

Análisis del Reto

Grupo 2 Sección 4

Requerimiento 3 - Daniel Rodriguez - ds.rodriquezf1@uniandes.edu.co - 202014760

Requerimiento 4 - Santiago Forero - s.forerog2@uniandes.edu.co - 202111446

N = Artists.csv

M = Artworks.csv

Requerimiento 1

- Análisis complejidad temporal:

La función SortArtists tiene una complejidad temporal de $O(n \log(n)) + O(n)$ debido a que se está usando una función de ordenamiento Merge Sort sobre un Array List en el modelo, la cual tiene tanto en su mejor como en su peor caso una complejidad temporal $O(n \log(n))$, y a esto se le suma el segundo recorrido que se realiza en el controlador para obtener el rango de fechas indicado por el usuario, el cual tiene como complejidad simplemente $O(n)$.

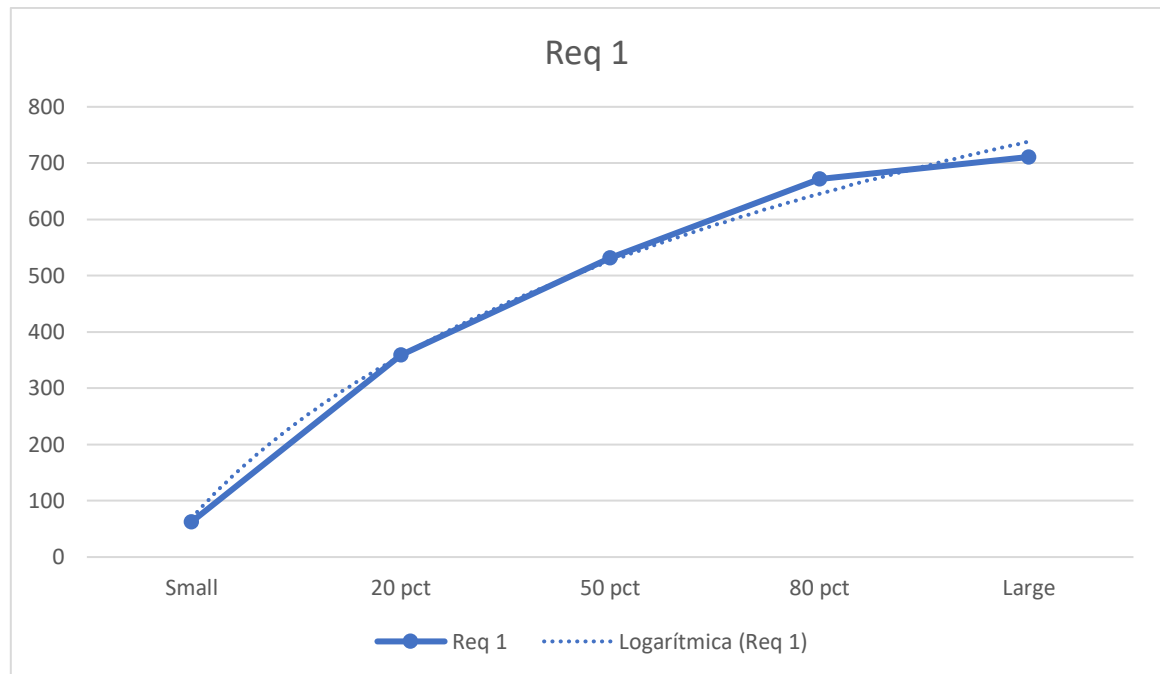
Así que la complejidad temporal del primer requerimiento es:

$$O(n \log(n)) + O(n)$$

- Pruebas de tiempo – Tablas

	Small	20 pct	50 pct	80 pct	Large
Req 1	62.5	359.375	531.25	671.875	710.93

- Gráfica



Se puede ver que tiende a ser logarítmica, tal como se esperaría teniendo en cuenta la teoría

Requerimiento 2

- Análisis complejidad temporal:

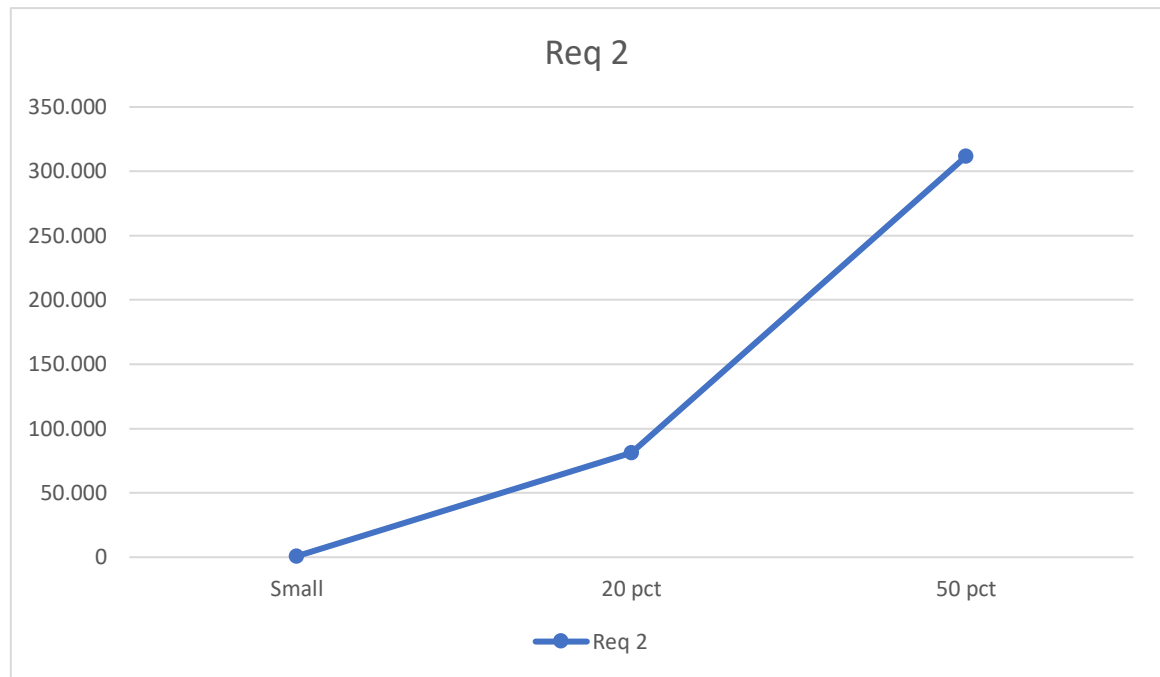
La función SortArtworks tiene una complejidad temporal de $O(m \log(m)) + O(m*n)$ debido a que se está usando una función de ordenamiento Merge Sort sobre un Array List en el modelo, la cual tanto en su mejor como en su peor caso tiene como complejidad temporal $O(m \log(m))$, y a esto se le suma el segundo recorrido que se realiza en el controlador para obtener el rango de fechas indicado por el usuario, el cual tiene como complejidad de $O(m)$, lo cual se multiplica por $O(n)$ debido a que se hace necesario buscar los nombres de los artistas de cada obra, por lo que para cada recorrido de m , también se hace uno de n . Así que la complejidad temporal del segundo requerimiento es:

$$O(m \log(m)) + O(m*n)$$

- Pruebas de tiempo – Tablas

	Small	20 pct	50 pct	80 pct	Large
Req 2	765.625	81093.75	311593.75	NA	NA

- Gráficas



- Por cuestiones del procesador del computador, las pruebas no dan resultados muy reveladores.

Requerimiento 3 – Daniel Rodriguez

- Análisis complejidad temporal:
- Pruebas de tiempo – Tablas

	Small	20 pct	50 pct	80 pct	Large
Req 3					

- Gráficas

Requerimiento 4 – Santiago Forero

- Análisis complejidad temporal:
 P = Número de nacionalidades
 Q = Número de artistas de la nacionalidad con el mayor número de obras
- La función ArtworksNacionalidad tiene una complejidad temporal de $2 \cdot O(m \cdot n) + O(p) + O(p \log(p)) + O(q)$, el primer $O(m \cdot n)$ surge debido a que se está buscando por cada obra la nacionalidad de su artista asociado para lograr hacer el conteo, posteriormente surge el primer $O(p)$ al pasar la información de diccionario a Array List para lograr realizar un ordenamiento de Merge Sort, de donde se obtiene el $O(p \log(p))$. Posteriormente para

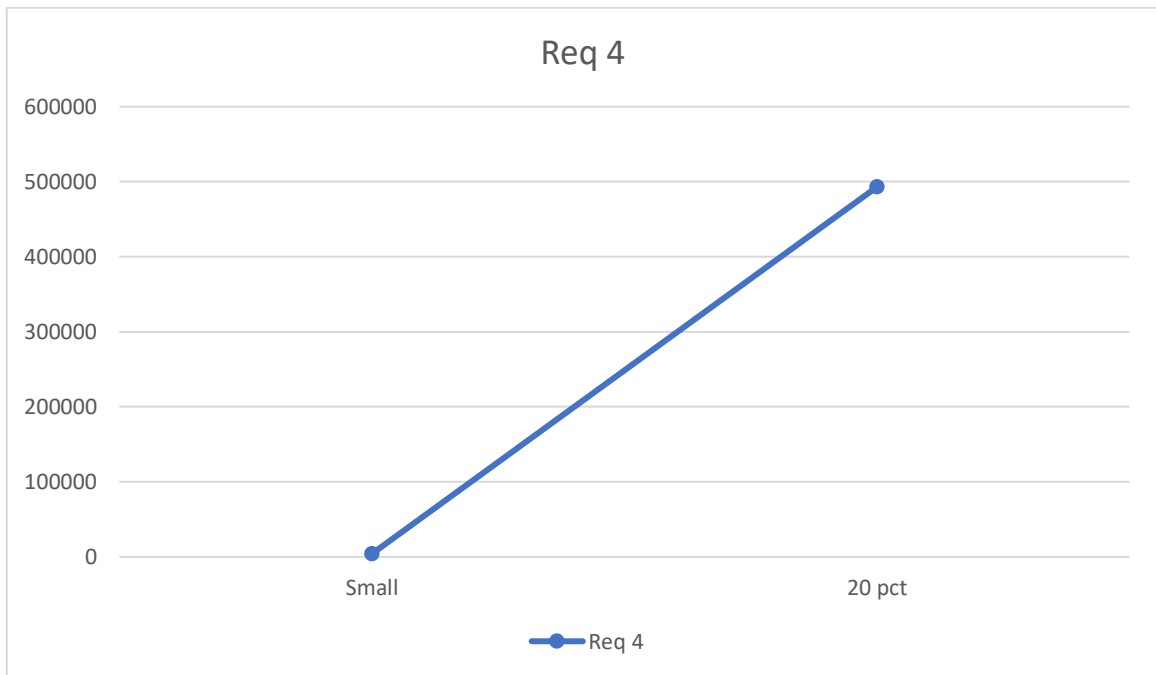
obtener la información de las obras de la nacionalidad con el mayor número de estas, se buscó por cada artista de esta nacionalidad una obra asociada, obteniendo el segundo $O(m*n)$ y finalmente al ordenar la información para imprimir en pantalla surgió $O(q)$. Así que la complejidad temporal del cuarto requerimiento es:

$$2*O(m*n) + O(p) + O(p \log(p)) + O(q)$$

- Pruebas de tiempo – Tablas

	Small	20 pct	50 pct	80 pct	Large
Req 4	4135.42	493015.625	NA	NA	NA

- Gráficas



Por cuestiones del procesador del computador, las pruebas no dan resultados muy reveladores.

Requerimiento 5

- Análisis complejidad temporal:
 R = Primeras 5 obras más costosas
 S = Primeras 5 obras más antiguas
- La función CostoTransDept tiene una complejidad temporal de $O(m*r*s) + O(r \log(r)) + O(s \log(s)) + O(m*n)$. Esto debido a que realiza un recorrido por todas las obras mientras crea listas más pequeñas con las 5 obras más antiguas y costosas respectivamente. No obstante, ya que cada una de estas listas tiene únicamente 5 elementos siempre, se podría

escribir también la complejidad temporal como $O(m) + O(m*n)$. En todo caso, durante este recorrido cada lista r y s se ordena por medio de Merge Sort una sola vez al inicio con complejidades de $O(r \log(r)) + O(s \log(s))$ o $2*O(5 \log(5))$. Finalmente para imprimir en pantalla se buscan los nombres de los artistas con complejidad temporal $O(m*n)$. Así que la complejidad temporal del quinto requerimiento es:

$$O(m*r*s) + O(r \log(r)) + O(s \log(s)) + O(m*n)$$

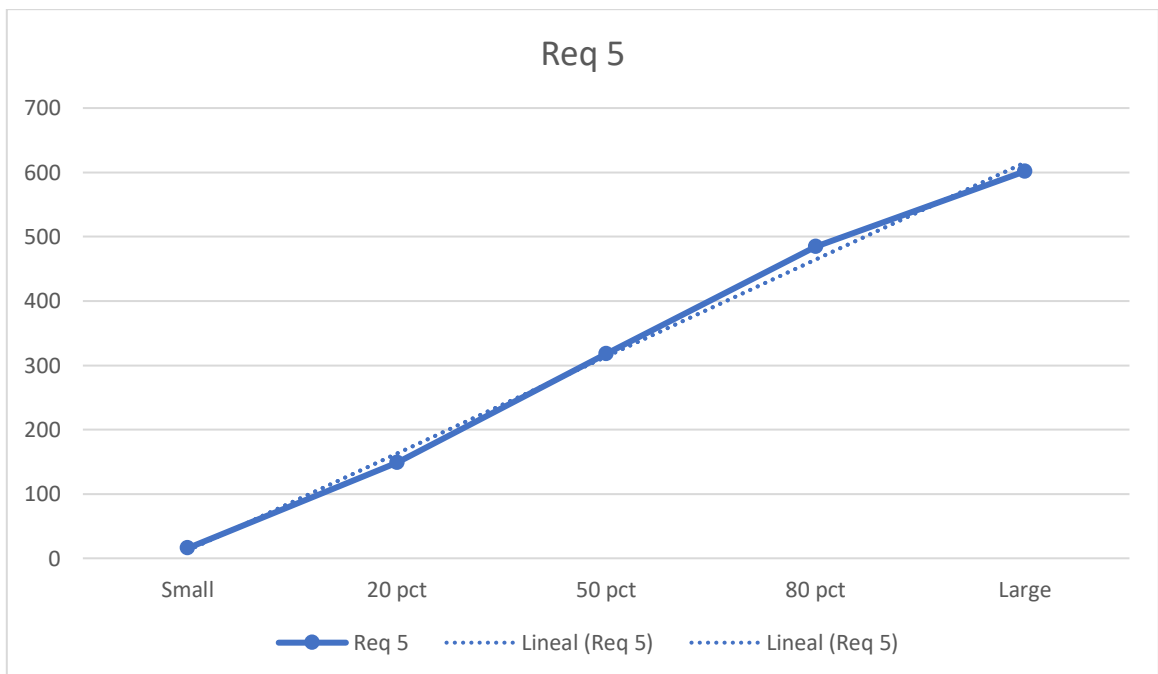
o

$$O(m) + O(m*n)$$

- Pruebas de tiempo – Tablas

	Small	20 pct	50 pct	80 pct	Large
Req 5	15.625	148.437	317.7	484.375	601.56

- Gráficas



Se puede ver que tiende a ser lineal, tal como en la teoría

Requerimiento 6

- Análisis complejidad temporal:
- La función NuevaExpo tiene una complejidad temporal de $O(m) + O(m*n)$. Esto debido a que recorre las obras filtrando aquellas que sean planas para ir las sumando hasta llegar al

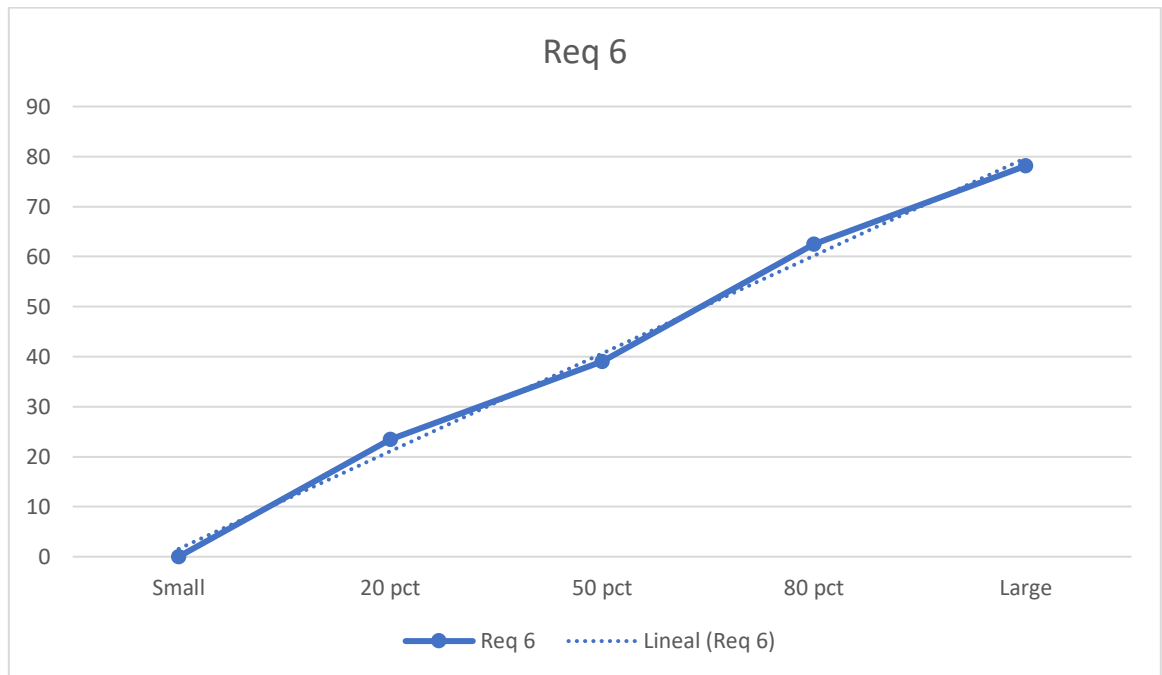
límite. Y posteriormente recorre los artistas por cada obra para encontrar los nombres es estos. Así que la complejidad temporal del sexto requerimiento es:

$$O(m) + O(m*n)$$

- Pruebas de tiempo – Tablas

	Small	20 pct	50 pct	80 pct	Large
Req 6	0	23.43	39.06	62.5	78.125

- Gráficas



Se puede ver que tiende a ser lineal, tal como en la teoría