

Documento Análisis
Estructura de datos y algoritmos
Reto 3

Integrantes:

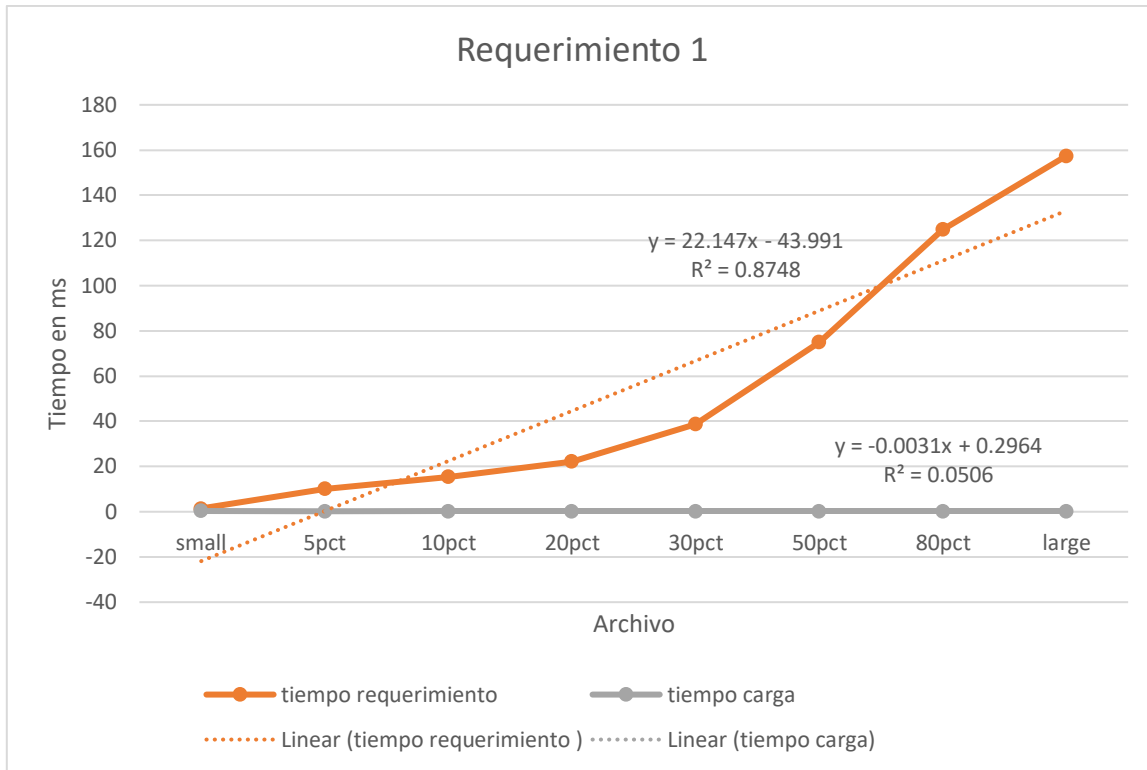
Juan Manuel Jauregui Rozo - j.jauregui@uniandes.edu.co (requerimiento 2)

Sofía Escobar Tamayo - s.escobart@uniandes.edu.co (requerimiento 3)

Análisis de complejidad de cada requerimiento:

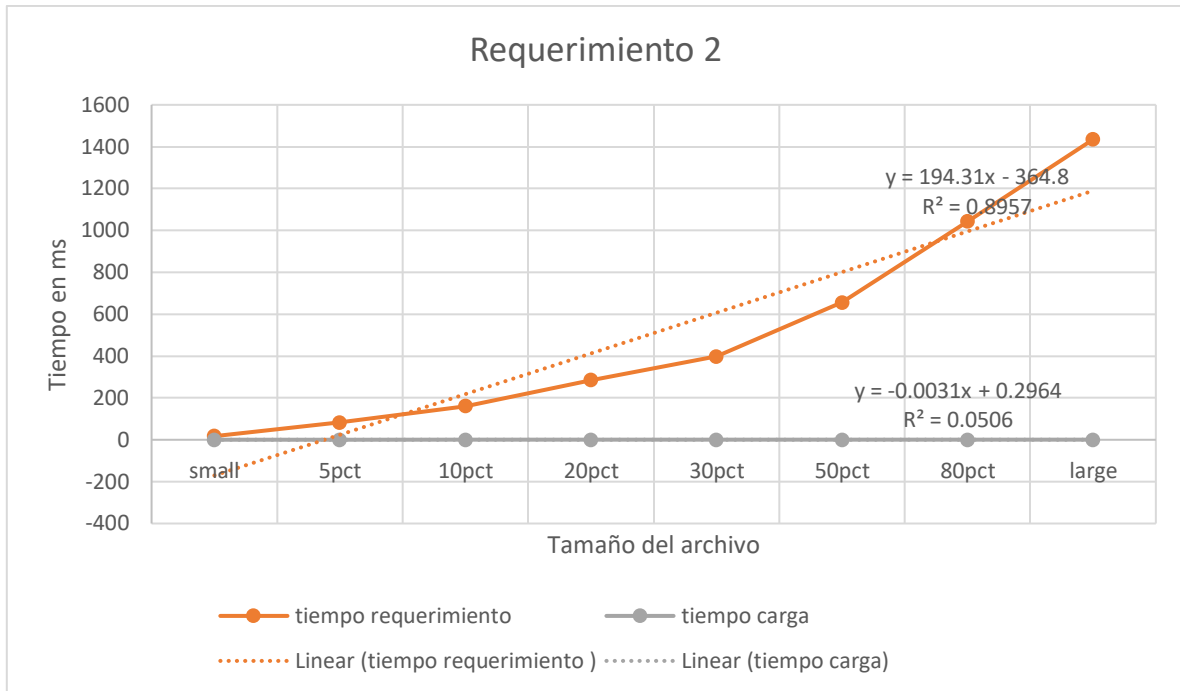
Requerimiento 1: Al analizar el código del requerimiento 1 se observan 3 funciones que se usan para este, la principal es GetUfosByCity que es una función que recibe el mapa y la ciudad en cuestión para retornar el total de las ciudades, el total de casos en la ciudad dada, los primeros 3 y los últimos 3 avistamientos en esa ciudad. La complejidad del requerimiento es $O(N\log(N))$ ya que esta función principal determina la complejidad en su mayoría y tiene dos ciclos distintos pero un ciclo es de complejidad $O(K)$ y el otro es $O(N)$ además, tiene un sort de complejidad $O(N\log(N))$, por lo tanto, la función es de complejidad temporal $O(N\log(N))$.

archivo	Datos	Tiempo requerimiento (ms)	Tiempo carga (ms)
small	803	1,47	0,34
5pct	4016	10,17	0,22
10pct	8033	15,48	0,3
20pct	16066	22,2	0,28
30pct	24099	38,68	0,29
50pct	40166	75,08	0,29
80pct	64265	124,86	0,27
large	80332	157,41	0,27



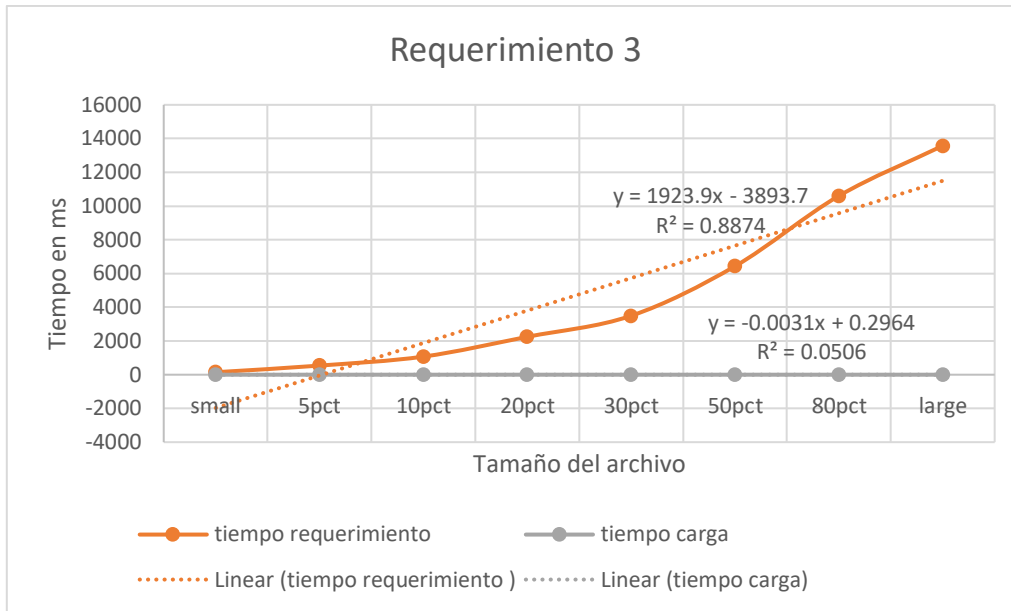
Requerimiento 2: Al analizar el código del requerimiento 2 se observan 3 funciones, la principal es GetUfosByDuration. Esta función recibe el mapa, el limite inferior de la duración del avistamiento y el limite superior. Dado que la función principal va a determinar la complejidad temporal y esta función cuenta con cinco ciclos $O(N)$ y un sort $O(N\log(N))$ entonces el orden de complejidad temporal del requerimiento es $O(N\log(N))$. Aunque se pueda pensar que el doble ciclo al inicio de la función es de complejidad $O(N^2)$, esto es relativo ya que como la complejidad se mide basándose en el total de los datos, podemos ver como este ciclo únicamente reconstruye linealmente estos datos. Sabiendo lo anterior, se va a considerar el doble ciclo como complejidad temporal $O(N)$.

archivo	Datos	Tiempo requerimiento (ms)	Tiempo carga (ms)
small	803	17,29	0,34
5pct	4016	82,06	0,22
10pct	8033	161,07	0,3
20pct	16066	284,66	0,28
30pct	24099	398,37	0,29
50pct	40166	656,52	0,29
80pct	64265	1042,14	0,27
large	80332	1434,66	0,27



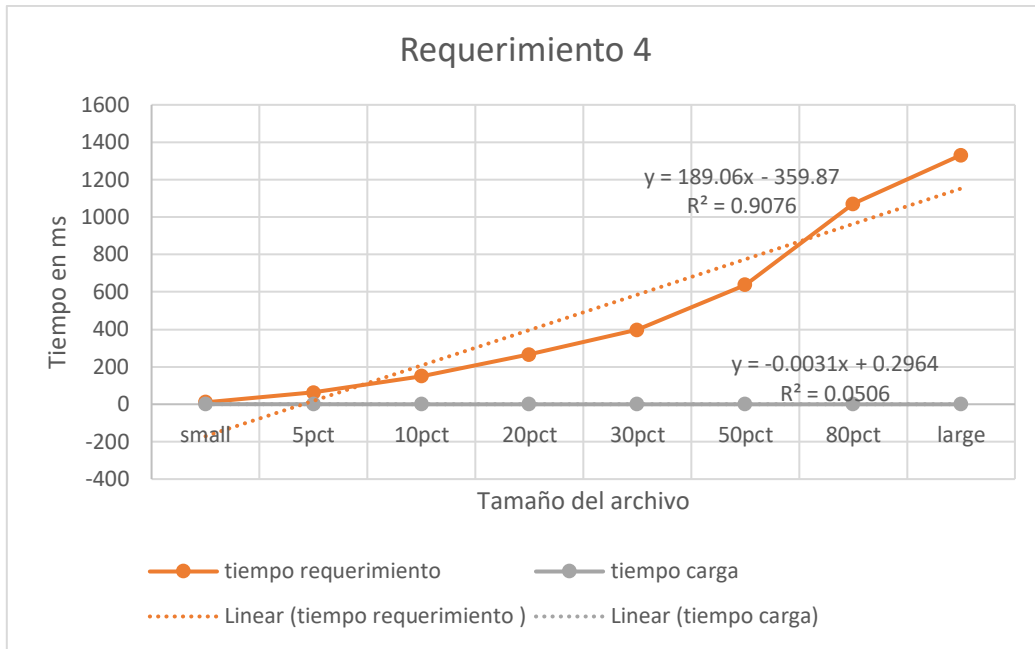
Requerimiento 3: Al analizar el código del requerimiento 3 se observan 3 funciones, la principal es GetUfosByTime. Esta función recibe el mapa, el limite inferior de la fecha del avistamiento y el limite superior. Dado que la función principal va a determinar la complejidad temporal y esta función cuenta con cinco ciclos $O(N)$ y un sort $O(N\log(N))$ entonces el orden de complejidad temporal del requerimiento es $O(N\log(N))$. Aunque se pueda pensar que el doble ciclo al inicio de la función es de complejidad $O(N^2)$, esto es relativo ya que como la complejidad se mide basándose en el total de los datos, podemos ver como este ciclo únicamente reconstruye linealmente estos datos. Sabiendo lo anterior, se va a considerar el doble ciclo como complejidad temporal $O(N)$.

archivo	Datos	Tiempo requerimiento (ms)	Tiempo carga (ms)
small	803	147,59	0,34
5pct	4016	540,92	0,22
10pct	8033	1073,71	0,3
20pct	16066	2246,48	0,28
30pct	24099	3494,89	0,29
50pct	40166	6432,54	0,29
80pct	64265	10600,97	0,27
large	80332	13573,57	0,27



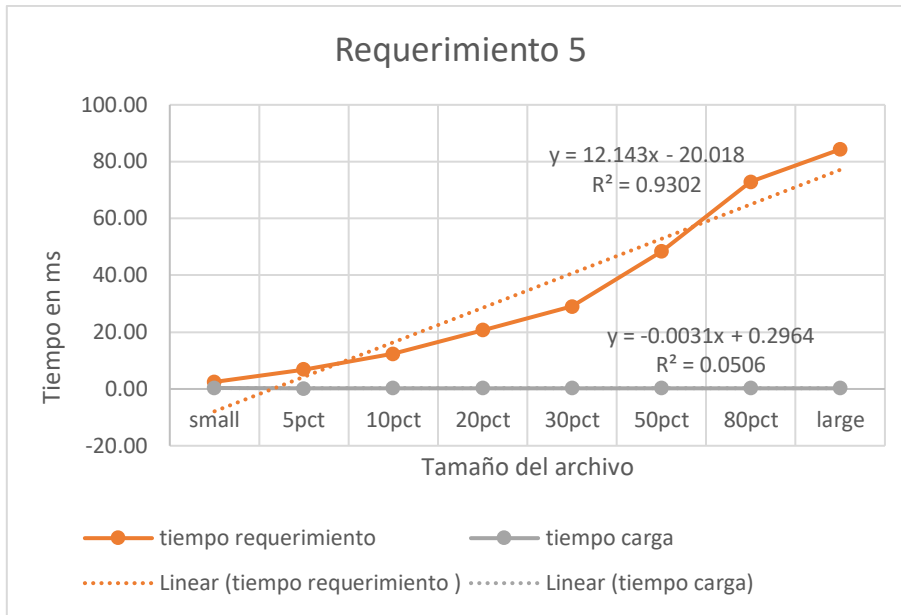
Requerimiento 4: Al analizar el código del requerimiento 4 se observan 3 funciones, la principal es GetUfosByDateTime. Esta función recibe el mapa, el limite inferior de la hora del avistamiento y el limite superior. Dado que la función principal va a determinar la complejidad temporal y esta función cuenta con cinco ciclos $O(N)$ y un sort $O(N\log(N))$ entonces el orden de complejidad temporal del requerimiento es $O(N\log(N))$. Aunque se pueda pensar que el doble ciclo al inicio de la función es de complejidad $O(N^2)$, esto es relativo ya que como la complejidad se mide basándose en el total de los datos, podemos ver como este ciclo únicamente reconstruye linealmente estos datos. Sabiendo lo anterior, se va a considerar el doble ciclo como complejidad temporal $O(N)$.

archivo	Datos	Tiempo requerimiento (ms)	Tiempo carga (ms)
small	803	10,35	0,34
5pct	4016	63,20	0,22
10pct	8033	150,49	0,3
20pct	16066	265,45	0,28
30pct	24099	397,45	0,29
50pct	40166	637,50	0,29
80pct	64265	1071,06	0,27
large	80332	1331,56	0,27



Requerimiento 5: Al analizar el código del requerimiento 5 se observan 3 funciones, la principal es GetUfosByLonLat. Esta función recibe el mapa, el limite inferior de la latitud, y de la longitud del avistamiento y el limite superior de ambos. En este caso, la función tiene un doble ciclo al inicio que es de complejidad temporal $O(N^2)$, por lo tanto, el orden de complejidad temporal de la función es de $O(N^2)$ y sabiendo que esta es la función principal entonces el orden de complejidad temporal del requerimiento $O(N^2)$.

archivo	Datos	Tiempo requerimiento (ms)	Tiempo carga (ms)
small	803	2,47	0,34
5pct	4016	6,83	0,22
10pct	8033	12,37	0,3
20pct	16066	20,67	0,28
30pct	24099	29,00	0,29
50pct	40166	48,37	0,29
80pct	64265	72,96	0,27
large	80332	84,33	0,27



BONO: Al analizar el código del requerimiento 6 se observan 3 funciones, la principal es GetUfosByLonLat. Esta función recibe el mapa, el limite inferior de la latitud, y de la longitud del avistamiento y el limite superior de ambos. En este caso, la función tiene un doble ciclo al inicio que es de complejidad temporal $O(N^2)$, por lo tanto, el orden de complejidad temporal de la función es de $O(N^2)$ y sabiendo que esta es la función principal entonces el orden de complejidad temporal del requerimiento $O(N^2)$. Esta es igual a la del requerimiento 5 porque todo el bono esta basado en lo que se hace en el requerimiento 5, lo único que cambia es la generada del mapa y esto no afecta el orden de complejidad temporal.

El archivo mapita.html va a aparecer en la carpeta del reto 3 (no va a estar dentro de ninguna subcarpeta) y se debe hacer click en este archivo que va a abrir el mapa en el navegador.

Adjunto evidencia de como se ve el mapa:

