

OBSERVACIONES RETO 3

Juan José Osorio (202017020)

Thais Tamaio Ramírez (202022213)

	Máquina 1	Máquina 2
Procesadores	3,1 GHz Intel Core i5 de dos núcleos	Intel(R) Core(TM) i7-3667U CPU @ 2.00GHz 2.50 GHz
Memoria RAM (GB)	8 GB 2133 MHz LPDDR3	4.00 GB (3.90 GB usable)
Sistema Operativo	macOS Big Sur versión: 11.2.1	Windows 10 Home 64-bits

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

Para realizar el análisis de los resultados en términos de memoria y tiempos de ejecución, decidimos medir el tiempo de ejecución y la memoria consumida para la carga de datos y cada uno de los requerimientos del reto 3, en dos máquinas diferentes. Adicionalmente, se usaron 3 archivos para realizar estas pruebas, donde cada uno tiene una cantidad diferente de datos. Los archivos utilizados para estas pruebas corresponden a: small.csv, 5pct.csv y 10pct.csv. En las siguientes tablas se muestran los resultados al ejecutar la carga de datos y los cinco requerimientos del reto 3, con las diferentes cantidades de datos utilizadas:

Máquina 1 / Small.csv

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5
Tiempo de ejecución [ms]	30721,03	1525,86	440,97	574,48	863,43	5985,47
Memoria consumida [kB]	1079410,45	12477,3	5328,71	7975,36	75639,69	456414,47

Máquina 1 / 5pct.csv

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5
Tiempo de ejecución [ms]	248475,14	15954,79	3730,81	4198,72	1729,61	20461,26
Memoria consumida [kB]	3400000,01	49393,75	10293,59	12965,87	78656,92	505097,93

Máquina 1 / 10pct.csv

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5
Tiempo de ejecución [ms]	519615,12	41289,68	9824,98	7929,65	12135,13	44390,15
Memoria consumida [kB]	5625526,57	90310,88	15249,79	17660,54	81366,7	674319,35

Máquina 2 / Small.csv

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5
Tiempo de ejecución [ms]	115750,33	1029,69	691,58	825,97	4690,55	6644,2
Memoria consumida [kB]	2358364,47	12078,59	5176,31	7726,39	73311,69	440926,99

Máquina 2 / 5pct.csv

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5
Tiempo de ejecución [ms]	811004,1	10249,23	6809,31	7083,25	5439,01	91343,71
Memoria consumida [kB]	3325933,97	48108,78	10128,73	12705,83	76286,15	489642,1

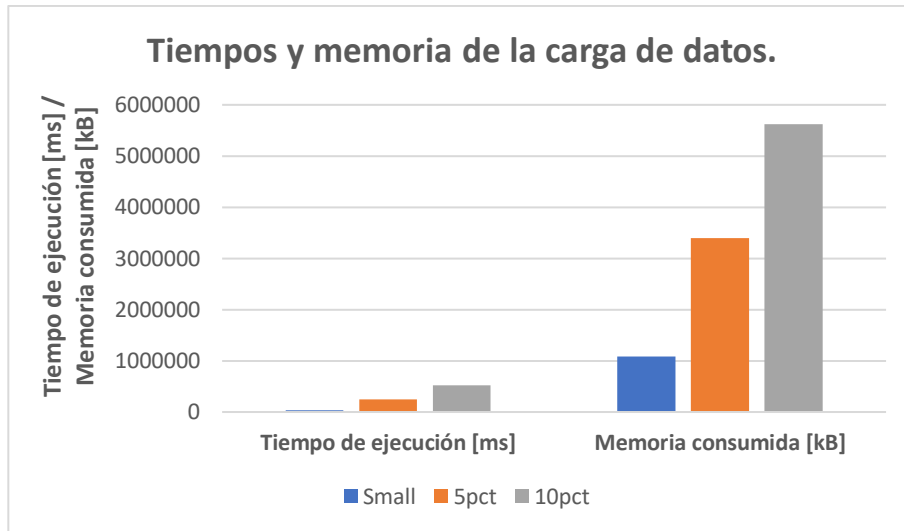
Máquina 2 / 10pct.csv

Debido a que al realizar las pruebas de rendimiento la carga del catálogo superó los 20 minutos con el archivo 10pct.csv, se omitió esta prueba para la máquina 2.

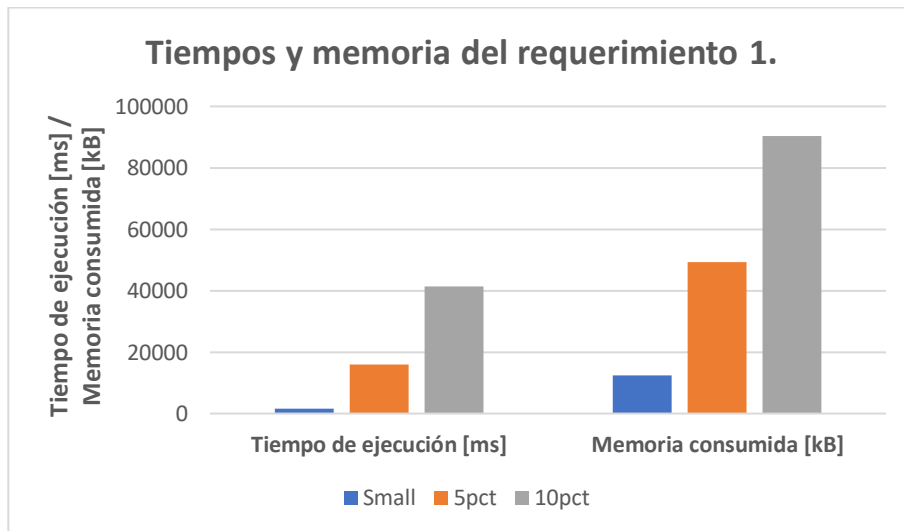
Por medio de las tablas que se mostraron anteriormente, se obtuvieron las siguientes gráficas para la máquina 1 y la máquina 2:

1. Gráficas máquina 1:

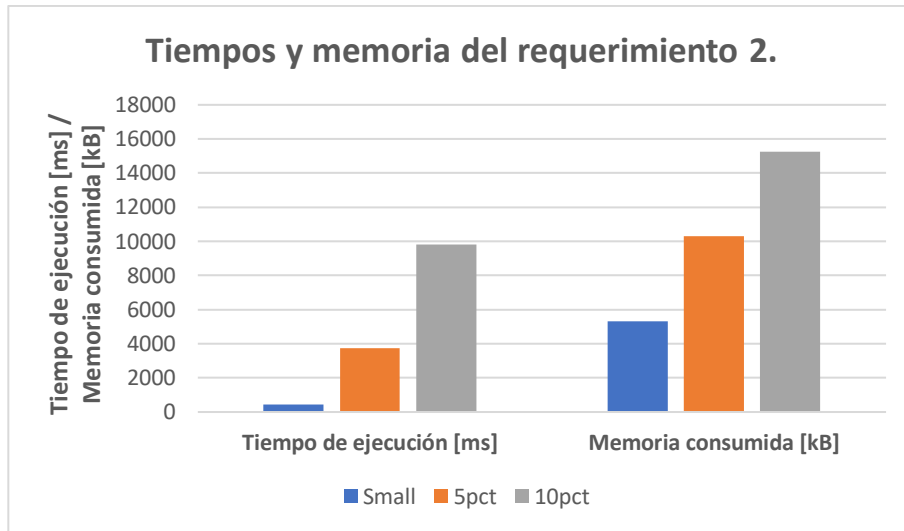
- Carga de datos:



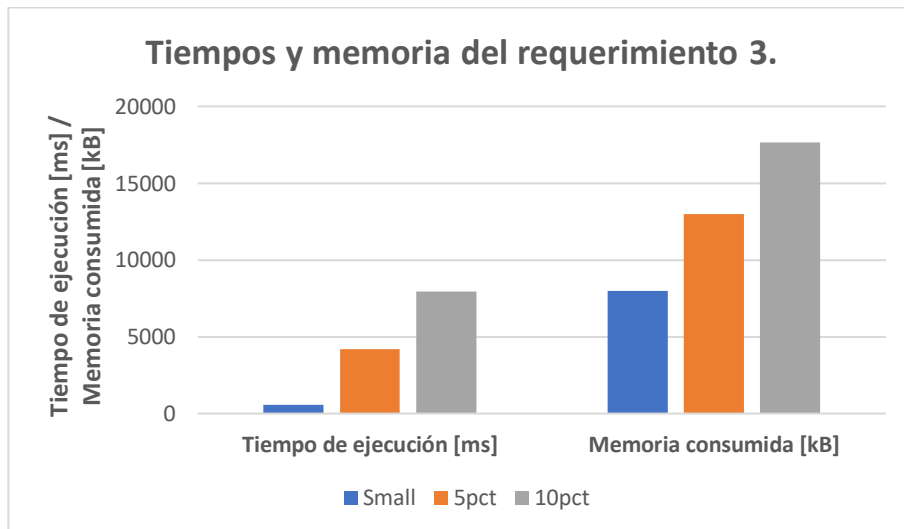
- Requerimiento 1:



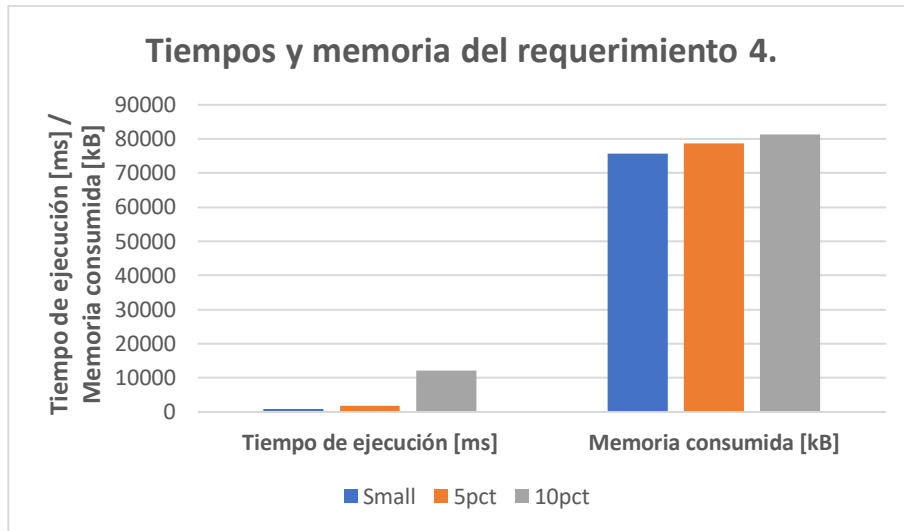
- Requerimiento 2:



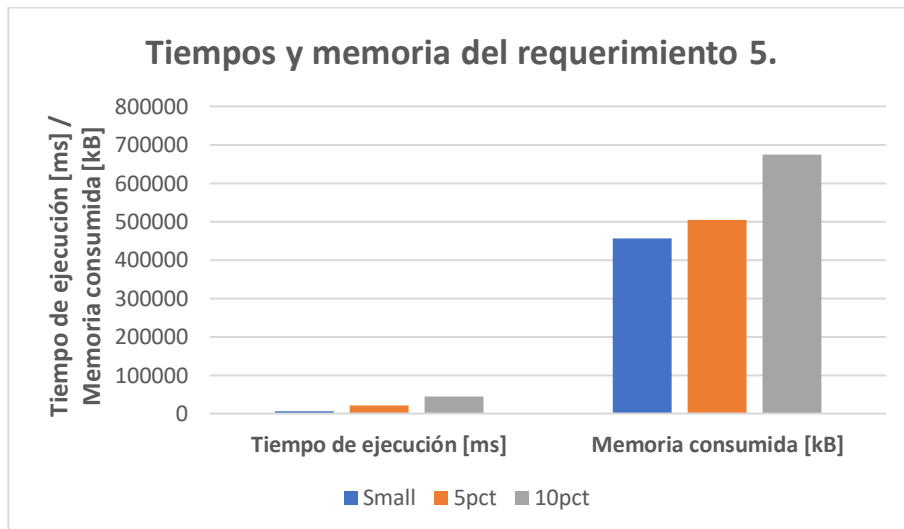
- Requerimiento 3:



- Requerimiento 4:

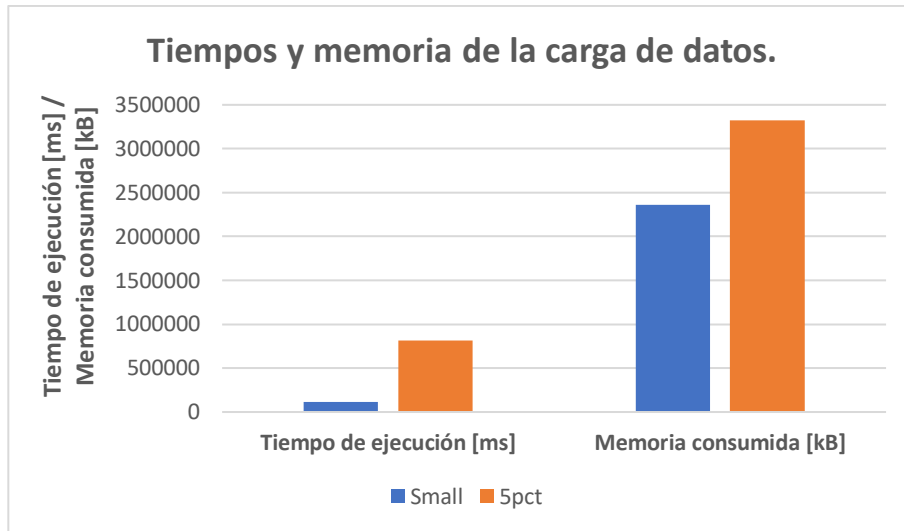


- Requerimiento 5:

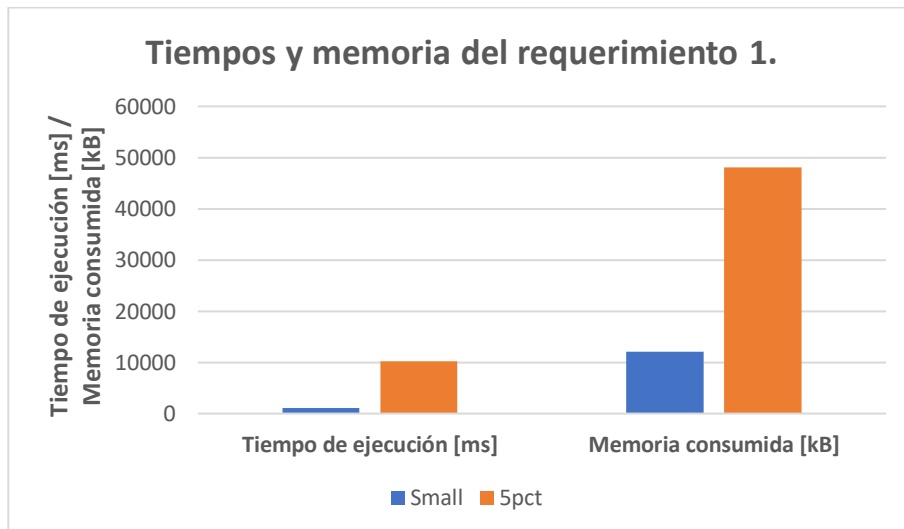


2. Gráficas máquina 2:

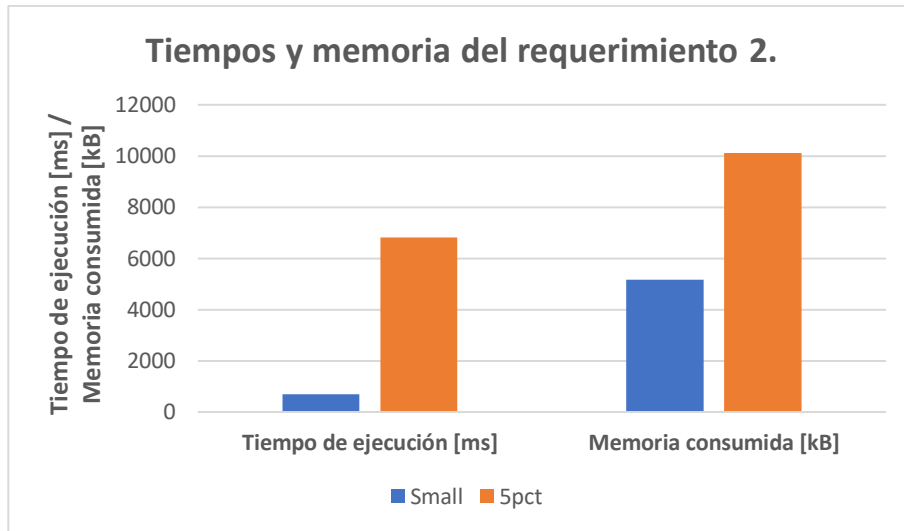
- Carga de datos:



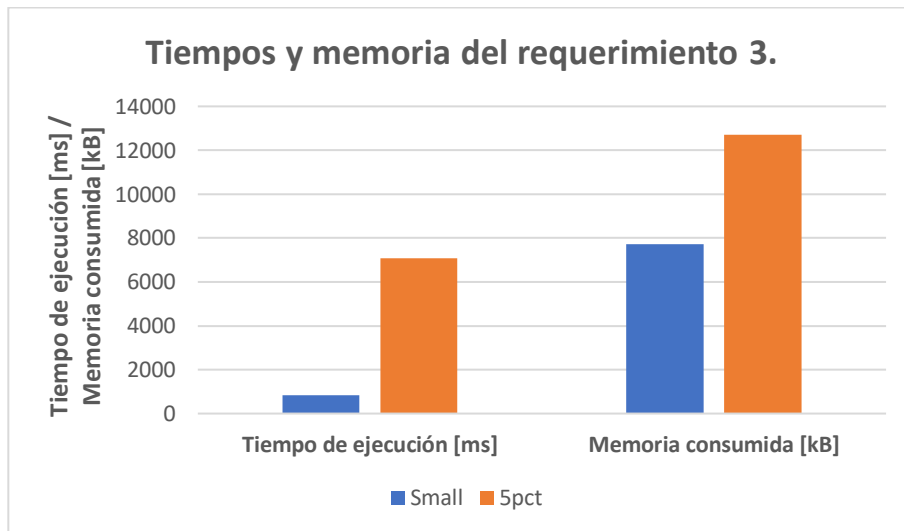
- Requerimiento 1:



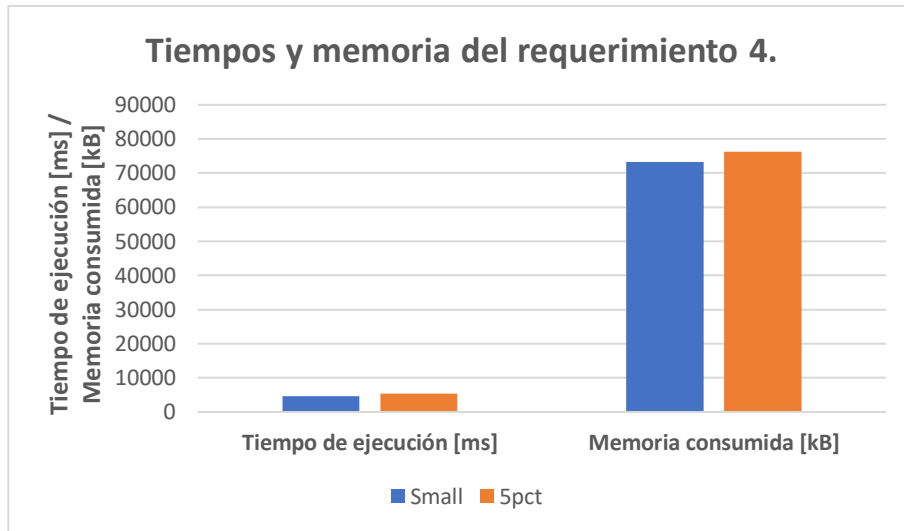
- Requerimiento 2:



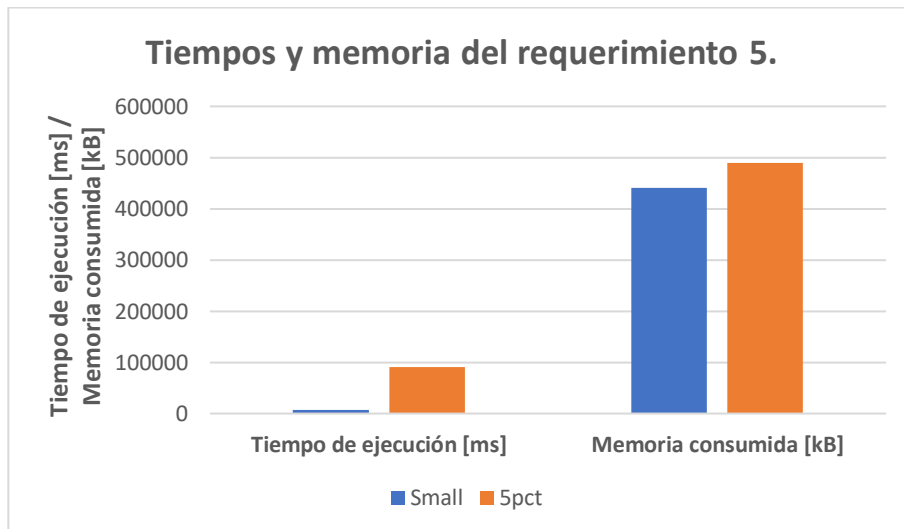
- Requerimiento 3:



- Requerimiento 4:



- Requerimiento 5:



De manera general se puede determinar lo siguiente:

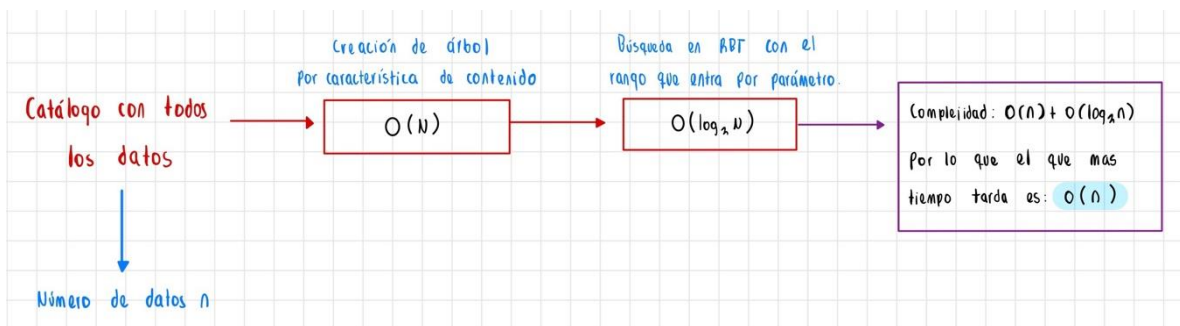
Al observar los resultados expuestos con anterioridad, se puede determinar que para ambas máquinas el consumo de memoria y el tiempo de ejecución de la carga de datos y de los 5 requerimientos aumenta a medida que aumenta la cantidad de datos que se deben procesar. Por otro lado, para ambas máquinas y para todos los requerimientos, junto con la carga de datos, se puede apreciar que existe una diferencia significativa entre los tiempos de ejecución y la memoria consumida. Esto se debe a que en todos los resultados, el consumo de memoria es de mayor magnitud a comparación de los tiempos de ejecución, los cuales suelen ser muy bajos.

Por otro lado, sobre la variedad de estructuras de datos en nuestro programa, se escogieron únicamente tablas de hash de tipo PROBING ya que consumen menos memoria puesto que,

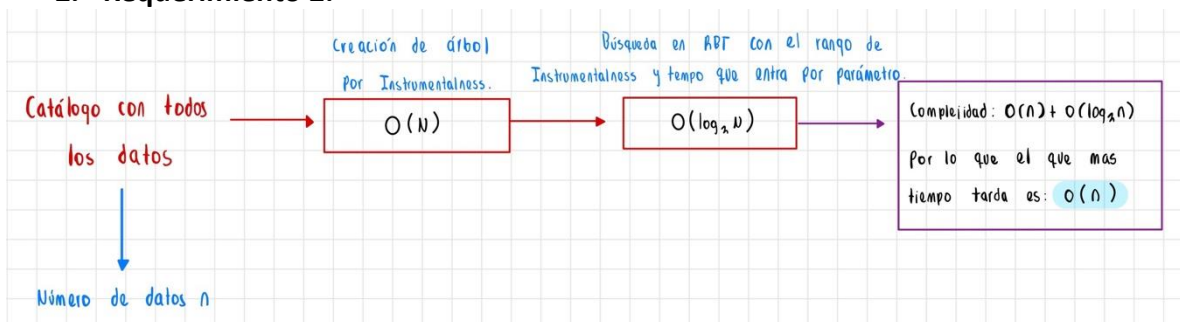
a diferencia de las tablas de tipo CHAINING, estas no crean una lista encadenada cuando ocurre una colisión, sino que buscan un espacio disponible. Además, se eligió factor de carga 0.5 para mantener la complejidad lo más constante posible. Por otro lado, cuando hablamos de listas utilizamos ARRAY's, ya que consumen menos memoria y además es más fácil llegar a un elemento puesto que éstos se guardan en posiciones contiguas en memoria. Para cada arreglo la función de comparación varía dependiendo del requerimiento. Finalmente, los árboles que se implementaron en este reto son de tipo RBT, esto se debe principalmente a que estos aseguran que el árbol se encuentre siempre parcialmente balanceado. Por consiguiente, en un RBT las búsquedas de información son más rápidas y la construcción de estos árboles es mas lenta que la de los árboles de tipo BST. Es por esto, que los tiempos de ejecución de cada uno de los requerimientos tarda menos que el tiempo de ejecución de la carga del catálogo.

A continuación se presenta el análisis de complejidad de cada uno de los requerimientos del reto 3:

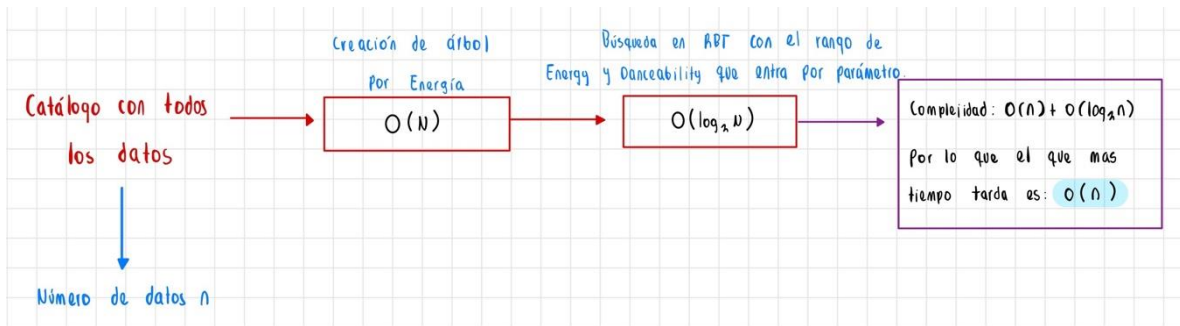
1. Requerimiento 1:



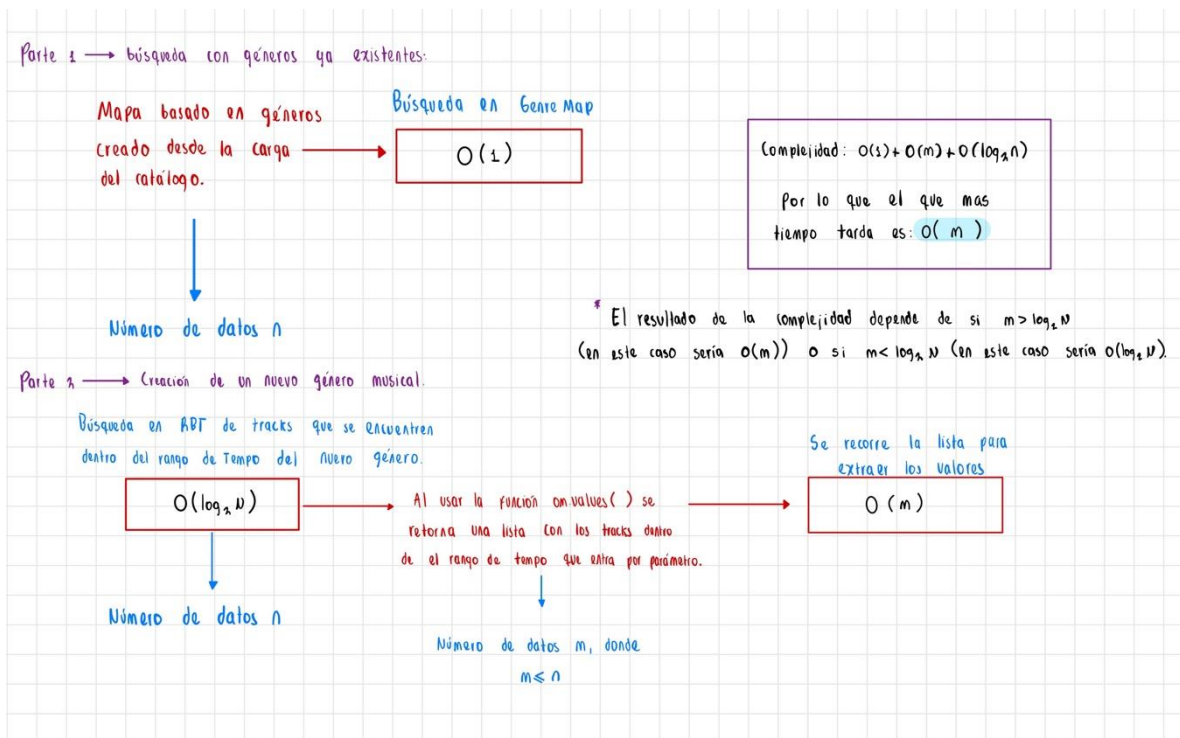
2. Requerimiento 2:



3. Requerimiento 3:



4. Requerimiento 4:



5. Requerimiento 5:

