**Análisis de complejidad Reto #3**

**Requerimiento 1:**

**n\_1 =** Cantidad de estructuras

**n\_2 =** Cantidad de reproducciones

**n\_3** = Tamaño del árbol según característica

**Complejidad temporal:** O(k) + O(2log(n\_3)) + O(n\_1) \*O(n\_2)

**Requerimiento 2:**

Iván Camilo Ballén Méndez

**n\_1** = Tamaño del árbol según característica ‘energy’

**n\_2** = Tamaño del árbol según característica ‘danceability’

**n\_3 =** Cantidad de estructuras para la característica ‘energy’

**n\_4 =** Cantidad de tracks para la característica ‘energy’

**n\_5 = =** Cantidad de estructuras para la característica ‘danceability’

**n\_6 = =** Cantidad de tracks para la característica ‘danceability’

**Complejidad temporal:** O(k) + O(2log(n\_1)) + O(2log(n\_2)) + O(n\_3) \* O(n\_4) + O(n\_5) \* O(n\_6)

**Requerimiento 3:**

María José Sáenz Rodríguez

**n\_1** = Tamaño del árbol según característica ‘instrumentalness’

**n\_2** = Tamaño del árbol según característica ‘tempo’

**n\_3 =** Cantidad de estructuras para la característica ‘instrumentalness’

**n\_4 =** Cantidad de tracks para la característica ‘instrumentalness’

**n\_5 = =** Cantidad de estructuras para la característica ‘tempo’

**n\_6 = =** Cantidad de tracks para la característica ‘tempo’

**Complejidad temporal:** O(k) + O(2log(n\_1)) + O(2log(n\_2)) + O(n\_3) \* O(n\_4) + O(n\_5) \* O(n\_6)

**Requerimiento 4:**

**n\_1 =** Tamaño del árbol tempo

**n\_2=** Cantidad de géneros

**n\_3=** Cantidad de estructuras

**n\_4=** Cantidad de tracks en el mapa completo

**n\_5=** Cantidad de artistas en el mapa de trabajo

**Complejidad temporal**: O(k) **+** O(2log(n\_1)) \* O(n\_2) **+** O(n\_3) \* O(n\_4) \* O(n\_5)

**Requerimiento 5:**

**Complejidad temporal:**

**n\_1 =** Cantidad de géneros

**n\_2 =** Cantidad de estructuras

**n\_3 =** Tamaño de árbol tempo

**Primera parte:** O(n\_1) \* [O(n\_2) + O(2log(n\_3))] + O(n) \* O(n) + O(2log(n)) + O(n) + O(k)

**Segunda parte:** O(n) + O( ) + O(n) + O(n) + O(n) \* [ O(n\*k) \* O(n) \* O(n) + O(2log(n))]

**Complejidad final:** O(n) \* [O(n) + O(2log(n))] + O(n) \* O(n) + O(2log(n)) + O(n) + O(k) + O(n) + O( ) + O(n) + O(n) + O(n) \* [ O(n\*k) \* O(n) \* O(n) + O(2log(n))]

**Análisis de tiempo de ejecución y uso de memoria**

**Gráfico, Gráfico de barras

Descripción generada automáticamente**

**Gráfico, Gráfico en cascada

Descripción generada automáticamente**

En cuanto al análisis de complejidad temporal, es claro que los datos parecen seguir lo predicho teóricamente. Tenemos un aumento en la complejidad temporal proporcional al número del requerimiento tal y como se había previsto teóricamente. Sin embargo, nace una discordancia con la teoría y es que el requerimiento 4 dentro de sus incertidumbres pareciera superar en complejidad temporal al requerimiento 5 lo cual según la complejidad teórica no debería pasar. Nuestra hipótesis es que el requerimiento 5 que evidentemente está mal implementado debido a los resultados que arroja no está ejecutando correctamente todas sus líneas de código. A lo mejor, el try - except que busca evitar divisiones por cero simplifica el proceso y un indicio de esto es que en el resultado todos aparecen con la misma cantidad de Hashtags y el mismo VADER, como si solo se estuviera ejecutando una vez.

Ahora bien, en el análisis espacial no se desarrolló una teoría analítica compleja, pero por lo que se desarrolló en el código y la cantidad de estructuras que se utilizaron en cada requerimiento, era de esperarse un crecimiento que siguiera la misma tendencia de la complejidad temporal. Esto lo deducimos teóricamente debido a que la esencia de la complejidad temporal de los requerimientos desarrollados está en varios ciclos y búsquedas sobre varias estructuras, lo que nos lleva a pensar analíticamente que mayor complejidad temporal es la consecuencia de un mayor uso de estructuras (arboles, mapas y listas) lo que a su vez desemboca en una mayor complejidad espacial. Esto es claro si se considera el requerimiento 2, 3, 4 y 5 (en el 5 se ve de forma muy clara esta "inducción" analítica). Sin embargo, llama la atención que el requerimiento 1 el cual tiene la menor complejidad temporal de todos los requerimientos tenga mayor complejidad espacial que los requerimientos 2 y 3. No encontramos una justificación lógica a esta anomalía y concluimos que puede ser "ruido" de la maquina al tratarse de ser un requerimiento que se ejecutó justo después de la carga de datos (aunque se supone que el tracemalloc debería evitar este error).

Para encontrar la complejidad se realizó un análisis procediendo desde el ciclo más profundo hacia afuera realizando las operaciones algebráicas aprendidas en clase

**Diagrama requerimiento #4**

1. En primer lugar, se utiliza el map ‘EvByCaracteristics’ donde se encuentran los datos clasificados por cada característica de contenido que se encuentra en el archivo csv ‘context\_content\_features’. Para la solución del requerimiento se busca la llave que corresponda con la característica ‘tempo’. Allí, como valor se encuentra un árbol RBT, donde su llave pertenece a un tempo en especifico y su valor a un diccionario. En este, se encuentran dos tablas de hash con diferentes datos, como se observa en el diagrama.

Diagrama

Descripción generada automáticamente

1. En segundo lugar, se debe tener en cuenta el map ‘Genders’, donde se encuentran todos los géneros por defecto y sus rangos correspondientes del tempo. Con las dos estructuras ya mencionadas, se hace una búsqueda en el árbol RBT y se procede a utilizar un mapa de trabajo.

Diagrama

Descripción generada automáticamente

1. En el mapa de trabajo se agregan como llaves todos los ‘artist\_id’, y como valor el número 1 (por defecto). Luego al verificar y agregar todos los artistas unicos en el mapa, se procede a agregar toda la información en el mapa final.

Diagrama

Descripción generada automáticamente

1. En el mapa final, se agrega toda la información para cada género que se desea consultar, esto se hace mediante una tupla. Finalmente, se retorna el mapa junto al número total de reproducciones de todos los géneros.

Diagrama

Descripción generada automáticamente