

Documento de análisis reto 1

Alejandro Borda – 202020727

Juan Sebastián Lache - 201821331

Máquina de prueba:

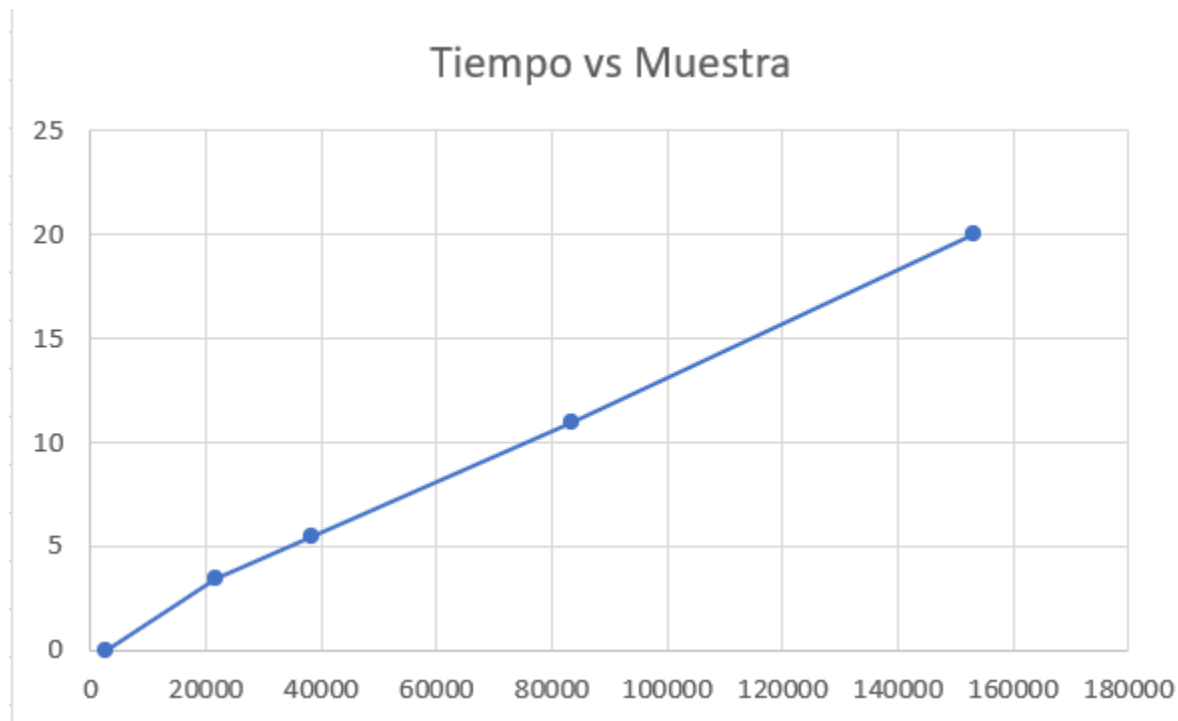
Procesadores => AMD Ryzen 5 3550H 2.10 GHz

Memoria RAM => 8 GB

Sistema operativo => Windows 10, 64 Bits

Carga de datos:

Muestra	Tiempo (segundos)
small	0
10%	3.5
20%	5.5
50%	11
large	20



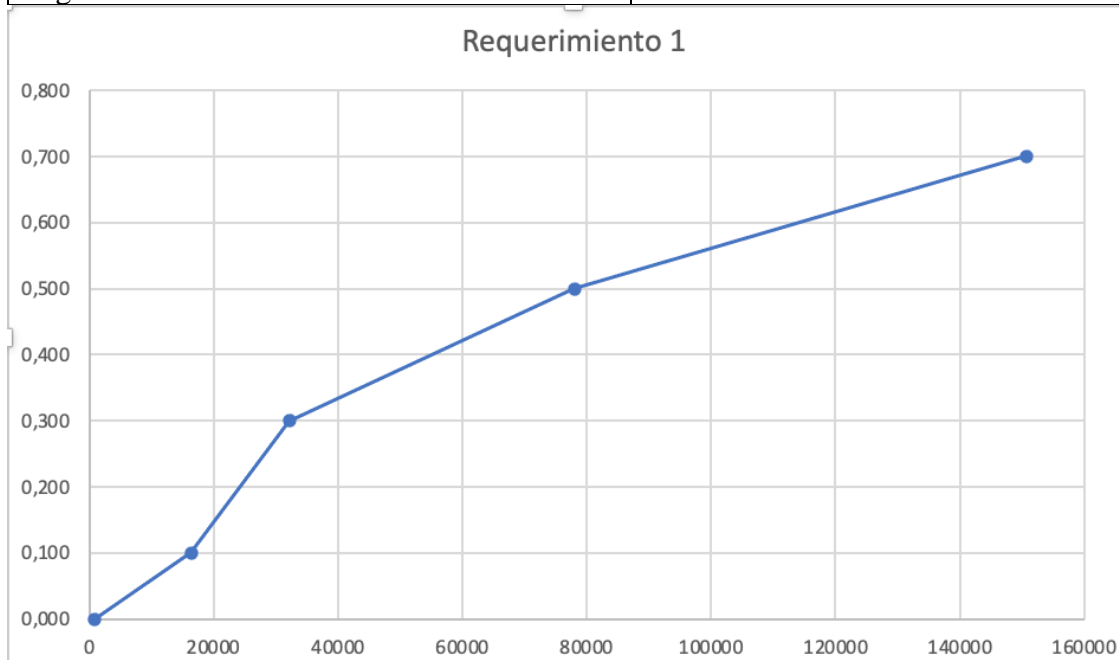
En base al código se podría esperar que la carga de datos tiene una complejidad alrededor de $O(4n)$ con n siendo el numero de obras, pues debe ser alrededor de 4 comparaciones por cada obra, por cada artista se hacen alrededor de dos comparaciones pero el numero de artistas es mucho menor al de obras en la mayoría de las muestras, con lo cual en notación big O, se podría aproximar a $O(n)$. Esta complejidad también es apoyada por la gráfica.

Complejidad espacial:

Previo a la ejecución del archivo large, la memoria figuraba en 6.52/7.81 GB, durante la carga del archivo con mas datos el máximo incremento de memoria fue 6.64/7.81

Requerimiento 1

Muestra	Tiempo (segundos)
small	0
10%	0.1
20%	0.3
50%	0.5
large	0.7



En base al código es razonable concluir que la complejidad es de tipo $O(m \log m)$, con m el numero total de artistas pues en el peor caso se da que todos los artistas se encuentran en el rango de fechas y por lo tanto se haría el merge sort sobre todos los artistas de la lista, sin embargo esta lista es más pequeña que la de obras y los tiempos de carga son tan cortos, al igual que su diferencia que no se puede concluir nada en base a la gráfica, además de constar de una alta incertidumbre en la medición de los datos relativo a la magnitud del tiempo medido.

Requerimiento 2

Muestra	Tiempo (segundos)
small	0
10%	0.5
20%	0.8
50%	1.1
large	1.6



En base al código es razonable esperar que la complejidad de este requerimiento sea de $O(n \log n)$ debido al uso de merge sort, pero sin embargo este sorting se uso sobre una list filtrada por fecha con lo cual en su mayoría de casos es mejor que $O(n \log n)$. Sin embargo en el peor caso en donde se tomen todas las fechas su complejidad seria $O(n \log n)$. Esta complejidad es apoyada por la grafica pues para realizar las pruebas se toma siempre el peor caso (todas las fechas) con lo cual refleja su complejidad en big O.

Requerimiento 3 (Juan Sebastian Lache):

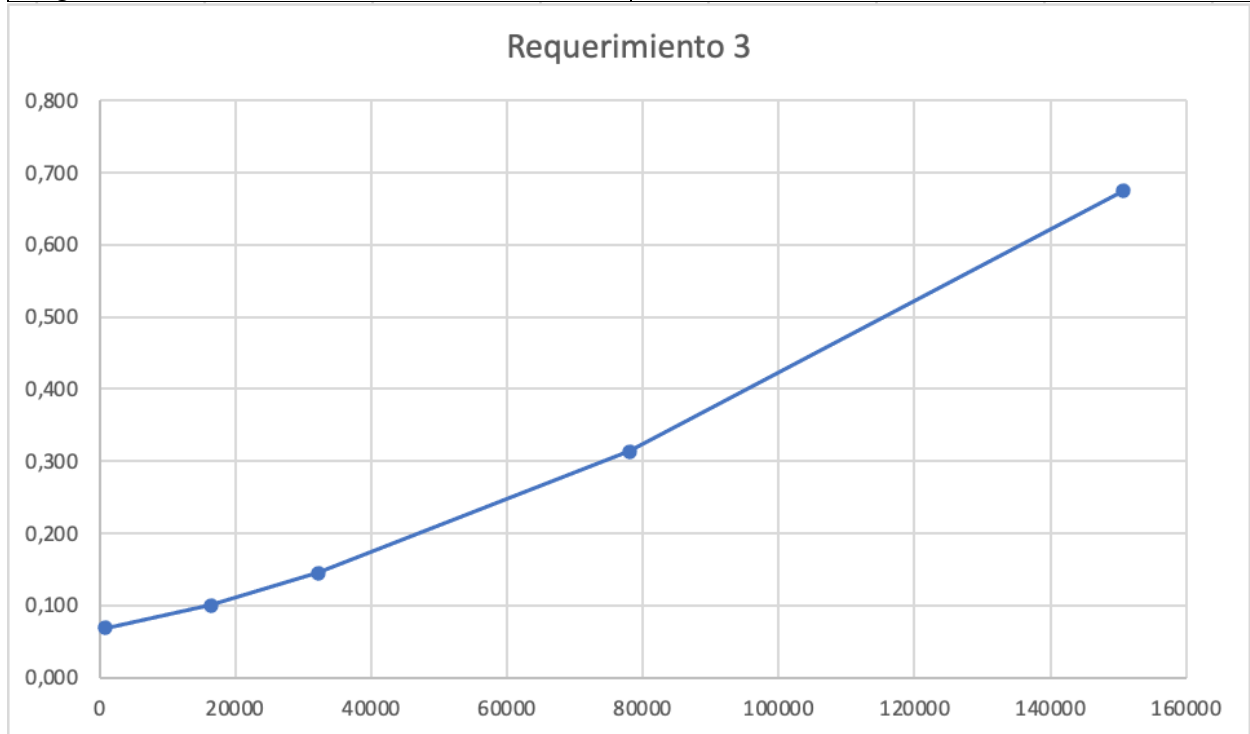
*Para esta prueba se usó una máquina diferente:

Procesador: 2,6 GHz Intel Core i5

Memoria RAM: 8 GB

Sistema operativo: MacOS Mojave

Muestra	Tiempo (segundos)
small	0.069
10%	0.1
20%	0.145
50%	0.313
large	0.674



Se puede decir que la complejidad en este requerimiento es de $O(n)$ debido al merge sort se realiza sobre una lista más pequeña que la de los artistas. Debido a esto se puede considerar el primer ciclo en el que se recorre toda la lista de artistas como el peor caso y por esto la complejidad sería de $O(n)$.

Requerimiento 4 (Alejandro Borda):

Muestra	Tiempo (segundos)
small	0
10%	0.4
20%	0.7
50%	1
large	1.5



En base al código no hay una clara complejidad pues el merge sort se usa sobre una lista de nacionalidades que es siempre considerablemente menor a la cantidad de obras y artistas, también se tiene un ciclo sobre la lista de obras de la primera nacionalidad que representa una fracción del total de las obras con lo que es razonable aproximar a $O(n)$, a pesar de que la gráfica parezca tratarse de otra complejidad, estos datos no son muy precisos en su tendencia debido a los cortos tiempos de ejecución

Requerimiento 5:

Muestra	Tiempo (segundos)
small	0
10%	1
20%	1.8
50%	4.3
large	9.5

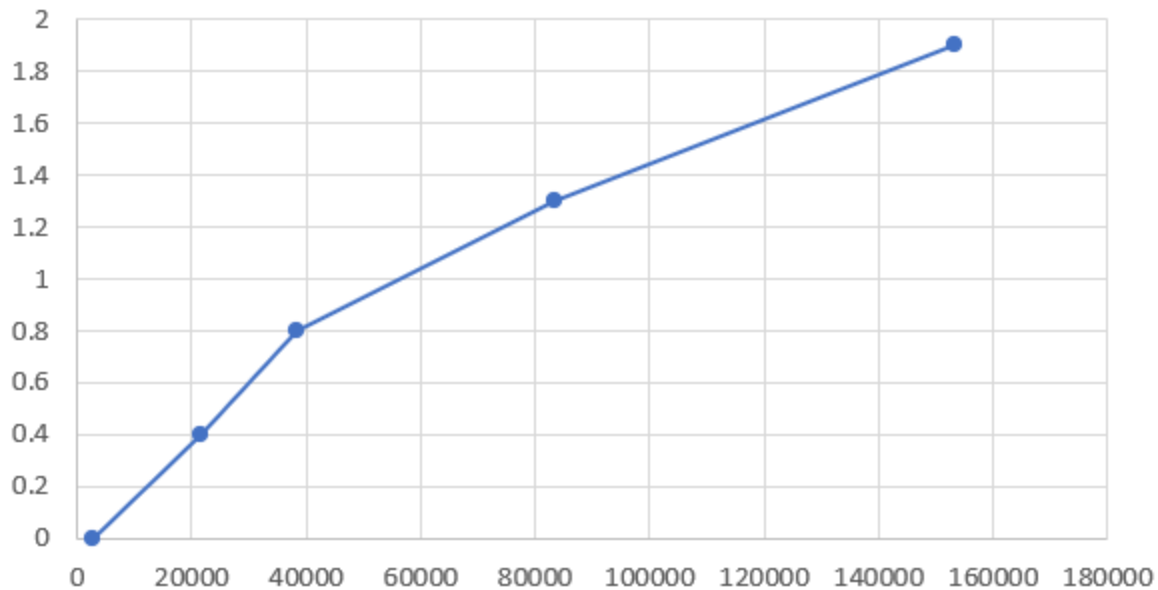


En base al código es razonable aproximar a $O(n \log n)$ pues se realiza un merge sort sobre una fracción de las obras (un departamento), a pesar de que esta fracción nunca sea igual al total de obras en notación big O se toma una fracción de n como n , nuevamente la grafica parece indicar que esta es la tendencia y efectivamente $O(n \log n)$ parece ser una cota superior.

Requerimiento 6:

Muestra	Tiempo (segundos)
small	0
10%	0.4
20%	0.8
50%	1.3
large	1.9

Tiempo vs muestra



La complejidad de este requerimiento es de $O(m \log m)$ con m siendo el numero de artistas, esto es claro debido a que se debe realizar un merge sort sobre todos los artistas, sin embargo el numero de artistas siempre es menor al de obras en especial para los archivos mas grandes. Nuevamente la gráfica parece apoyar la complejidad esperada.

