

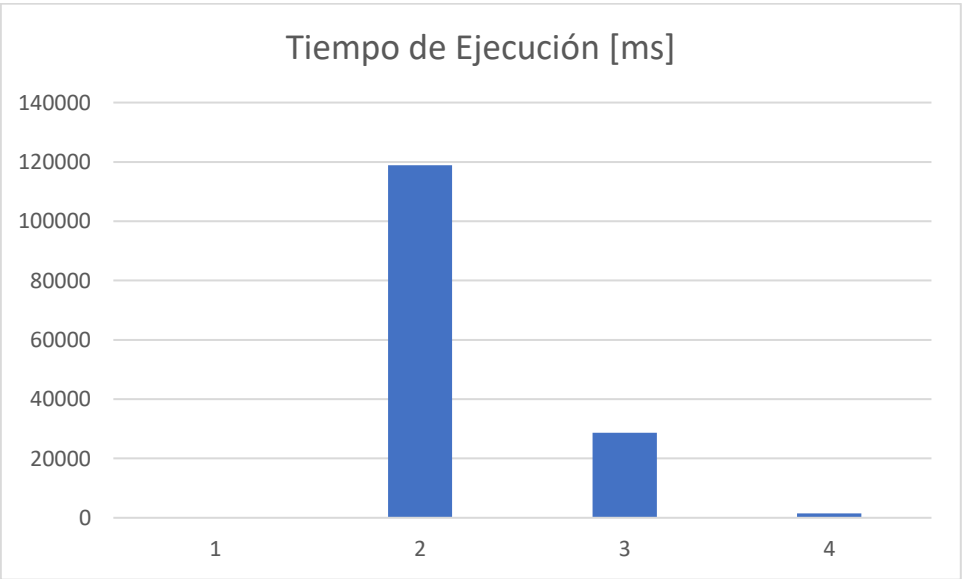
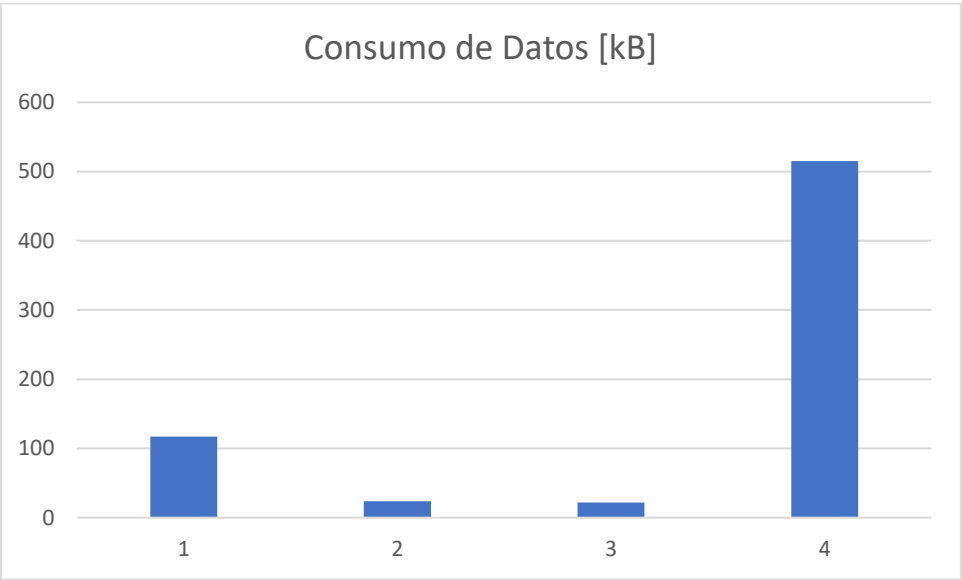
# Observaciones del Reto 2

Nicolás Guerrero Cod 201731839 (Maquina 1)  
Santiago Duque Cod 202021020 (Maquina 2)

## Maquina 1

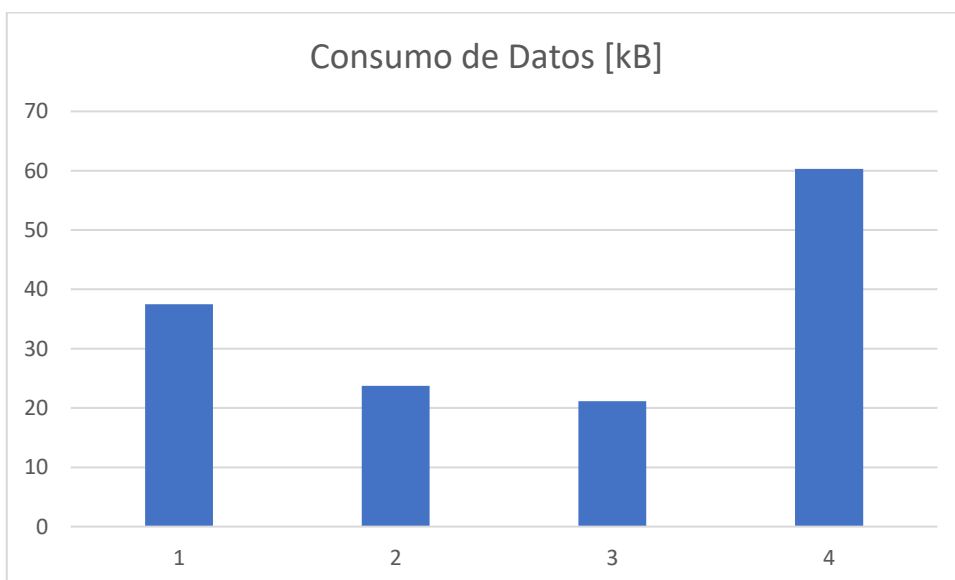
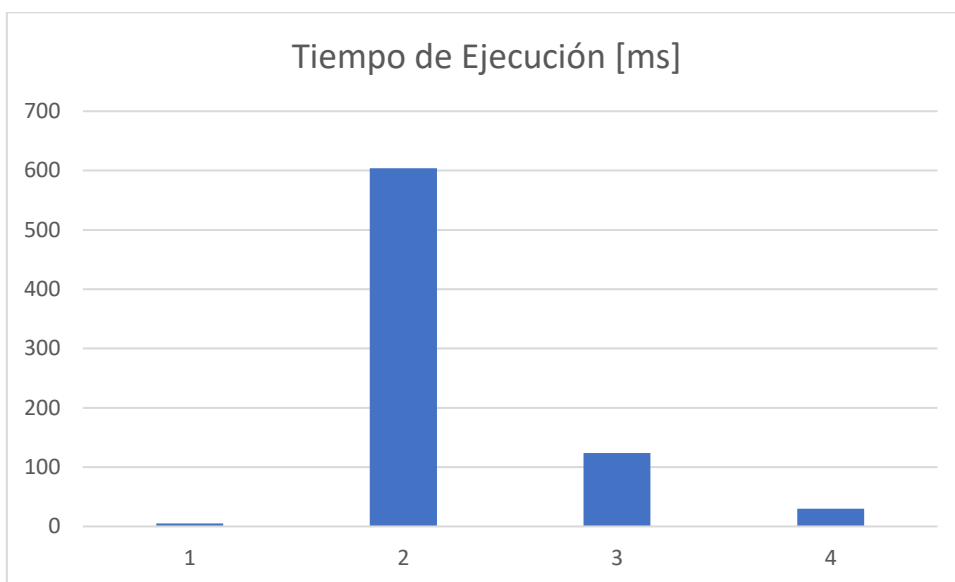
Archivo con el 10% de los datos (videos-10pct.csv)

Requerimiento	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
1	116.988	75.908
2	23.734	118924.356
3	21.672	28622.645
4	515.047	1487.345



Archivo de videos pequeño (videos-small.csv)

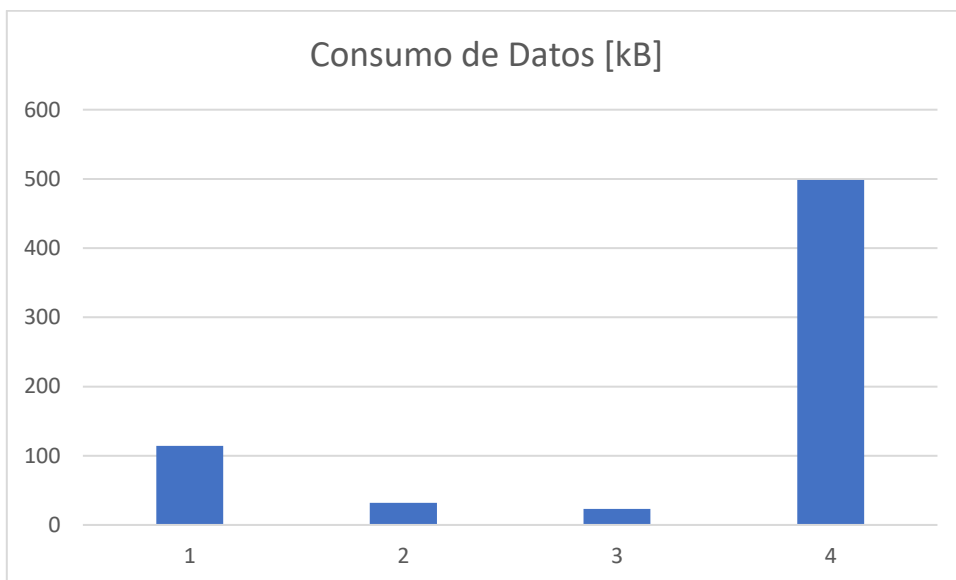
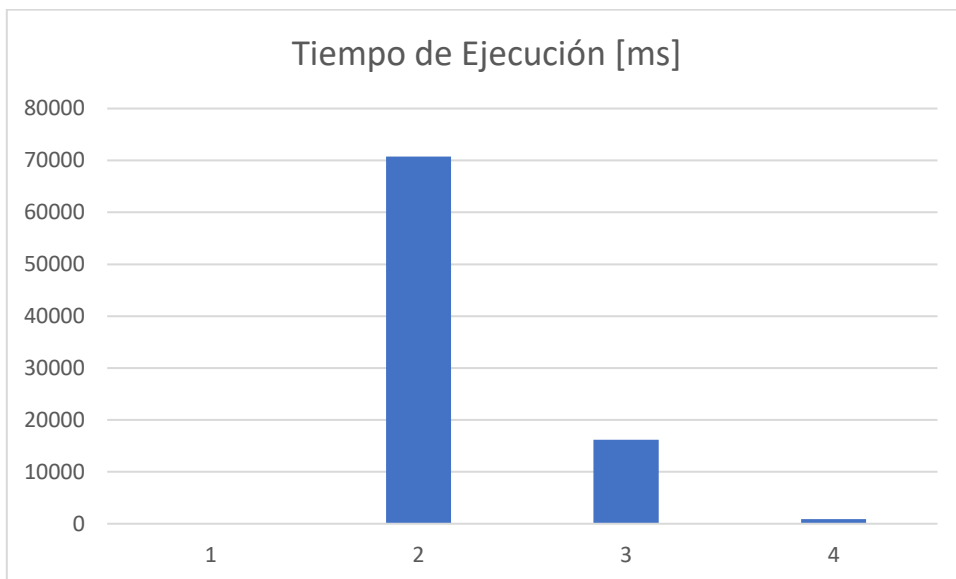
Requerimiento	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
1	37.484	5.045
2	23.734	603.946
3	21.125	123.753
4	60.305	30.109



## Maquina 2

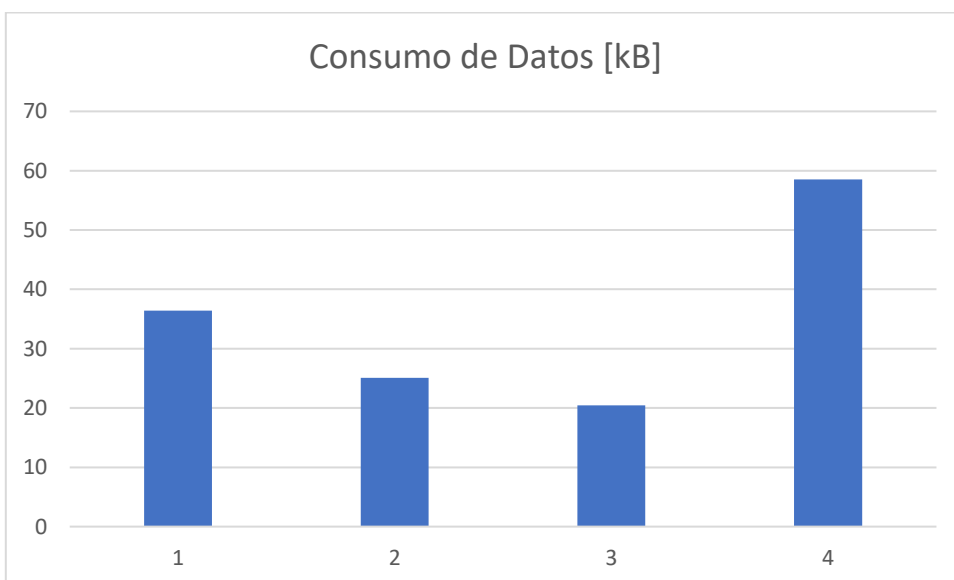
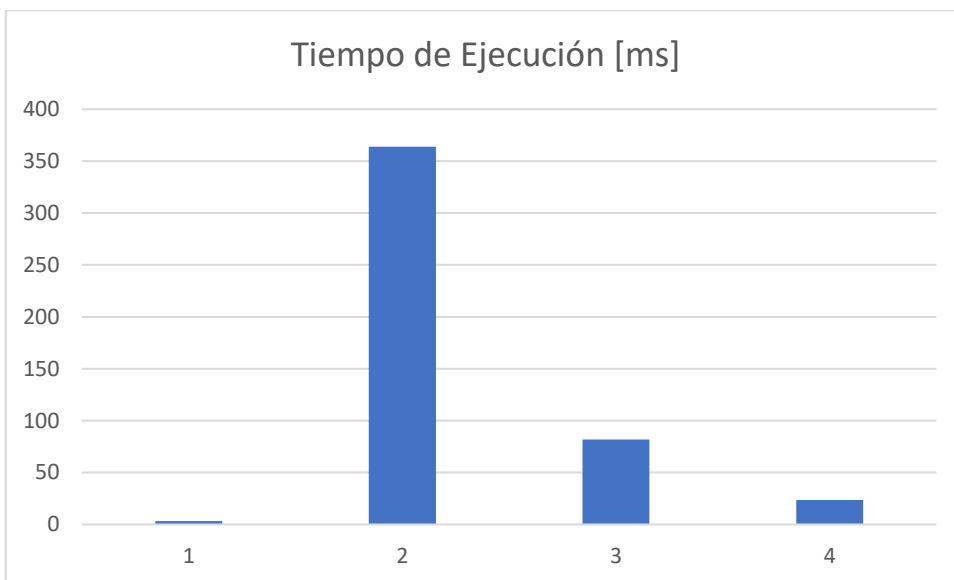
Archivo con el 10% de los datos (videos-10pct.csv)

Requerimiento	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
1	114.421	61.664
2	31.738	70723.846
3	23.148	16191.624
4	498.637	914.456



Archivo de videos pequeño (videos-small.csv)

Requerimiento	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
1	36.414	3.245
2	25.059	363.884
3	20.453	81.883
4	58.512	23.672



Análisis de datos

Al implementar los TAD maps logramos cargar muchos más datos y hacer diferentes operaciones sobre la base de datos de forma mucho más rápida. Ahora bien, en los requerimientos 2 y 3 la implementación de maps fue limitada a ciertas funciones de consulta por lo que estos siguieron implementando listas, cosa que explica sus altos tiempos de ejecución al compararse con los requerimientos 1 y 4. Los maps por lo tanto ayudaron a que consultar información sobre esta base de datos fuese una tarea mucho más veloz, pero el precio para esta rapidez de consulta es un mayor uso de espacio en la memoria del dispositivo.

En el requerimiento 1 se utilizó una función que consultaba cuales videos estaban relacionados a una categoría especificada por el usuario. Esto en contraste con el reto 1 hizo que los tiempos de búsqueda fuesen mucho mas bajos porque mientras que en el reto 1 se creaba una nueva lista después de comparar todos los elementos de una lista mas grande, en el reto 2 el map ahorro la consulta sobre la lista original. Esto implica que nunca se debieron comparar todos los datos del archivo original, sino que solo fue necesario comparar los videos de la lista que pertenecían a una categoría.

Algo similar ocurrió en el requerimiento 2 pero las diferentes operaciones aplicadas sobre la lista obtenida del map de países hizo que su rendimiento fuese mucho más bajo. Lo mismo ocurrió para el requerimiento 3 pero en vez de una consulta sobre un map de países se hizo una consulta sobre el de categorías.

Por último, en el requerimiento 4 solo fue necesario hacer una consulta sobre el map de países y de ahí solo fue necesario ordenar esta lista y sacar los primeros videos, cosa que mantuvo los tiempos de carga bajos comparados con los otros requerimientos.

En conclusión, los requerimientos 3 y 4 al tener que hacer varias operaciones sobre listas tuvieron tiempos de carga elevados. Si bien estos requerimientos implementaron maps, el uso de las listas aumento los tiempos de carga. Este no fue el caso para los requerimientos 1 y 4 en donde el uso de listas fue mínimo, cosa que permitió realizar las operaciones de consulta rápidamente. Esto también explica el uso de datos en los requerimientos 1 y 4 siendo mayor que en los 2 y 3 pues al implementarse su solución usando casi únicamente maps, los tiempos de búsqueda son menores pero el espacio en memoria requerido es mayor.