

*Análisis de Resultados y
Pruebas de rendimiento*

Reto 01

Universidad de los Andes, DISC.

Bogotá DC

10/03/2022

Tabla de contenido

1. Prefacio.....	Pág. 3
1.1 Integrantes, objetivos.....	Pág.3
2. Análisis y pruebas de rendimiento.....	Pág. 4 – 8
2.1 Análisis y pruebas requisito No 1.....	Pág. 4
2.1.1 Tablas de tiempos de ejecución.....	Pág. 4
2.2 Análisis y pruebas requisito No 2.....	Pág. 5
2.2.1 Tablas de tiempos de ejecución.....	Pág. 5
2.3 Análisis y pruebas requisito No 3.....	Pág. 6
2.3.1 Tablas de tiempos de ejecución.....	Pág. 6
2.4 Análisis y pruebas requisito No 4.....	Pág. 7
2.4.1 Tablas de tiempos de ejecución.....	Pág. 7
2.5 Análisis y pruebas requisito No 5.....	Pág. 8
2.5.1 Tablas de tiempos de ejecución.....	Pág. 8
3. Graficas de los tiempos de ejecución.....	Pág. 9-13
4. Análisis de la complejidad de los resultados.....	Pág. 14

Documentado y presentado Por:

- Juan Coronel. 202111207. J.coronel@uniandes.edu.co
- Santiago Rodríguez. 202110340. s.rodriguezr234@uniandes.edu.co

Diseño de requerimientos individuales:

- Los requerimientos individuales fueron realizados por Juan Coronel.

Objetivos del análisis:

- Evidenciar mediante métricas empíricas de qué manera se ve influenciado la complejidad temporal de un algoritmo, al variar sus parámetros de construcción. Por ejemplo, evidenciar el comportamiento del sistema cuando se usa: distintas listas (ARRAY_LIST, SINGLE_LINKED), distintos algoritmos de ordenamiento (Iterativos o Recursivos), muestras, entre otros.

Información Maquinas de trabajo:

Máquina 1	
Procesadores	Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz 3.00 GHz
Memoria RAM (GB)	8,00 GB
Sistema Operativo	Windows 10 Pro 64xbits

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

A continuación, se presentarán los análisis obtenidos en la *MÁQUINA 1* del estudiante Juan Coronel...

REQ No 1

Tablas de tiempos de ejecución

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	5398,32	1023,32	280.12	309.91	219.03
5.00%	22892	459084.03	293890,92	10390.20	8092.78	7059.74
10.00%	41714	2309045.42	1090490,43	20939.03	21002.23	18084.45
20.00%	73731			80238.00	93228.01	79399.42
30.00%	101276			123892.23	156340.40	129320.00
50.00%	147521			237872.14	178902.24	159772.54
80.00%	203011			340980.46	309239.32	281353.11
100.00%	233568			500299.32	385244.54	347132.35

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	53498.89	43003.03	8009.32	7042.00	4001.65
5.00%	1270	1093890.02	1384590.11	190389.85	79802.73	70932.92
10.00%	22892			1609422.20	445093.92	349052.92
20.00%	41714				1400223.77	1200389.02
30.00%	73731					
50.00%	101276					
80.00%	147521					
100.00%	203011					

Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	$O(n \log(n))$	$O(n)$
Quick sort	$O(n \log(n))$	$O(n^{1/2})$
Shell Sort [ms]	$O(n \log(n))$	$O(n)$
Selection Sort [ms]	$O(n^2)$	O
Insertion Sort [ms]	$O(n^2)$	$O(n^2)$

Anexo: el tamaño de la muestra se tomó como la suma de los 3 catálogos: artistas, canciones y álbumes. Esto, para generalizar que el tamaño de la muestra se aplica sobre estas listas en conjunto. Además, se usó siempre N como 10, fechas del 2000 al 2010, la banda clean bandit para todas las pruebas.

REQ No 2

Tablas de tiempos de ejecución

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	19,92	32.32	3.32	2.23	2.11
5.00%	22892	21,91	34.00	4,95	2.90	2.58
10.00%	41714	23.33	36.12	6.09	4.28	3.99
20.00%	73731	26,40	38.02	6.91	5.40	4.23
30.00%	101276	29.32	39,23	7.28	5.49	4.45
50.00%	147521	31.02	40.11	7.92	6.92	4.68
80.00%	203011	33.92	42.12	8,02	7.07	4.78
100.00%	233568	34.12	42.97	9.01	7.89	4.86

Tabla 4. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	102.32	109.31	4.22	3.91	3.50
5.00%	22892	23149.23	42874.02	62.32	42.52	39.03
10.00%	41714	921032.31	1092382.47	91.01	90.32	83.92
20.00%	73731	1239023.55		122.07	132,94	110.02
30.00%	101276			502.93	260.21	230.21
50.00%	147521			1132,85	439.23	320.93
80.00%	203011			1711.98	503.33	403.21
100.00%	233568			2319.29	600.81	423.02

Tabla 5. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	$O(1)$	$O(1)$
Quick sort	$O(n \log(n))$	$O(n^{3/2})$
Shell Sort [ms]	$O(1)$	$O(1)$
Selection Sort [ms]	$O(n)$	$O(n^2)$
Insertion Sort [ms]	$O(n)$	$O(n^2)$

REQ No 3

Tablas de tiempos de ejecución

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
0.50%	2655	12.14	30.41	9.14	10.23	8.98
5.00%	22892	15.15	36.44	10.42	10.18	10.03
10.00%	41714	16.72	38.67	11.57	11.86	10.51
20.00%	73731	16.37	49.23	11.89	12.27	10.93
30.00%	101276	17.99	52.21	12.56	13.13	12.02
50.00%	147521	19.10	54.11	12.92	13.71	12.48
80.00%	203011	21.00	60.46	13.14	13.77	12.12
100.00%	233568	23.34	61.95	13.11	14.91	12.74

Tabla 6. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	39.01	46.14	13.23	12.16	12.54
5.00%	22892	45.57	59.42	24.17	23.72	23.23
10.00%	41714	56.09	93.75	45.25	45.13	44.21
20.00%	73731	78.90	102.52	53.84	53.14	52.75
30.00%	101276	90.54	145.78	59.81	57.45	56.91
50.00%	147521	101.72	192.66	62.90	59.53	59.23
80.00%	203011	172.23	203.19	67.21	60.71	60.11
100.00%	233568	201.61	221.01	71.24	63.75	62.23

Tabla 7. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	$O(1)$	$O(1)$
Quick sort	$O(n \log(n))$	$O(n)$
Shell Sort [ms]	$O(1)$	$O(1)$
Selection Sort [ms]	$O(1)$	$O(n)$
Insertion Sort [ms]	$O(1)$	$O(n)$

REQ No 4

Tablas de tiempos de ejecución

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	2315.16	4125.13	35.78	24.45	22.19
5.00%	22892	41241.81	51255.15	560.11	280.46	250.08
10.00%	41714	129832.49	230914.61	874.34	410.81	400.23
20.00%	73731	203290.03	501969.49	1206.68	1011.16	902.21
30.00%	101276	701923.14	1030108.28	2018.35	1902.98	1603.11
50.00%	147521			3013.17	2501.11	2139.01
80.00%	203011			5052.12	4219.13	4023.42
100.00%	233568			7821.51	6001.41	5767.58

Tabla 8. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	3021.13	4209.32	200.23	159.13	157.74
5.00%	22892	25023.62	48921.72	5051.14	3331.64	2313.21
10.00%	41714	923812.77	1028313.16	204219.15	102302.66	93201.32
20.00%	73731			982367.61	454021.32	309123.32
30.00%	101276				1309202.12	1029132.21
50.00%	147521					
80.00%	203011					
100.00%	233568					

Tabla 9. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	$O(n \log(n))$	$O(n^2)$
Quick sort	$O(n)$	$O(n^2)$
Shell Sort [ms]	$O(n \log(n))$	$O(n^2)$
Selection Sort [ms]	$O(n^2)$	$O(n^2)$
Insertion Sort [ms]	$O(n^2)$	$O(n^2)$

REQ No 5

Tablas de tiempos de ejecución

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	231523.88	323982.26	291.12	102.16	75.12
5.00%	22892	349012.02	4298741.61	4509.51	2039.31	1733.45
10.00%	41714	930212.17		12350.22	9023.68	8023.16
20.00%	73731			25161.67	20125.37	16010.23
30.00%	101276			90211.74	72302.18	70231.14
50.00%	147521			201592.34	160231.51	142561.12
80.00%	203011			291202.23	201315.22	182514.51
100.00%	233568			500023.86	416586.21	223597.32

Tabla 10. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación arreglo.

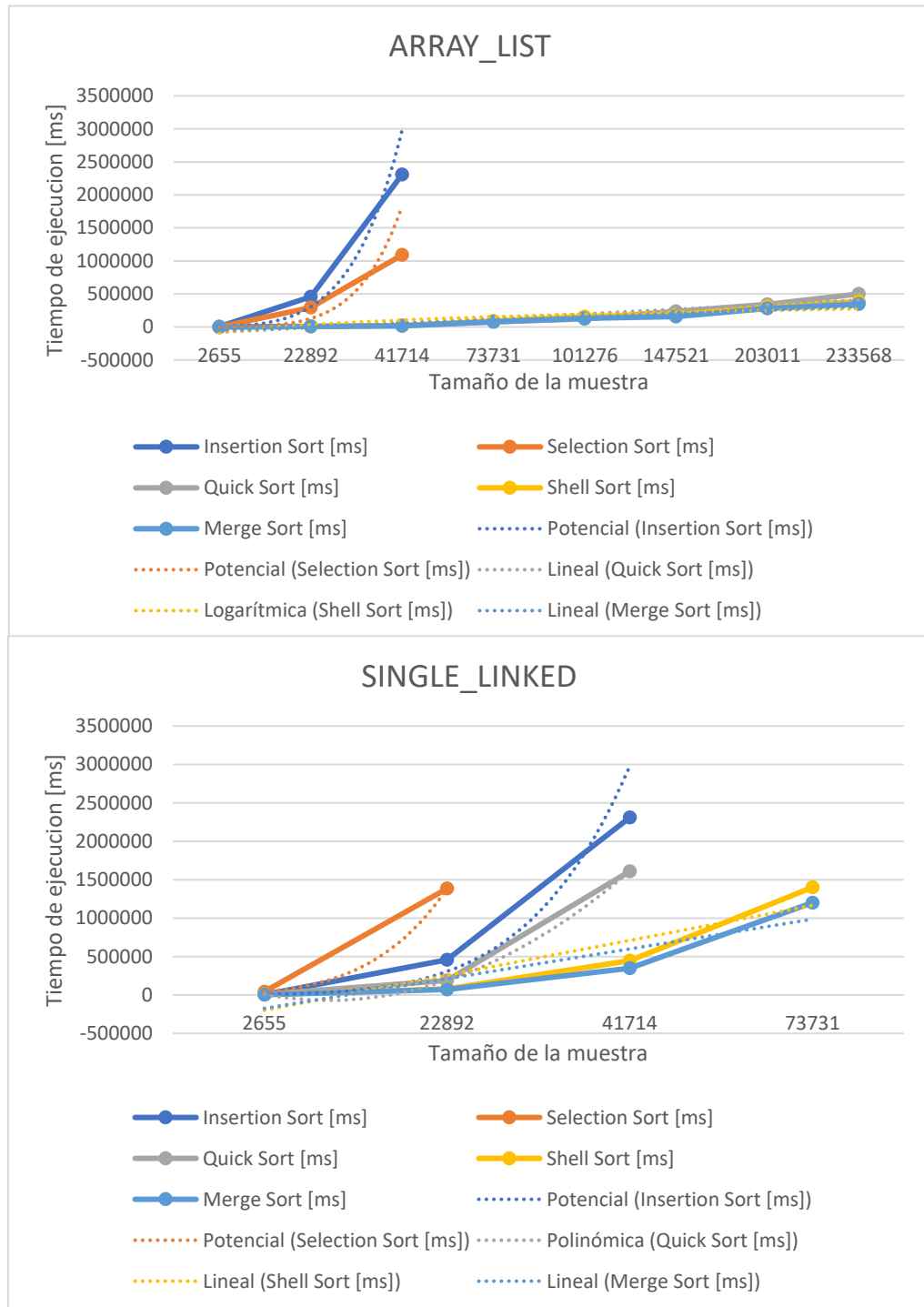
Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Quick Sort [ms]	Shell Sort [ms]	Merge Sort [ms]
0.50%	2655	120392.10	192840.66	49382.53	798.12	686.09
5.00%	22892			670292.18	48203.57	32389.16
10.00%	41714				609897.12	429012.33
20.00%	73731					1092321.51
30.00%	101276					
50.00%	147521					
80.00%	203011					
100.00%	233568					

Tabla 11. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

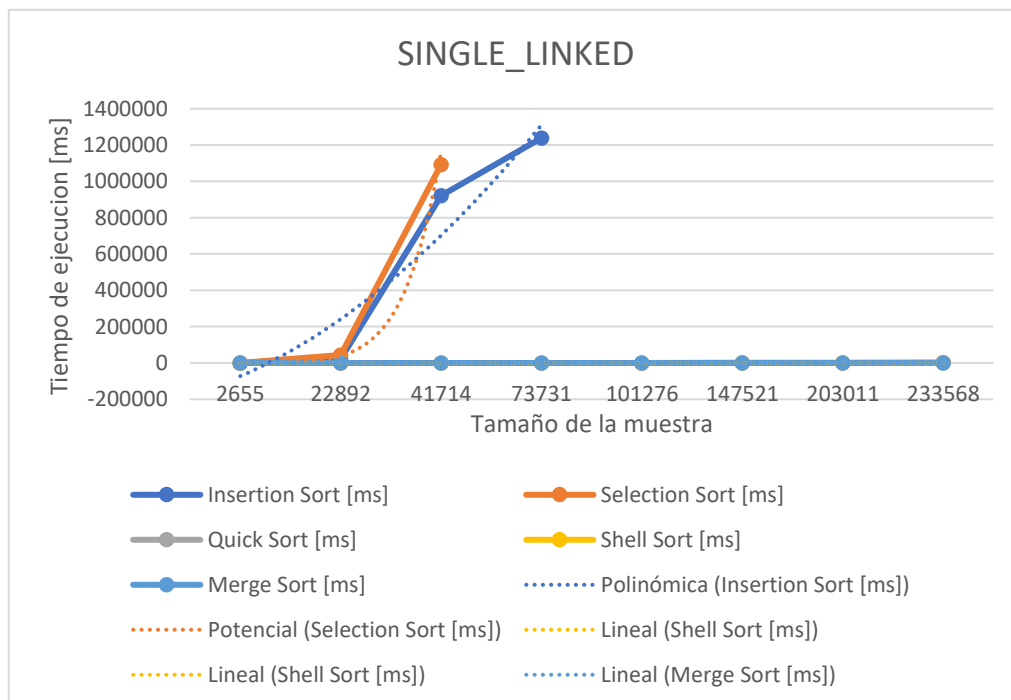
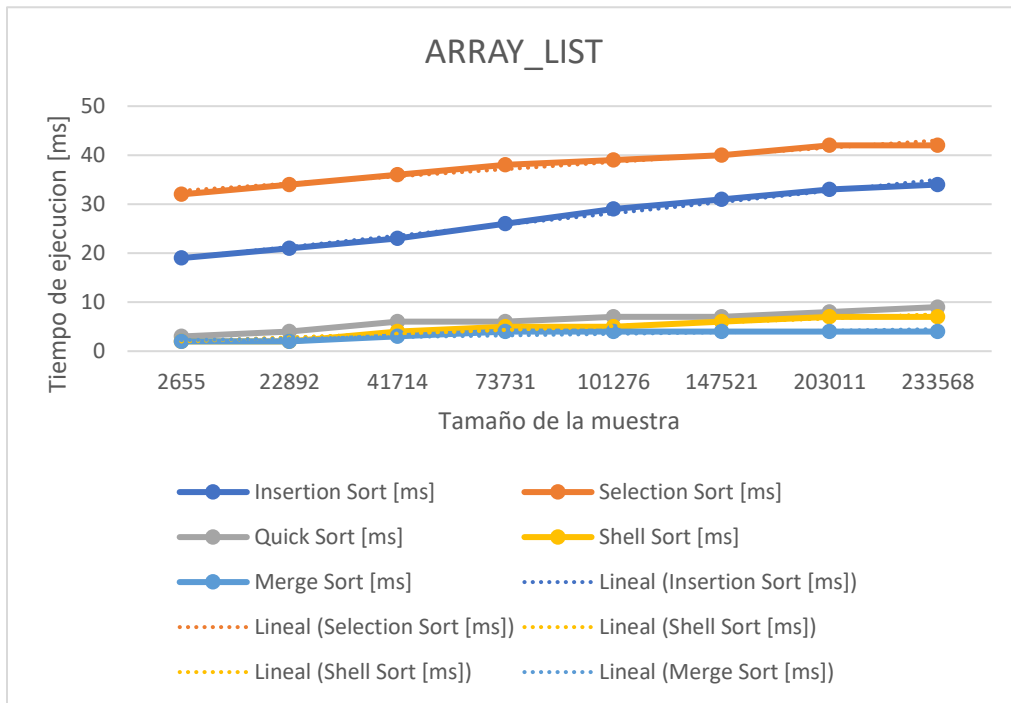
Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	$O(n)$	$O(n^2)$
Quick sort	$O(n)$	$O(n^2)$
Shell Sort [ms]	$O(n)$	$O(n^2)$
Selection Sort [ms]	$O(n)$	$O(n^2)$
Insertion Sort [ms]	$O(n^2)$	$O(n^2)$

Graficas analíticas de resultados

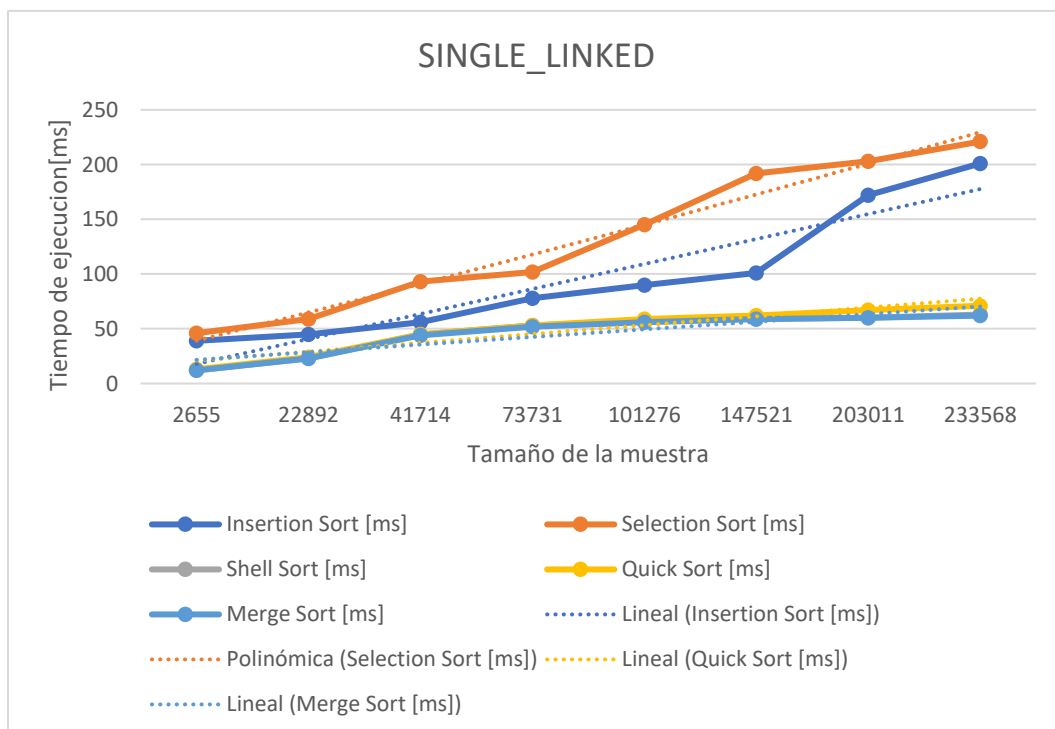
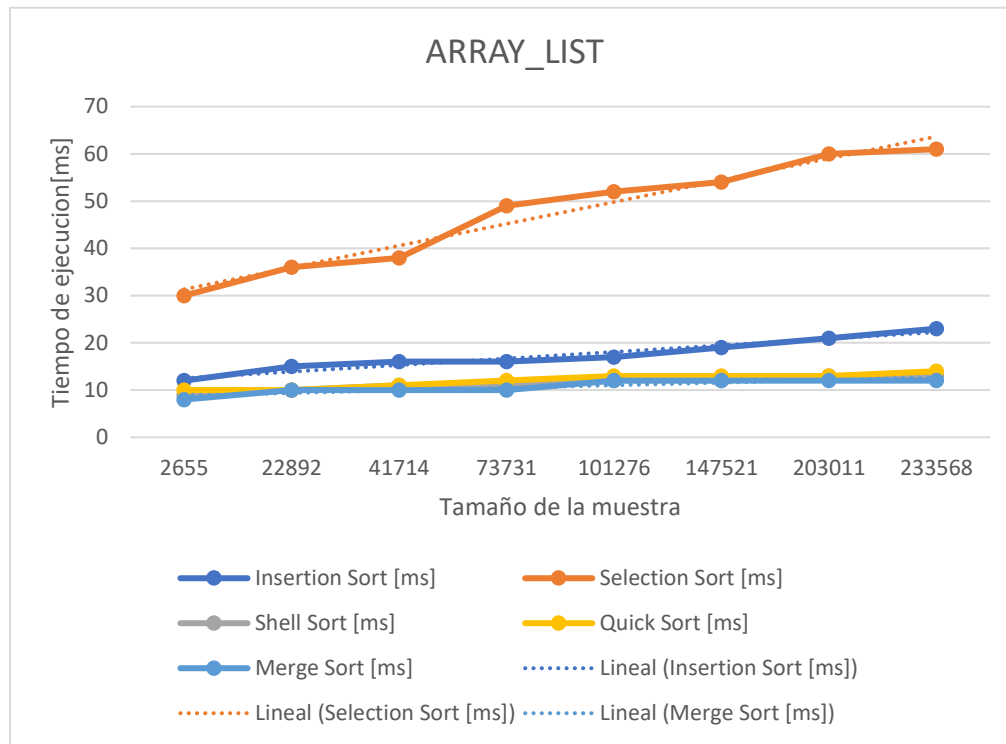
REQ 1



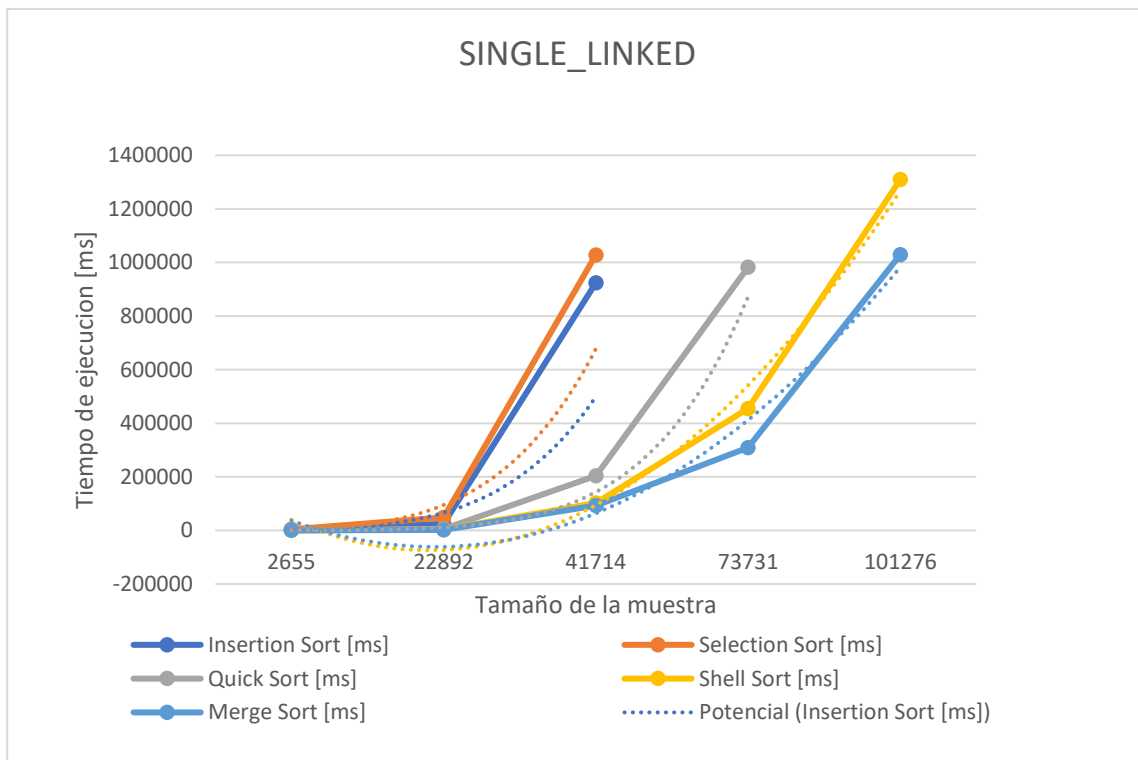
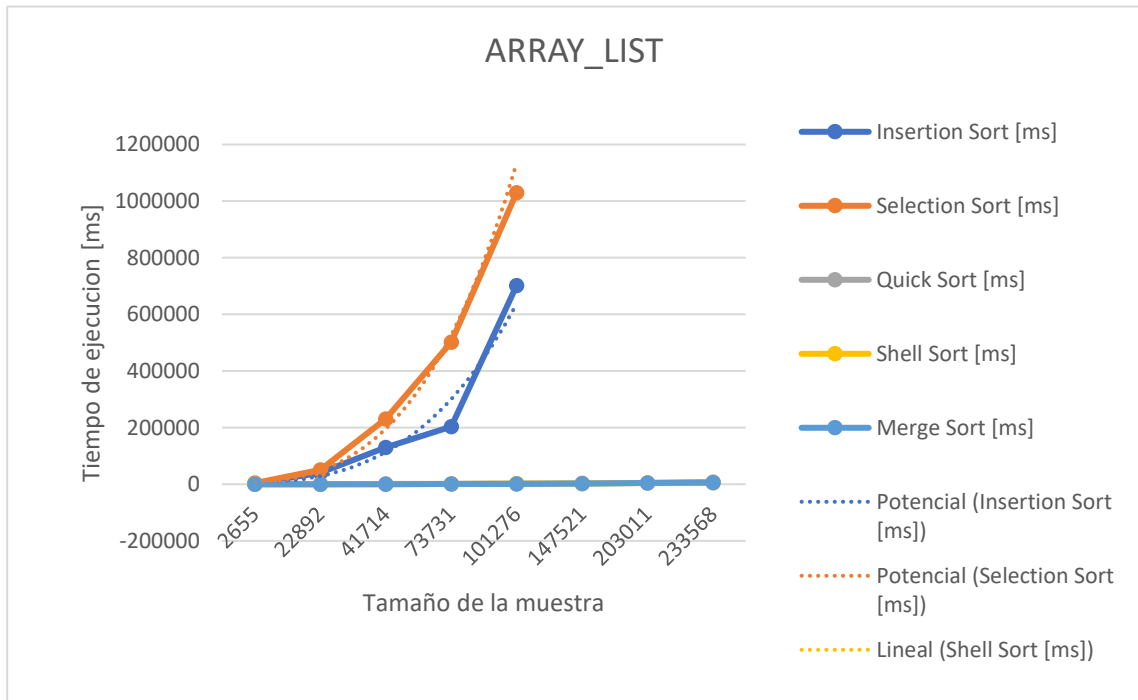
REQ 2



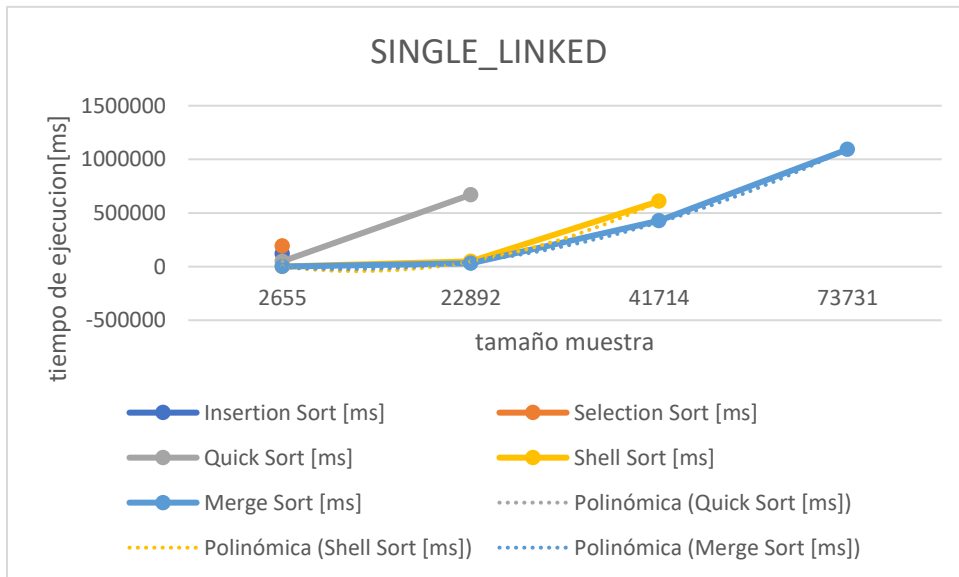
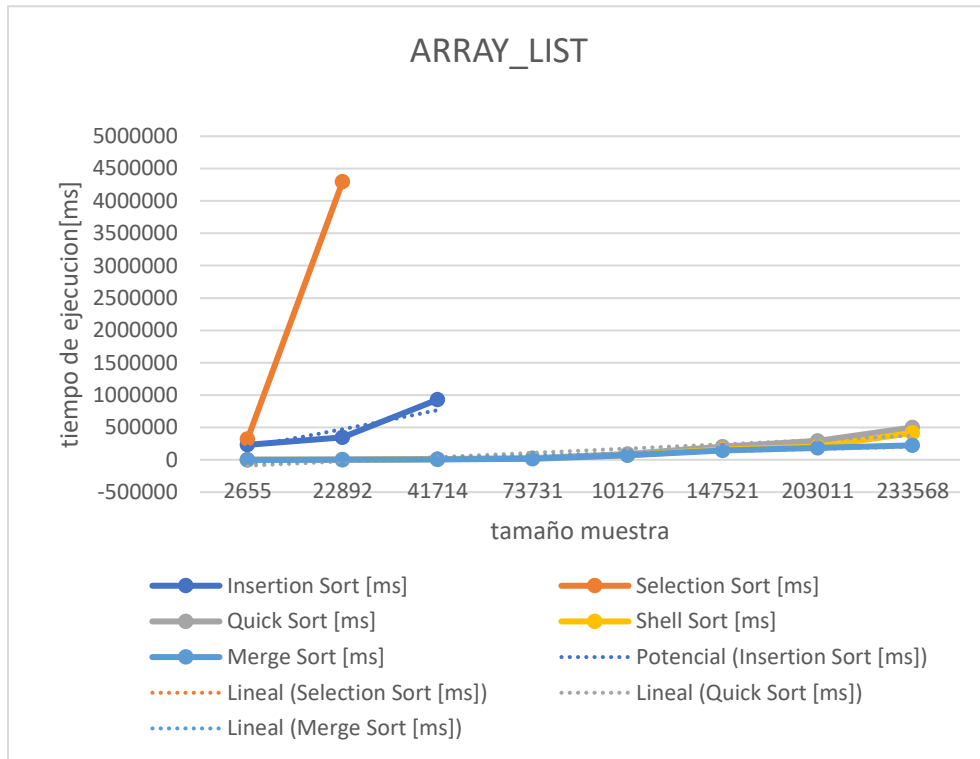
REQ 3



REQ 4



REQ 5



Resultados de análisis

¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?

R//

Es evidenciable que las graficas describen el comportamiento de los algoritmos según la teoría. Siendo así, los sort merge y Shell los mas apropiados para realizar las funciones de los requerimientos. Las tendencias de las líneas en las graficas se tiñe a lo presentado en la teoría de la clase. El REQ 2 es el que menos tiempo tarda en ejecutarse debido a que las funciones se ejecutan de manera casi que constante, en $O(1)$ o $O(n)$.

Sin embargo, si somos objetivos, nos damos cuenta de que los resultados en las graficas no son 100% precisas respecto a su tendencia. Esto se da debido a que hay funciones en las cuales algunos shorts no se evaluaban correctamente más allá del 5pct de la cantidad de archivos del catálogo. Es decir, no es una buena practica correr funciones con sorts que no son los adecuados para el proceso.

Finalmente, cabe recalcar que los tiempos de procesamiento dependen del tipo de maquina en el que se este desarrollando. Así pues, la toma de datos en solo un sistema puede generar respuestas arbitrarias. Es por eso, que, con fines de optimización, en el código del reto se interpreta todas las funciones con el sort Merge, así se genera mayor consistencia en los resultados.