

OBSERVACIONES DE LA PRACTICA

Estudiante 1: Ehimar Andres Vargas Malaver, 202014902, e.vargasm@uniandes.edu.co

Estudiante 2: Brigitte Juliana Arcos Gómez, 202014103, b.arcos@uniandes.edu.co

	Máquina 1	Máquina 2
Procesadores	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Memoria RAM (GB)	8.00 GB (6.94 GB utilizable)	8,00 GB (7,88 GB usable)
Sistema Operativo	Windows 10 Home Insider Preview	Windows 10

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

Maquina 1

Resultados

Tamaño de la muestra (ARRAYLIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
1000	531.25	515.63	31.25	15.63	31.25
2000	2296.88	2140.63	46.88	46.88	62.50
4000	9140.63	8781.25	171.87	109.38	125.00
8000	37828.13	39296.88	375.00	234.38	250.00
16000	162562.50	173750.00	890.63	500.00	562.50
32000	666890.25	713406.25	2156.25	1078.13	1187.50
64000			5109.38	2453.13	2578.13
128000			12765.63	4984.38	5593.75
256000			30250.00	10906.25	11953.13
512000					

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

Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
1000	37000.00	60875.00	3218.75	2765.63	187.50
2000	304781.25	468593.75	16312.50	10468.75	687.50
4000			75015.63	49562.50	2796.88
8000			331921.88	214109.38	11531.25
16000				947921.88	49234.38
32000					200828.13
64000					807906.25
128000					
256000					

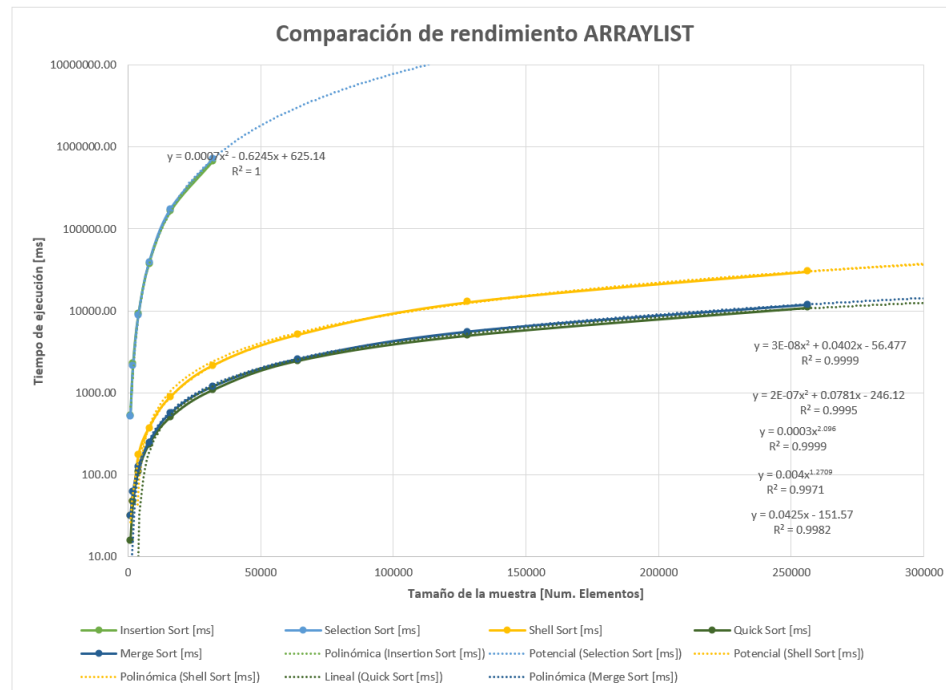
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	X	
Quick sort	X	

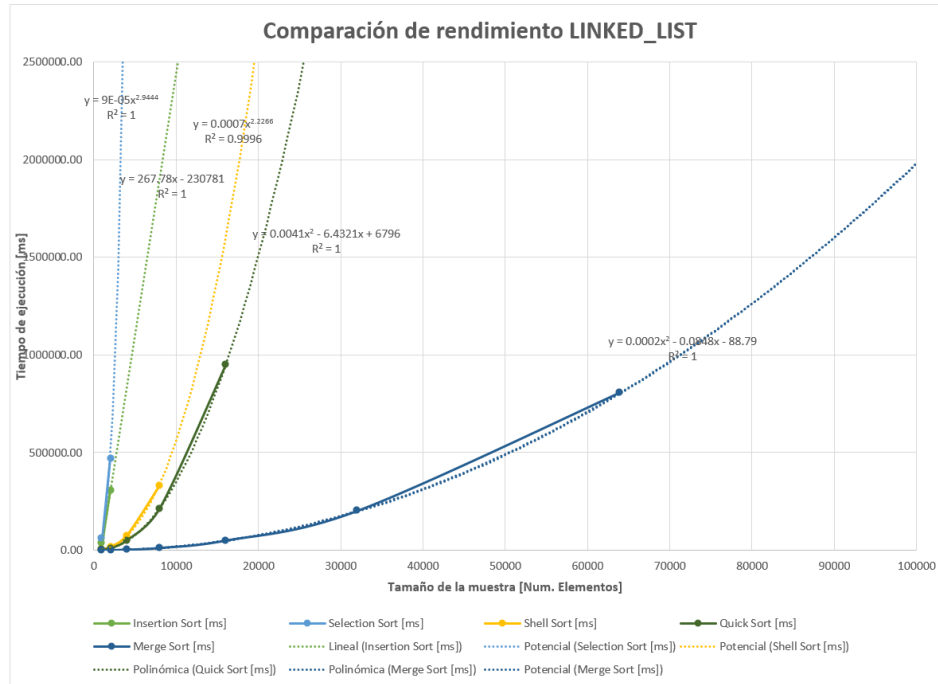
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Graficas

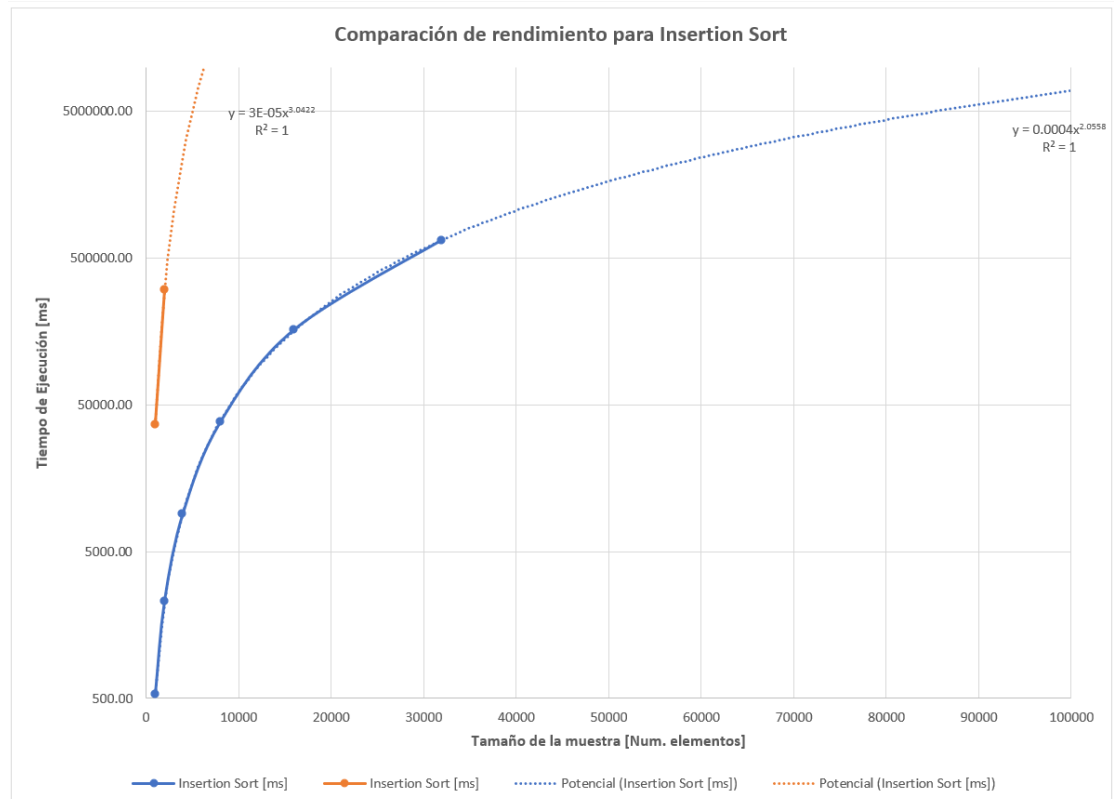
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1**.
 - Comparación de rendimiento ARRAYLIST.



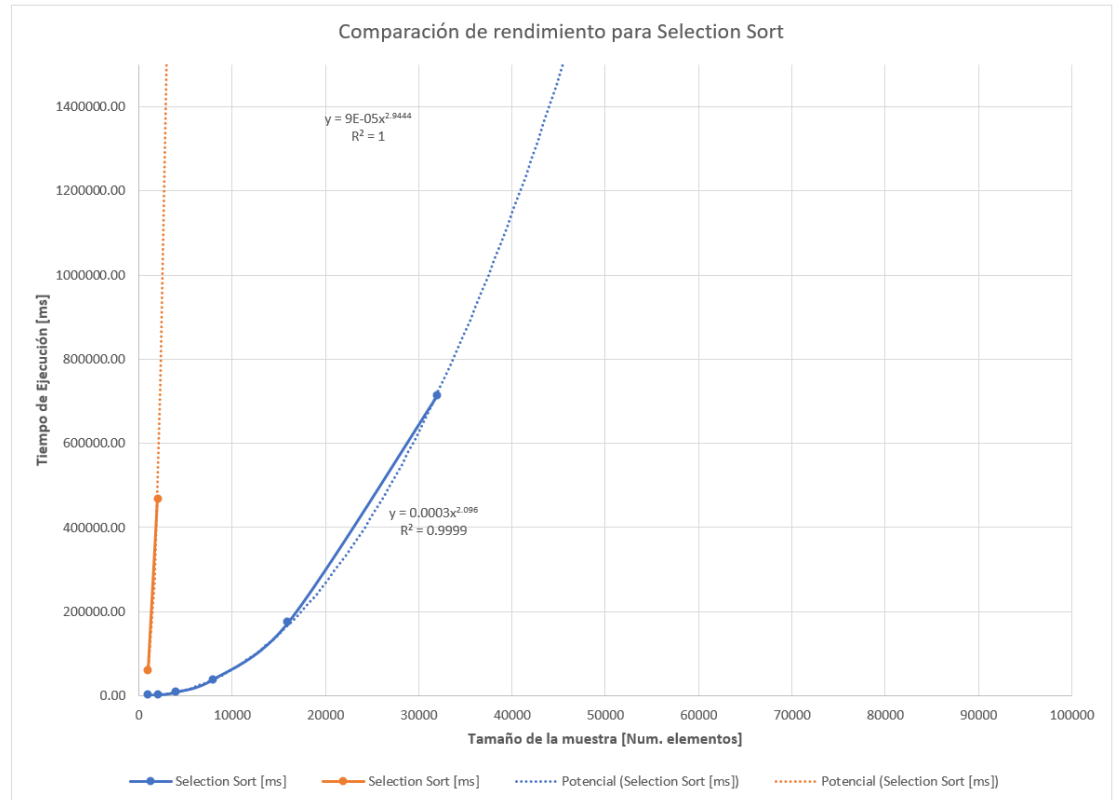
- Comparación de rendimiento LINKED_LIST.



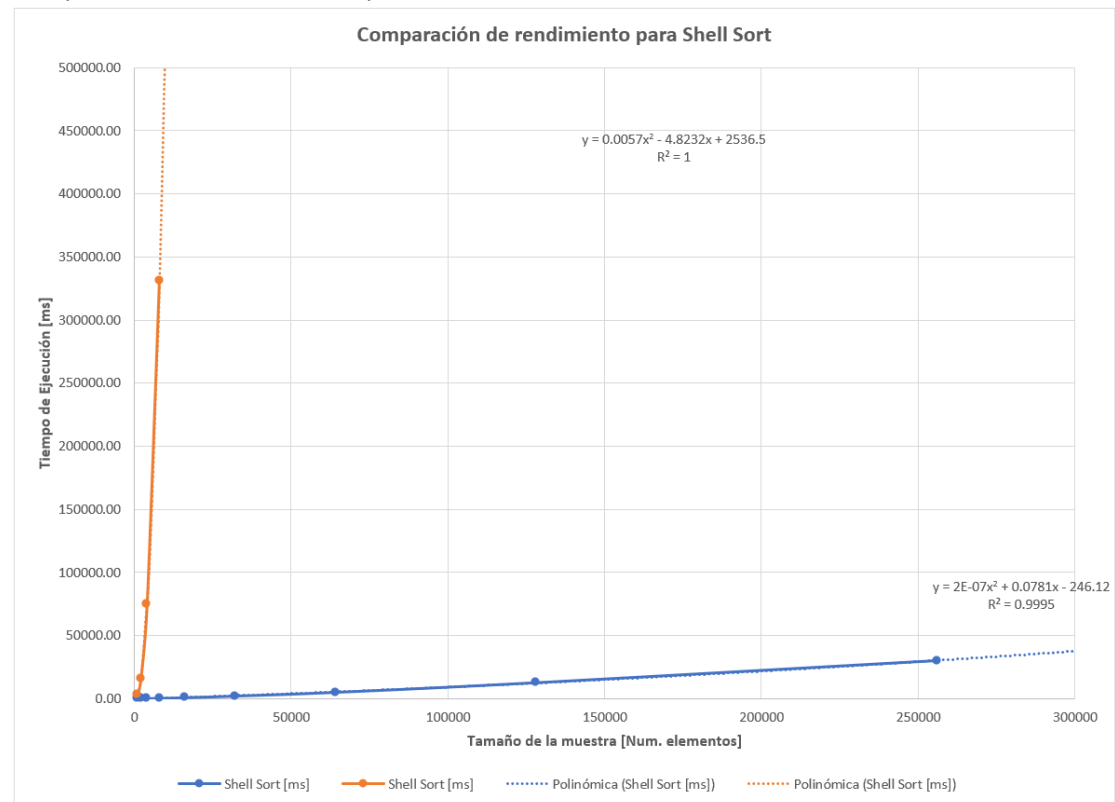
- Comparación de rendimiento para Insertion Sort.



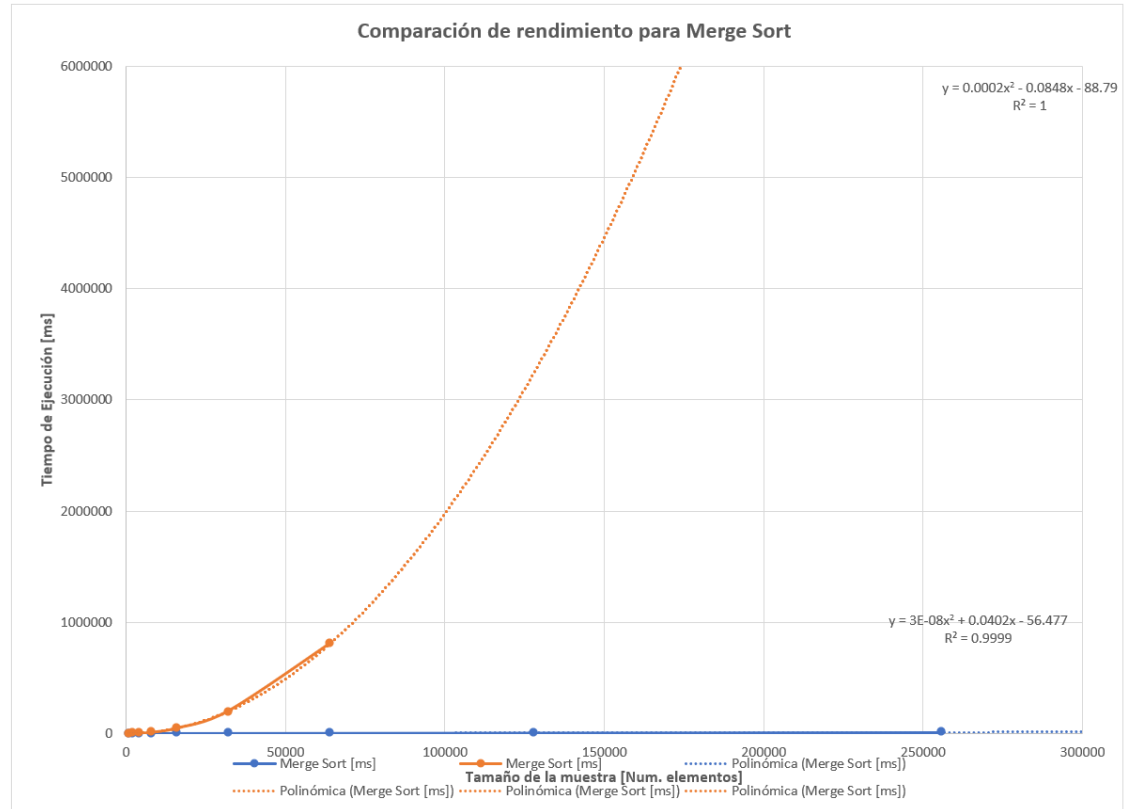
- Comparación de rendimiento para Selection Sort.



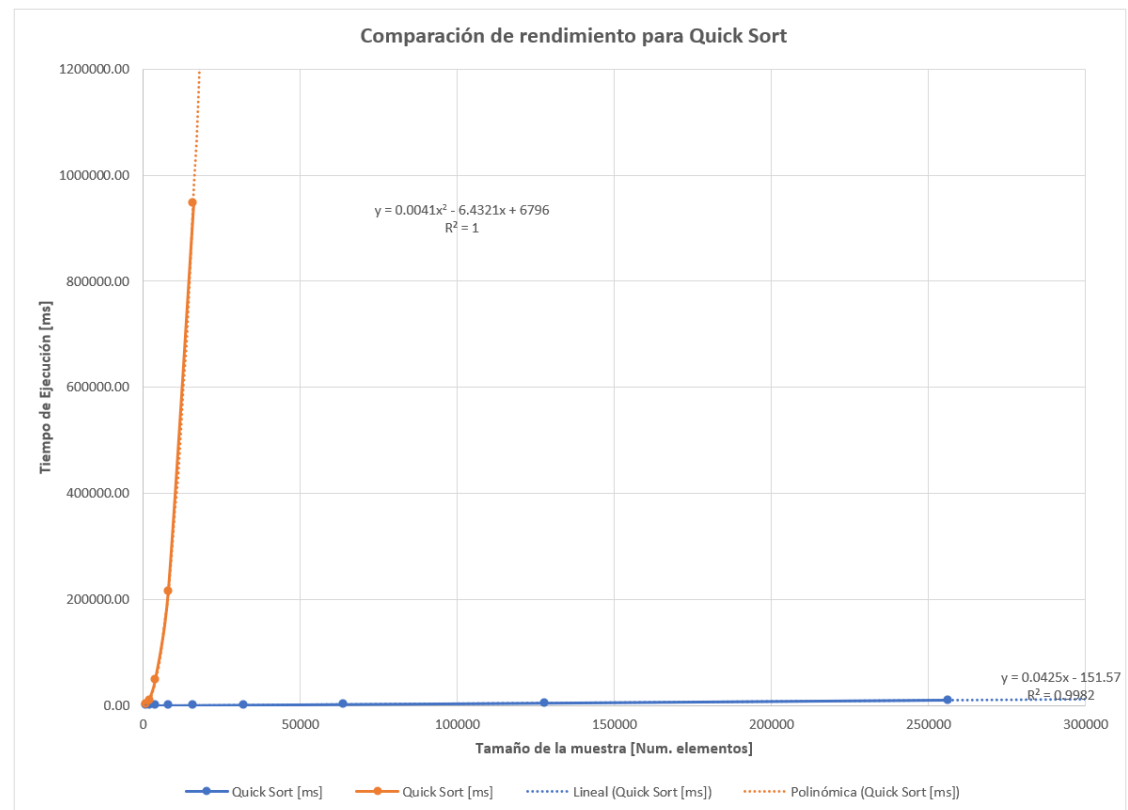
- Comparación de rendimiento para Shell Sort.



- Comparación de rendimiento para MergeSort.



- Comparación de rendimiento para QuickSort.



Maquina 2

Resultados

Tamaño de la muestra (ARRAYLIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
1000	468.75	515.625	31.25	15.625	31.25
2000	2125.0	2156.25	62.5	46.875	62.5
4000	8546.875	10421.875	156.25	93.75	109.375
8000	36359.375	40156.25	328.125	187.5	203.125
16000	147062.5	160281.25	765.625	687.5	484.375
32000	612453.125	667046.875	2046.875	906.25	1109.375
64000			4437.5	2218.75	2125.0
128000			11593.75	4593.75	4937.5
256000			26796.875	10640.625	9687.5
512000					

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

Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
1000	91125.0	37125.0	1734.375	2046.875	187.5
2000	832171.875	741312.5	9984.375	6687.5	890.625
4000			57750.0	32515.625	3390.625
8000			218609.375	141828.125	14265.625
16000				689578.125	53109.375
32000					209593.75
64000					
128000					
256000					
512000					

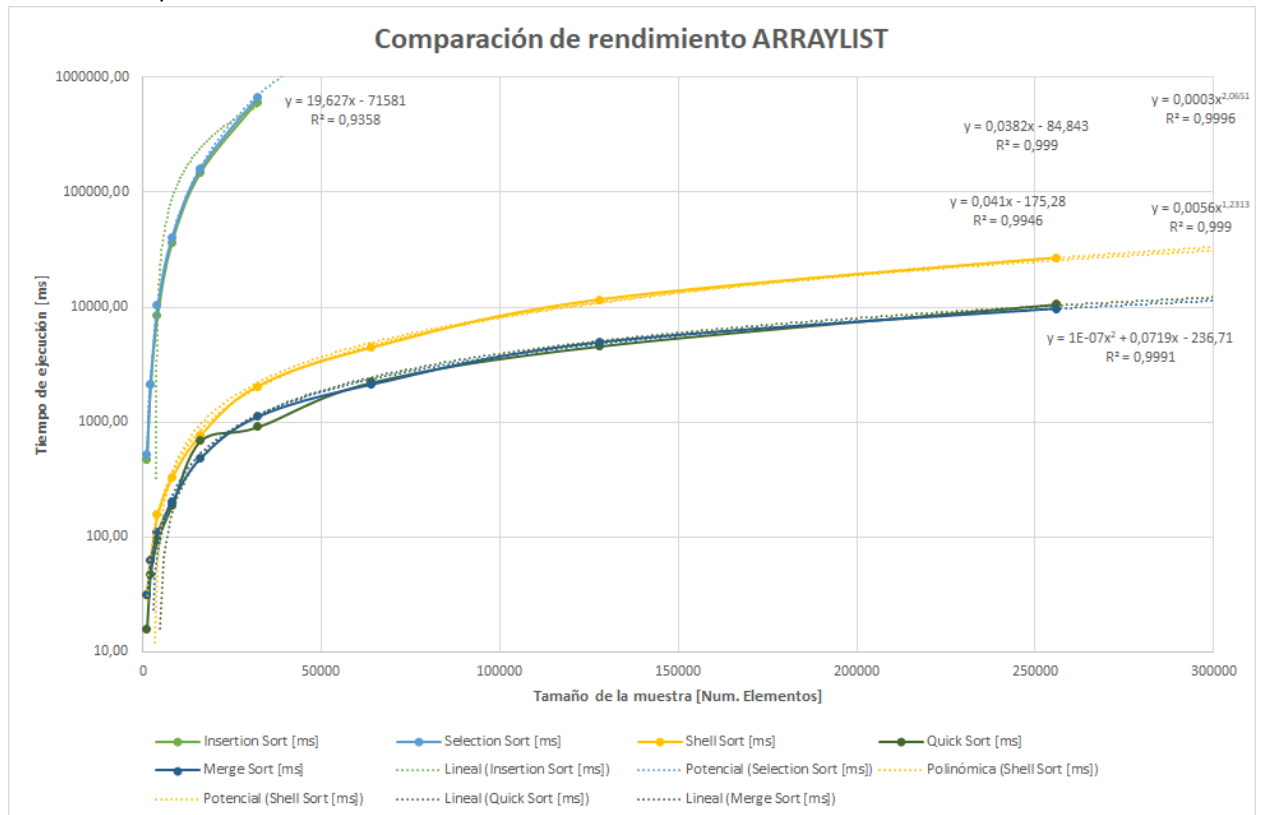
Tabla 6. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	x	
Quick sort	x	

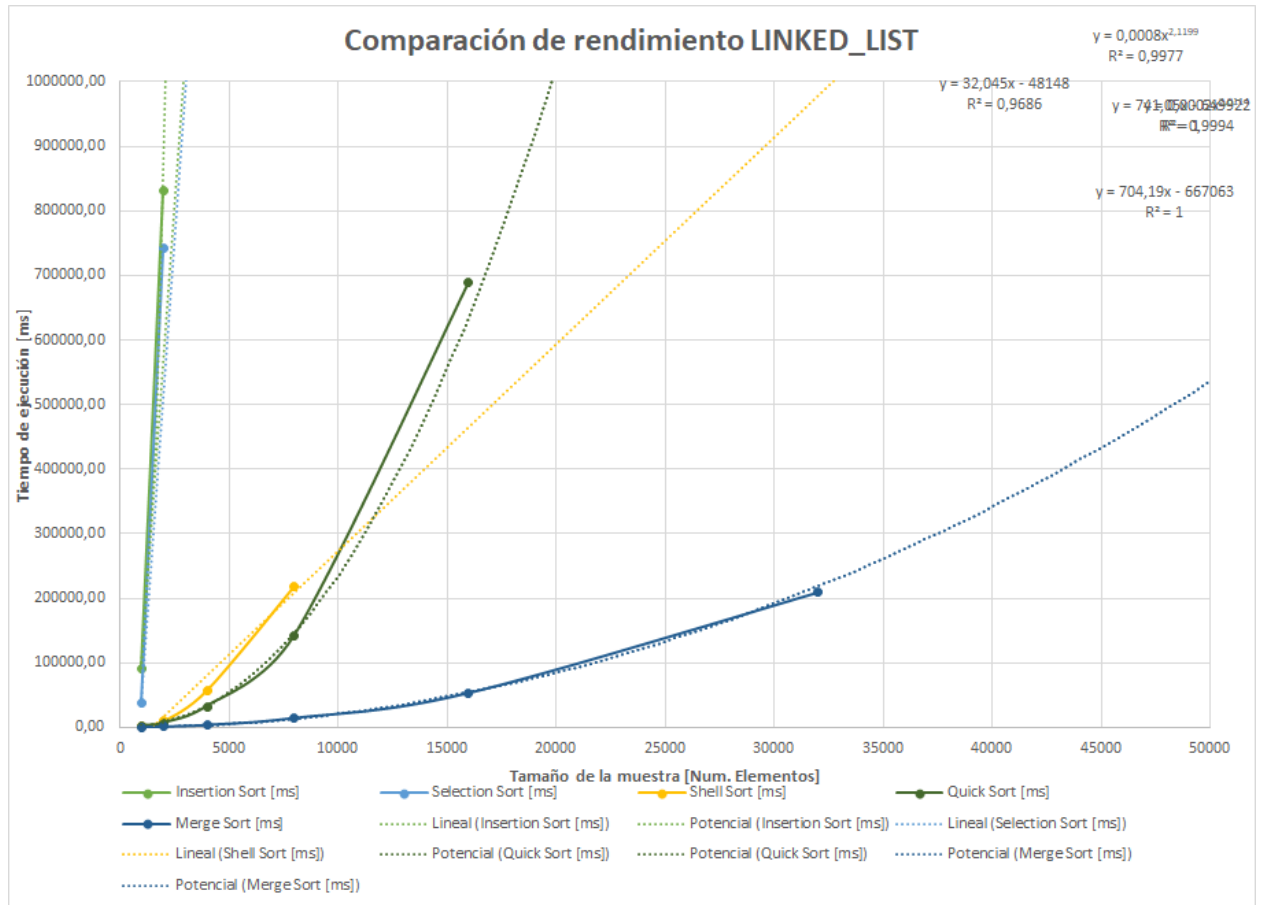
Tabla 7. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Graficas

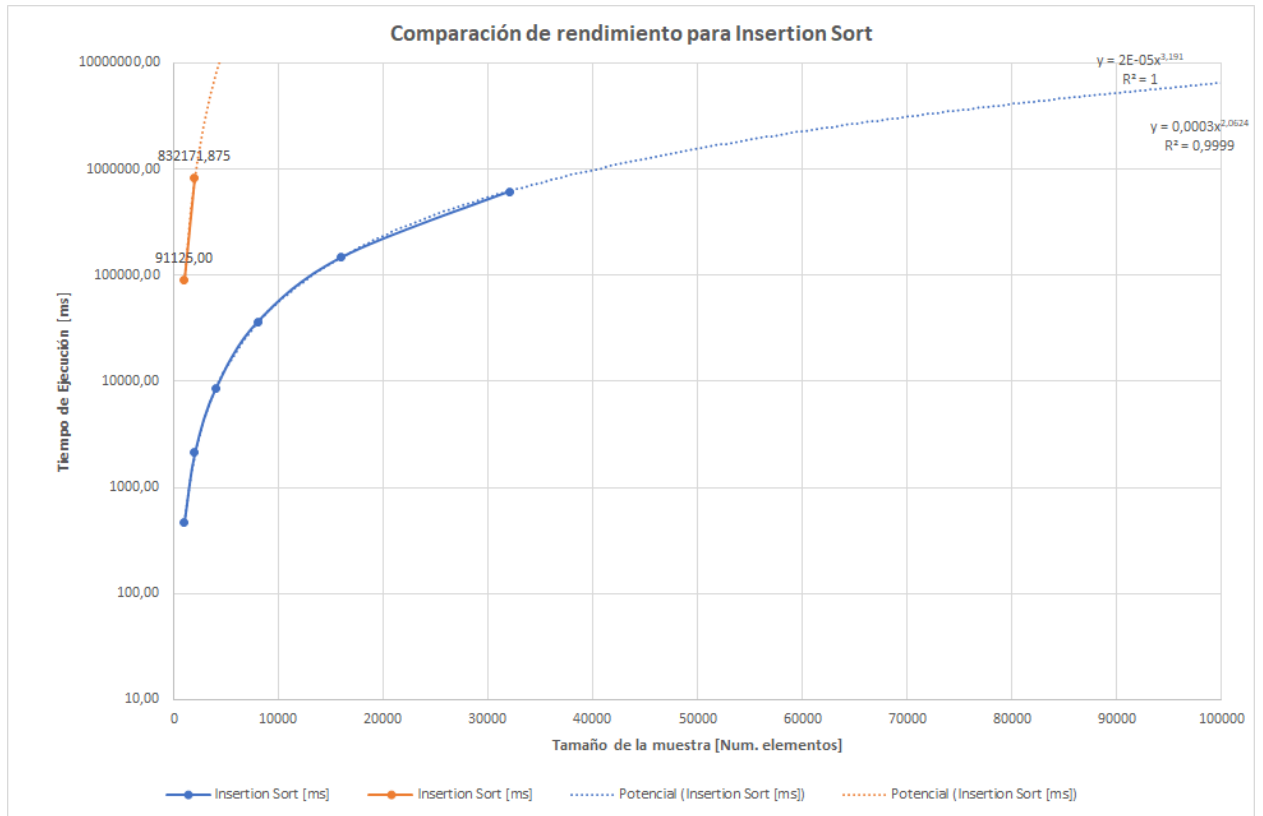
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2**.
 - Comparación de rendimiento ARRAYLIST.



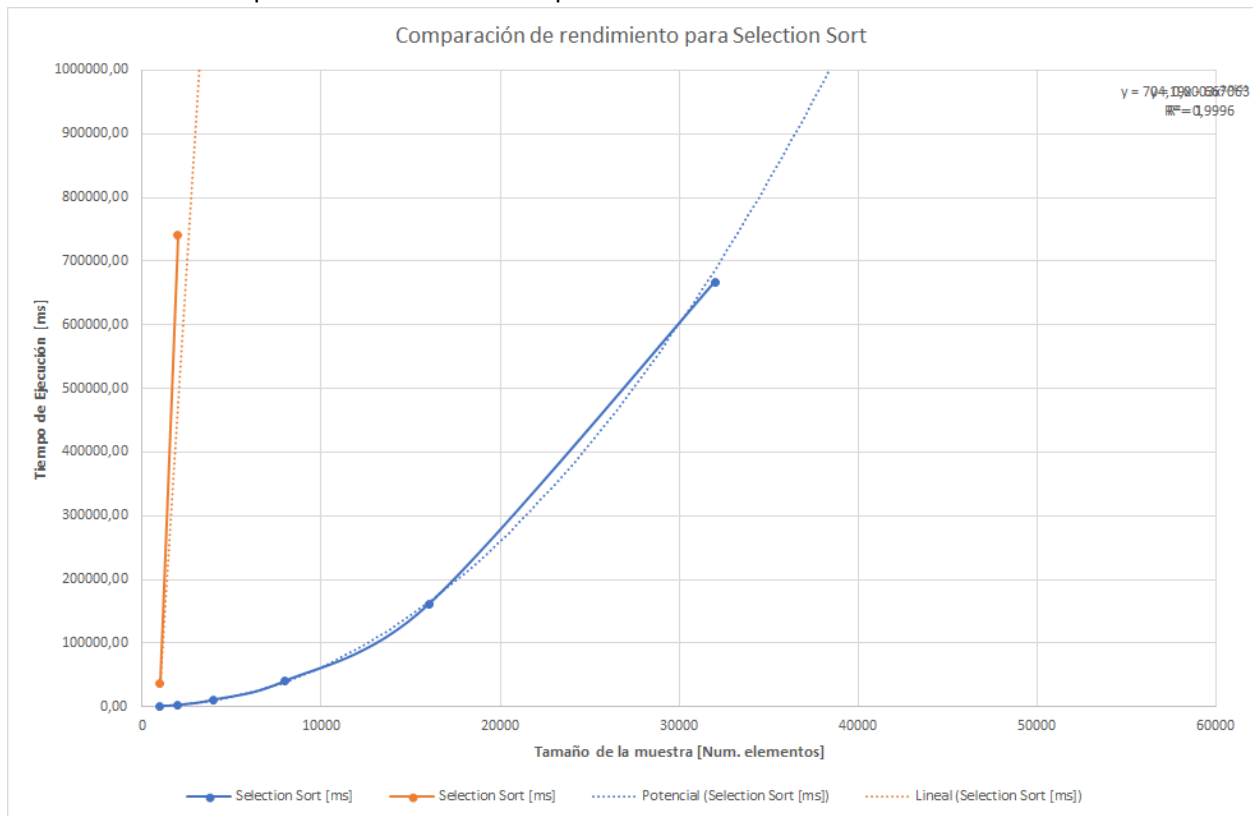
- Comparación de rendimiento LINKED_LIST.



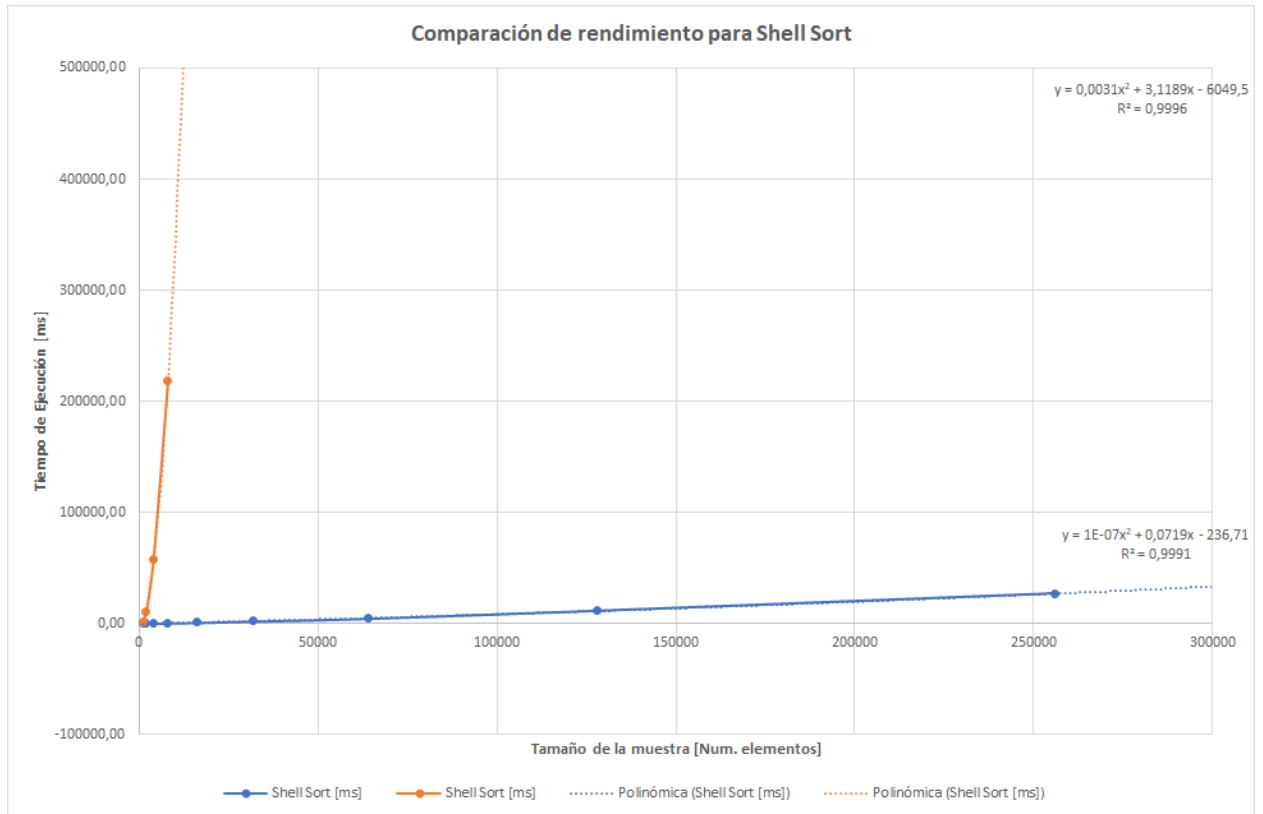
- Comparación de rendimiento para Insertion Sort.



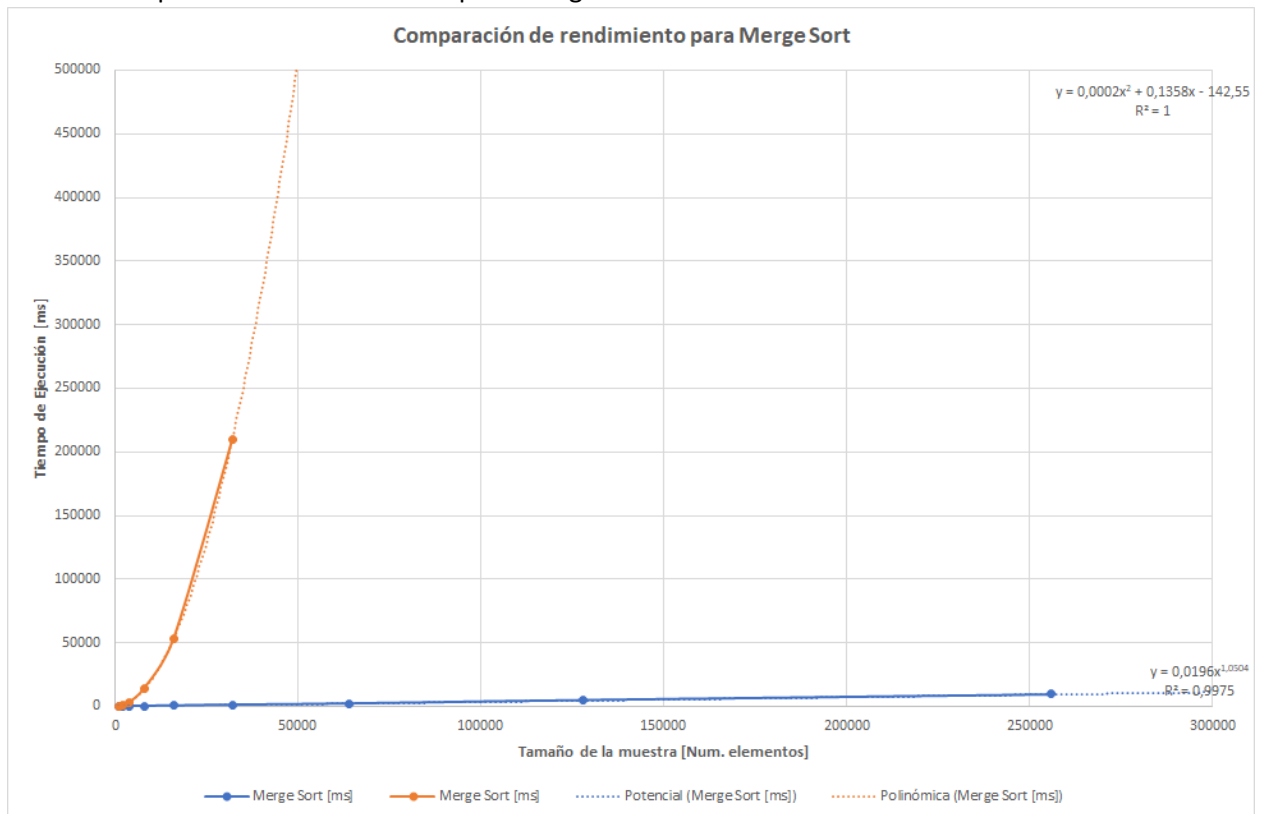
- Comparación de rendimiento para Selection Sort.



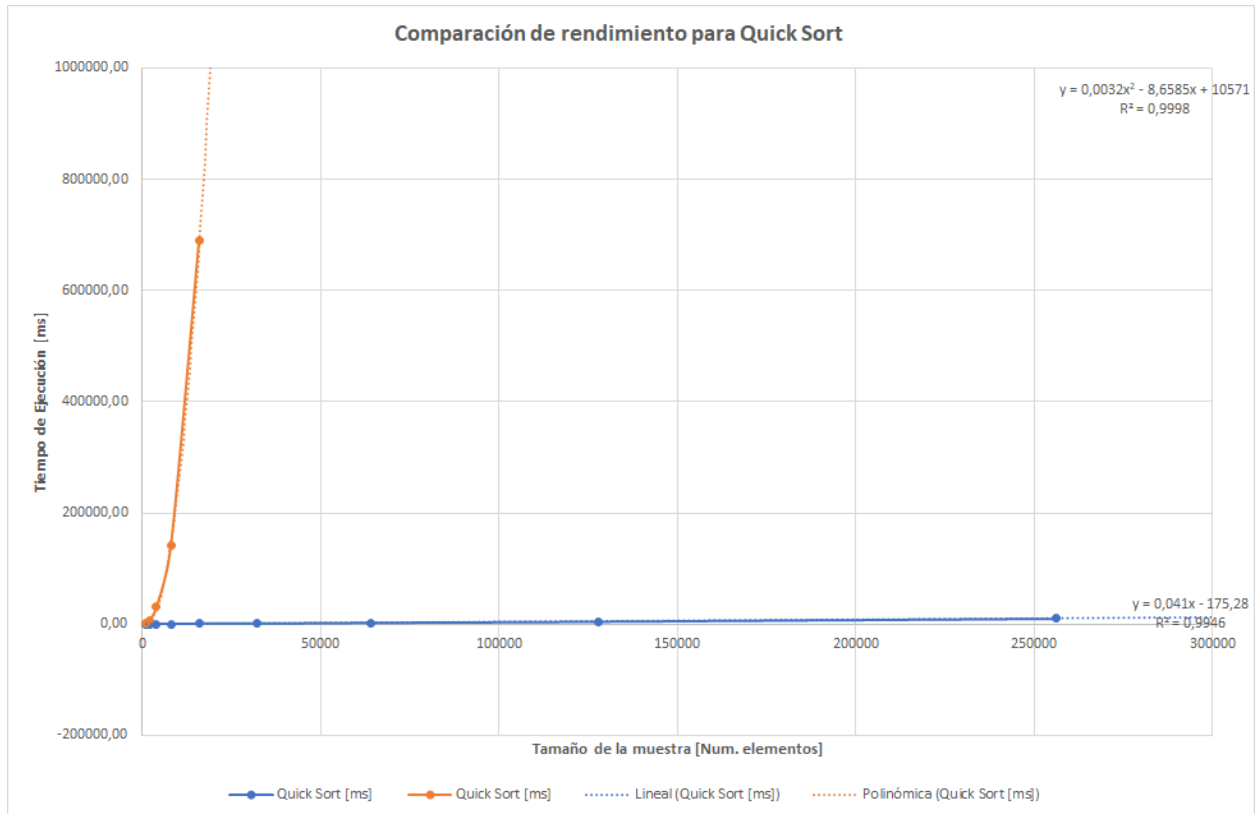
- Comparación de rendimiento para Shell Sort.



- Comparación de rendimiento para MergeSort.



- Comparación de rendimiento para QuickSort.



Preguntas de análisis

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

En efecto, Selection Sort e Insertion Sort son los algoritmos que más presentan los ordenes de complejidad mas altos, por lo tanto, son los algoritmos de ordenamiento mas lentos, lo cual se puede evidenciar en las gráficas de ARRAY_LIST y SINGLE_LINKED.

2) ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?

Si, esto se puede deber a los programas que se tenían abiertos en ese momento, velocidades de lectura y escritura de SSD y algunos otros factores aleatorios.

3) De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?

Podrían deberse a diferencias en el procesador, cantidad de memoria RAM utilizable. La unidad de memoria y el rendimiento del computador que pude reducirse a lo largo del día debido a temas de calentamiento o consumo energético.

4) ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

sería más efectivo usar ARRAY_LIST, ya que, si se trabajan con grandes volúmenes de datos, los algoritmos de ordenamiento serán mas efectivos.

- 5) Para el caso analizado de ordenamiento de los videos, teniendo en cuenta los resultados de tiempo reportados por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los algoritmos de ordenamiento (de mayor eficiencia a menor eficiencia en tiempo) para ordenar la mayor cantidad de videos.

ARRAY_LIST	SINGLE_LINKED
Quick sort (más eficiente)	Merge Sort (más eficiente)
Merge Sort	Quick sort
Shell sort	Shell sort
Insertion sort	Insertion sort
Selection sort (menos eficiente)	Selection sort (menos eficiente)