Estructura del Informe y Rúbrica

Sección	Contenido esperado	Puntaje
[1 punto] Introducción	- Descripción del dominio de datos (texto, imágenes o audio). - Justificación de la necesidad de una **base de datos multimodal** para tareas de recuperación por contenido.	1

[7 puntos] Backend – Índice Invertido para Texto	- Construcción del índice invertido en memoria secundaria. - Ejecución eficiente de consultas utilizando **Similitud de Coseno**. - Explicación del mecanismo de construcción de índices invertidos en **PostgreSQL**.	7
[6 puntos] Backend – Índice Invertido para Descriptores Locales	- Descripción del proceso de construcción del **Bag of Visual Words / Acoustic Words**. - Diseño de la técnica de indexación utilizada para organizar los descriptores. - Implementación de la búsqueda KNN sobre los histogramas (secuencial e indexada). - Análisis del impacto de la **maldición de la dimensionalidad** y estrategias para mitigarla.	6
[4 puntos] Frontend	- Diseño y usabilidad de la interfaz gráfica (GUI). - Inclusión de un mini-manual de usuario. - Capturas de pantalla que evidencien la funcionalidad del sistema. - Comparación visual con otras soluciones similares (opcional).	4
[2 puntos] Experimentación	- Presentación de resultados mediante **tablas y gráficos comparativos**. - Análisis crítico de los resultados (rendimiento, precisión, escalabilidad).	2

Indicaciones adicionales:

- Incluir **diagramas arquitectónicos o ilustrativos** que faciliten la comprensión del flujo del sistema.
- Subir al repositorio de GitHub todo el material necesario para su correcta ejecución y revisión.
- La **presentación final del proyecto se realizará en clase**, siendo evaluada en función de su funcionamiento, claridad expositiva y originalidad.