

# **Análisis de resultados Reto 2**

François Morales – f.moraless - 202211168

Santiago Navarrete – s.navarretev - 202211202

Gabriela García – g.garcias2 – 202210869

## **Introducción**

El Reto 2 se basa en el manejo de un gran volumen de datos con el fin de realizar requerimientos útiles de búsqueda y análisis de ellos. Es importante resaltar que el tipo de estructura usado para este reto fue “Mapas No Ordenados”, asimismo, el parámetro para manejar las colisiones fue “Linear Probing” con un factor de carga de 0,5. Por otro lado, cuando así se requería, se utilizó únicamente la función de ordenamiento MERGE SORT, debido a su cualidad de estabilidad.

Por otro lado, con relación a los tiempos de ejecución, el reto se ejecutó en un solo computador cuyas características se describen a continuación:

- Procesador: AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz
- Memoria Ram: 16,0 GB (15,4 GB utilizable)
- Sistema Operativo: Sistema operativo de 64 bits, procesador x64 (Windows 11)

## **Requerimientos**

### **Carga de datos**

Para el análisis de los tiempos de ejecución del reto 2 se realizó la carga del 100% de los datos (22998 películas). Estos datos se agregaron a listas y mapas, según era necesario, ocupando una memoria de aproximadamente 160900.6 Kb. Con respecto a los mapas, se crearon con el parámetro de "Linear Probing" y con un factor de carga de 0.5. Esto debido a que basados en el laboratorio 7, la configuración más óptima era esta. (Ver anexo para entender la decisión)

### **Requerimiento 1 (Grupal)**

- Comparación de Complejidades

<b>RETO 1</b>	<b>RETO2</b>
$O(n+N\log N + k)$	$O(N+N\log N + k)$

Pasos:

1. Recorrer las películas filtradas para guardar en una lista solo las que son películas:  $O(N)$
2. Organizar la lista por título con merge sort:  $O(N\log N)$

El N del reto 1 es mucho mayor al del reto 2, ya que en el reto 1 recorremos todo el catálogo del número total de películas, en cambio en el reto 2, recorremos solo las filtradas por el parámetro que ingresa por parámetro.

- Tiempos de ejecución comparados

<b>RETO 1</b>	<b>RETO2</b>
64,64 ms	1,68 ms

## Requerimiento 2 (Grupal)

- Comparación de Complejidades

<b>RETO 1</b>	<b>RETO2</b>
$O(n+N\log N + k)$	$O(N+N\log N + k)$

Pasos:

1. Recorrido de la lista con la información de la fecha dada por parámetro y comparación con el tipo "Tv Show ":  $O(N)$
2. Ordenamiento de la lista con el algoritmo merge sort:  $O(N\log N)$

El N del reto 1 es mucho mayor al del reto 2, ya que en el reto 1 recorremos todo el catálogo del número total de películas, en cambio en el reto 2, recorremos solo las filtradas por el parámetro que ingresa por parámetro.

- Tiempos de ejecución comparados

<b>RETO 1</b>	<b>RETO2</b>
739,416 ms	2,22 ms

### Requerimiento 3 (Gabriela)

- Comparación de Complejidades

RETO 1	RETO2
$O(N+N\log N + k)$	$O(N+N\log N + k)$

Pasos:

1. Asignación de variables que me da una lista con las películas filtradas por el actor ingresado por parámetro para después recorrerlo y crear una lista de Movies y TVShows:  $O(N)$
2. Se ordena por merge la lista filtrada:  $O(N\log N)$

El N del reto 1 es mucho mayor al del reto 2, ya que en el reto 1 recorremos todo el catálogo del número total de películas, en cambio en el reto 2, recorremos solo las filtradas por el parámetro que ingresa por parámetro.

- Tiempos de ejecución comparados

RETO 1	RETO2
15,432 ms	0,07 ms

### Requerimiento 4 (François)

- Comparación de Complejidades

RETO 1	RETO2
$O(N + N\log N + k)$	$O(N + N\log N + k)$

Pasos:

1. Asignación de variables: StartTime, doble asignación de contenido y creación del contador para movies y Tv Shows:  $O(6)$
2. Recorrido de la lista con los contenidos y comparaciones con los tipos : “Movie” y “Tv Show” para incrementar el contador:  $O(N + 1)$
3. Ordenamiento por el algoritmo merge y asignación de las variables: end\_times y times:  $O(N\log N + 2)$

El N del reto 1 es mucho mayor al del reto 2, ya que en el reto 1 recorremos todo el catálogo del número total de películas, en cambio en el reto 2, recorremos solo las filtradas por el parámetro que ingresa por parámetro.

- Tiempos de ejecución comparados

RETO 1	RETO2
--------	-------

47,072 ms	15,57 ms
-----------	----------

### Requerimiento 5 (Santiago)

- Comparación de Complejidades

RETO 1	RETO2
$O(n + 2N\log(N) + k)$	$O(n + N\log(N) + k)$

Pasos:

1. Recorrido de la lista con los contenidos y comparaciones con el país indicado:  $O(N)$
2. Ordenamiento por el algoritmo merge y asignación de las variables:  $O(N\log N)$

El N del reto 1 es mucho mayor al del reto 2, ya que en el reto 1 recorremos todo el catálogo del número total de películas, en cambio en el reto 2, recorremos solo las filtradas por el parámetro que ingresa por parámetro.

- Tiempos de ejecución comparados

RETO 1	RETO2
13,588 ms	5,42 ms

### Requerimiento 6 (Grupal)

- Comparación de Complejidades

RETO 1	RETO2
$O(n + m \log(m) + m*o + m + p + q)$	$O(n \log(n) + n*m + o + p)$

Pasos:

1. Se ordena el contenido encontrado por director:  $O(n\log(n))$
2. Se itera la lista encontrada por director para contar cantidad de películas y series por plataforma:  $O(n*m)$   $m < n$
3. Se realizan una iteración en un diccionario para agregar la información por género de cada director:  $O(o)$
4. Se realizan una iteración en un diccionario para agregar la información por plataforma de cada director:  $O(p)$

La complejidad del reto 2 es menor ya que no se tiene que iterar en todo el catálogo de contenido para hallar el director solicitado, sino que se accede directamente al mapa. Por otro lado, en el reto 1 se realizan más ciclos para crear listas de

información, cosa que en el reto 2 fue optimizado para ser realizado en menos iteraciones.

- Tiempos de ejecución comparados

RETO 1	RETO2
7,319 ms	0,059 ms

### Requerimiento 7 (Grupal)

- Comparación de Complejidades

RETO 1	RETO2
$O(N^2 + N\log(N) + 1)$	$O(n + N\log N + N^2)$

Pasos:

1. Se accede a el mapa de géneros y se crea una lista que va a contar las veces que los géneros aparecen en todas las películas. Se recorre el mapa de los géneros sacando, el género, la cantidad y las películas que corresponden al género:  $O(N)$
2. Se ordena por la cantidad del género  $O(N\log N)$
3. Se crea una sub lista con los correspondientes que pertenecen al top solicitado. Se crea un diccionario que tiene como llave el género de la sub lista y como valor: tv show, películas y los streaming services. Para agregar elementos, se hace:
  - i. Se recorre la sub lista para obtener el genero
  - ii. Se accede a las películas que contienen dicho genero
  - iii. Se recorre las películas correspondientes al género para verificar de que tipo son y que a que streaming service pertenecen, mediante un contador:  $O(N^2)$

La complejidad del reto 1 es mayor debido a que se tiene que realizar una iteración en todo el catálogo de contenido, cosa que en el reto 2 no es necesario, ya que, en este caso, cada género tiene como valor asociado la lista de contenidos y la cantidad de estos. Por lo tanto, el proceso de encontrar el mayor se realiza más rápido.

- Tiempos de ejecución comparados

RETO 1	RETO2
76,993 ms	19,09 ms

### Requerimiento 8 (Bono Grupal)

- Comparación de Complejidades

RETO 1	RETO2
--------	-------

$O(N \log N + M \log M + 2n + N * \text{Top} * \text{lista} + \text{colaboraciones} * \text{Top})$	$O(N + M + M \log(M) + \text{top} + 2O \log(O) + 2P \log(P))$
--	---

Pasos:

1. Se itera el contenido por el género especificado para contar por actor:  $O(N)$
2. Se itera la información encontrada y se organiza por cantidad de menor a mayor:  $O(M + M \log(M))$
3. Se realiza una sub lista con la cantidad del top deseado de actores y se itera para sacar la información necesaria de cada uno:  $O(\text{top})$
4. Para las películas y series encontradas por actor se organizan por criterio del año de lanzamiento de estas:  $O(2 * O \log(O))$
5. Se realizan dos ciclos para escribir la información de las colaboraciones entre los actores del top y otros actores y directores, los cuales paran cuando la cantidad de caracteres supera 400:  $(2 * P \log(P))$

La complejidad es menor en este reto, ya que cuando se tiene el top de actores, no se hace necesario recorrer todo el catálogo de contenido para hallar la información de cada uno, sino que se accede al mapa de actores para solo iterar en el contenido respectivo de cada autor.

- Tiempos de ejecución comparados

RETO 1	RETO2
5382,068 ms	1928,53 ms

## Conclusiones

- El modelo de trabajo MVC permite la fácil implementación y modificación de la aplicación en general, adicionalmente, permite encontrar y solucionar errores de manera más fácil y eficiente
- Las complejidades teóricas coinciden con los tiempos de ejecución, por ejemplo, la complejidad teórica más grande se da en el requerimiento 8 (BONO) y así lo demuestran sus resultados con respecto a los tiempos de ejecución.
- La forma de manejo de colisiones 'Linear Probing' con un factor de carga de 0,5 es la más óptima, ya que el tiempo de ejecución y la memoria utilizada para guardar la información es el más pequeño comparado con los otros parámetros.
- La función de ordenamiento más eficiente en este Reto fue "Merge Sort", ya que, al ser una función estable, no requiere que se lleve a cabo un análisis de como vienen los datos de entrada. Adicionalmente, obtuvo los mejores tiempos de ejecución para cada requerimiento.

- Teniendo en cuenta las comparaciones con el reto 1, en el presente reto, al usar mapas no ordenados para acceder a la información, permite que los tiempos de ejecución sean menores. Por lo tanto, se concluye que la manera más eficiente de acceder a la información es con el uso de ADT mapas.

## Anexos

Tabla de tiempos (medidos en ms) utilizadas para definir el factor de carga y parámetro anticollisiones

### Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	375523.5	6894.7
0.5	152843.4	5075.9
0.7	135502.1	5144.1
0.9	124618.3	6278.7

*Tabla 1. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina Utilizada.*

### Carga de Catálogo CHAINING

Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	151216.7	7965.6
4.00	134550.2	5982.3
6.00	131946.6	5986.3
8.00	130315	5989.7

*Tabla 2. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina Utilizada.*