OBSERVACIONES DE LA PRACTICA

Estudiante 1: 202211498
Estudiante 2: 202213709
Estudiante 3: 202212100

	Máquina 1	Maquina 2	Máquina 3
Procesadores	**AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx 2.60 GHz**	**11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz**	**Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz**
Memoria RAM (GB)	**12,0 GB (9,92 GB usable)**	**16,0 GB DDR4**	**12 GB**
Sistema Operativo	**Windows 10 pro. Sistema operativo de 64 bits, procesador basado en x64**	**Windows 11 Home Single Language. 64-bit operating system, x64-based processor**	**Windows 10. Sistema operativo de 64 bits, procesador x64**

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

Maquina 1

Resultados

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]
5.00%	1148	8.82	1345.6	49.87	105.6	37.6
20.00%	4598	87.14	27541.1	303.59	1107.97	136.4
30.00%	6898	129.81	54403.4	845.11		213.5
50.00%	11498	251.12	162230.6	765.33		339
100.00%	22998	403.69	735696.78	1591.76		761.1

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]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1148	493.55	69509.6	1389.9	13374.24	284.312
20.00%	4598	9733.7		32905.5	No funciono mas	4384.6
30.00%	6898	22465.09		80293.9		11777.19
50.00%	11498	21308.04		268459.5		27787.53
100.00%	22998	251282.78				126290.69

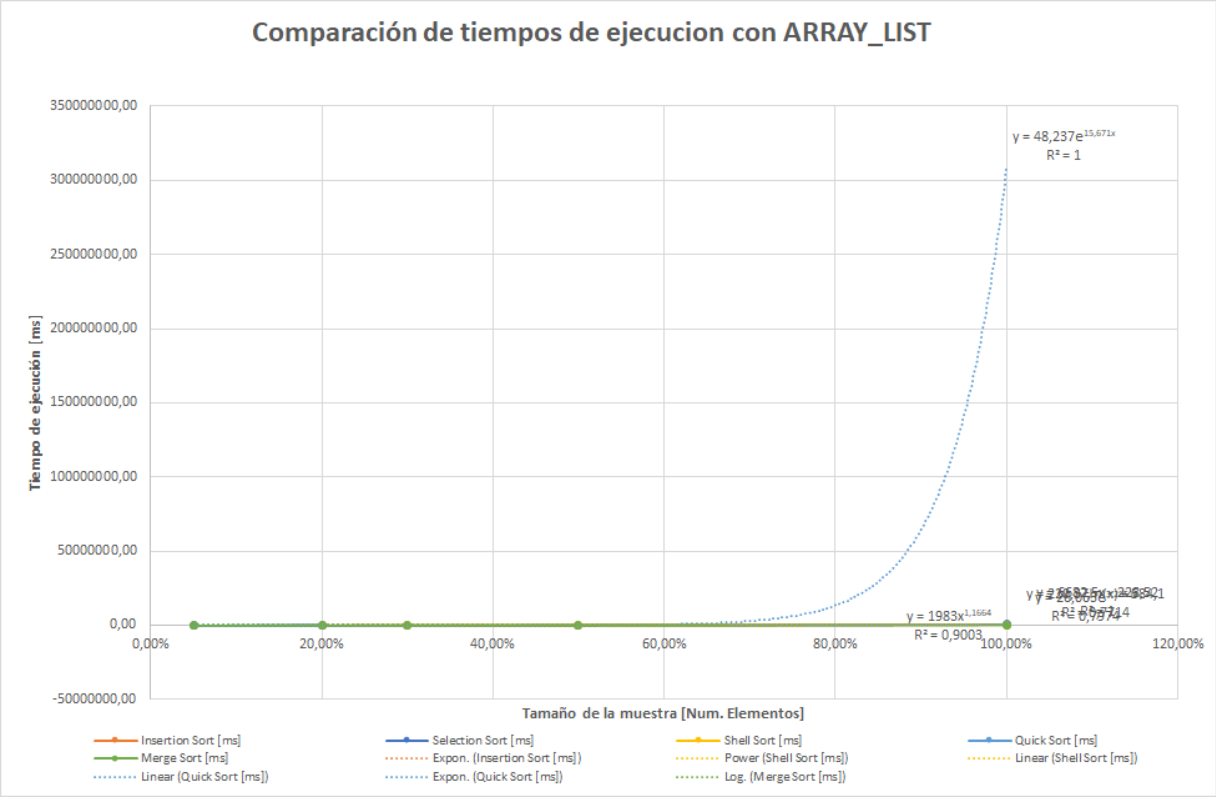
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

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

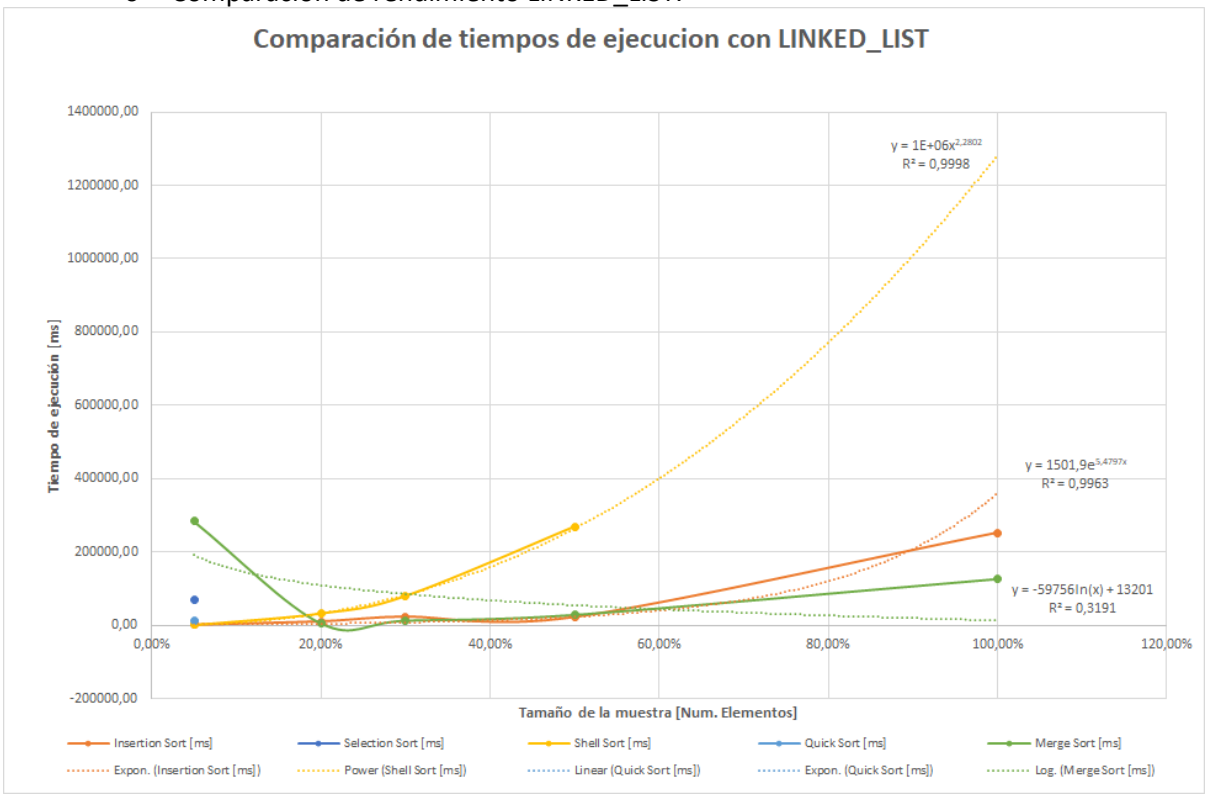
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Graficas

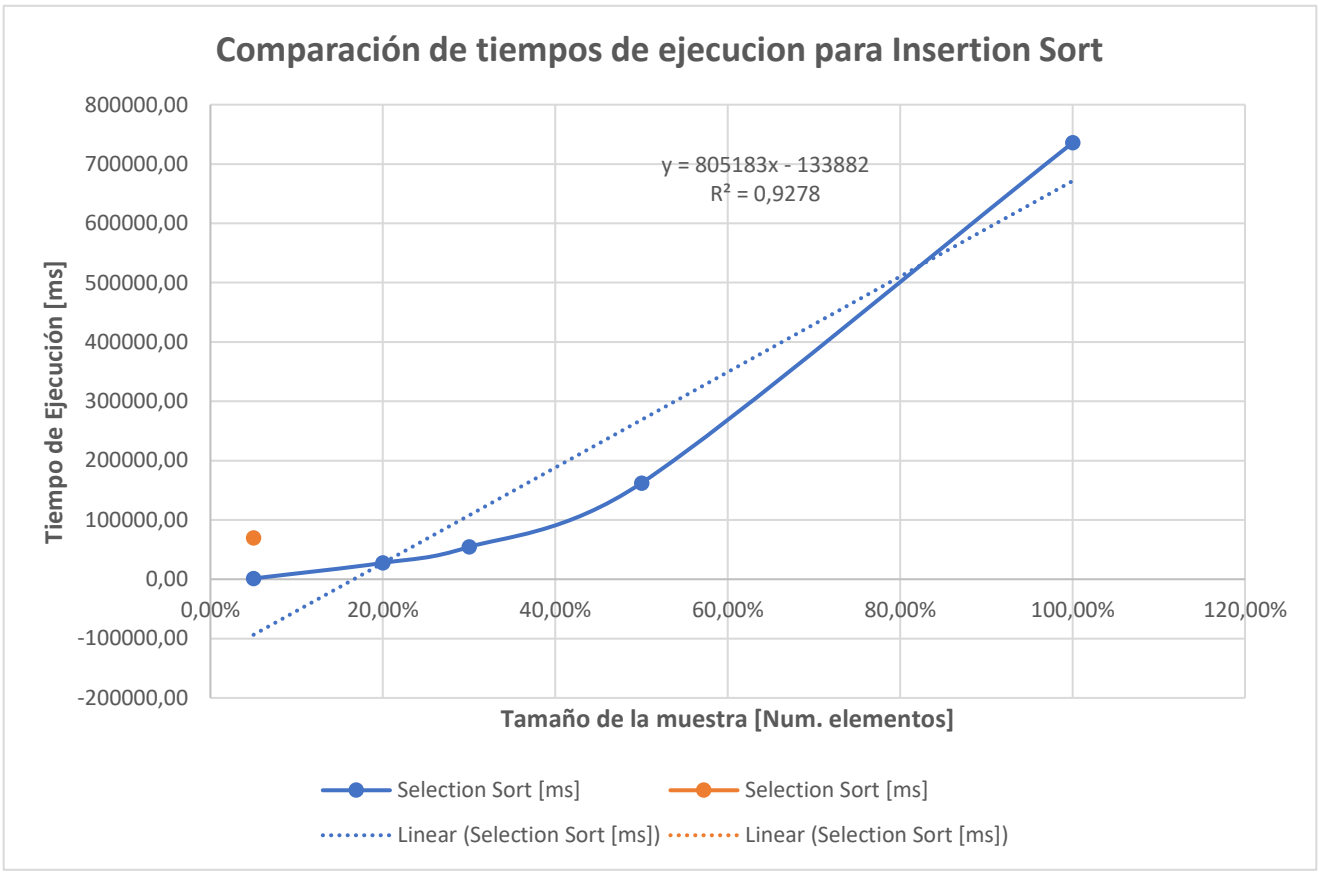
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
    - Comparación de rendimiento ARRAYLIST.



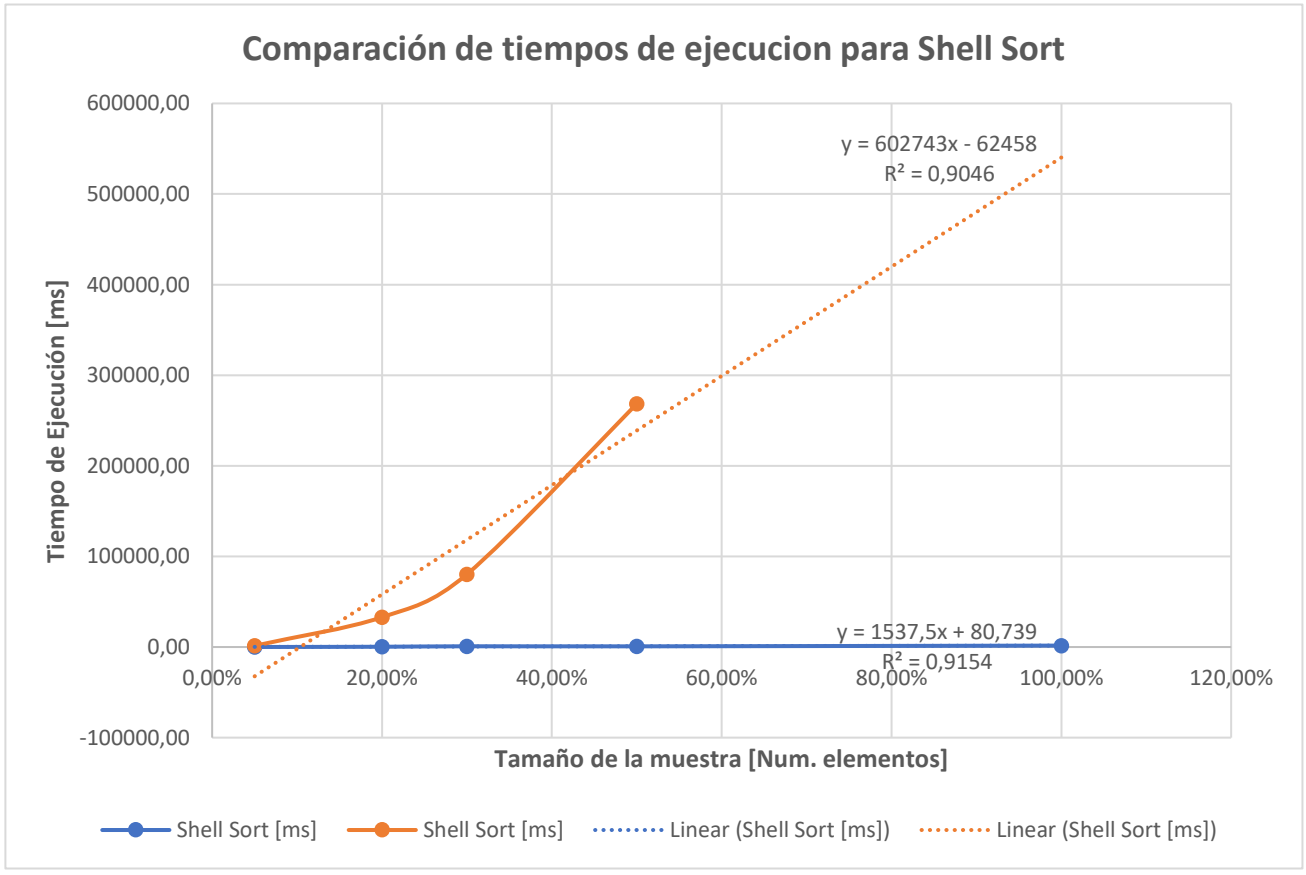
    - Comparación de rendimiento LINKED_LIST.



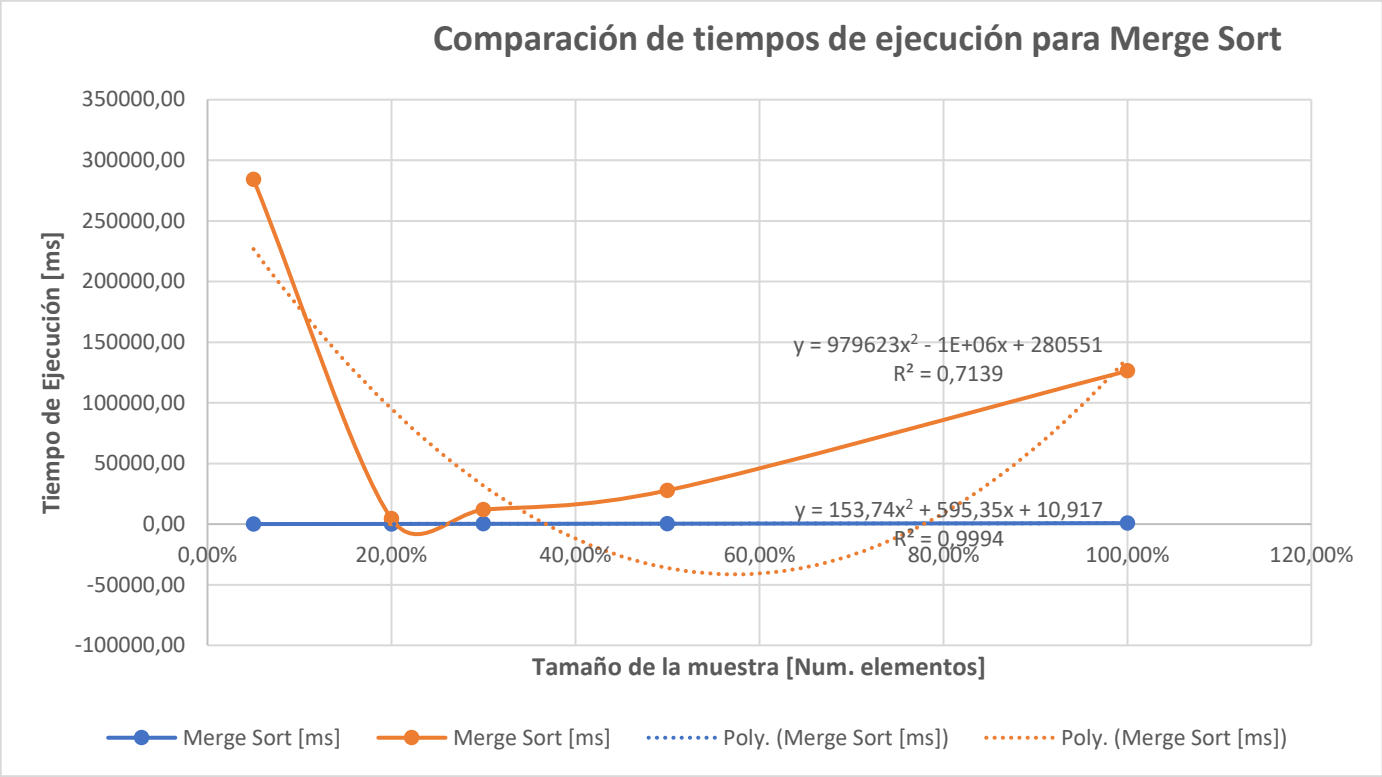
- Comparación de rendimiento para Selection Sort.



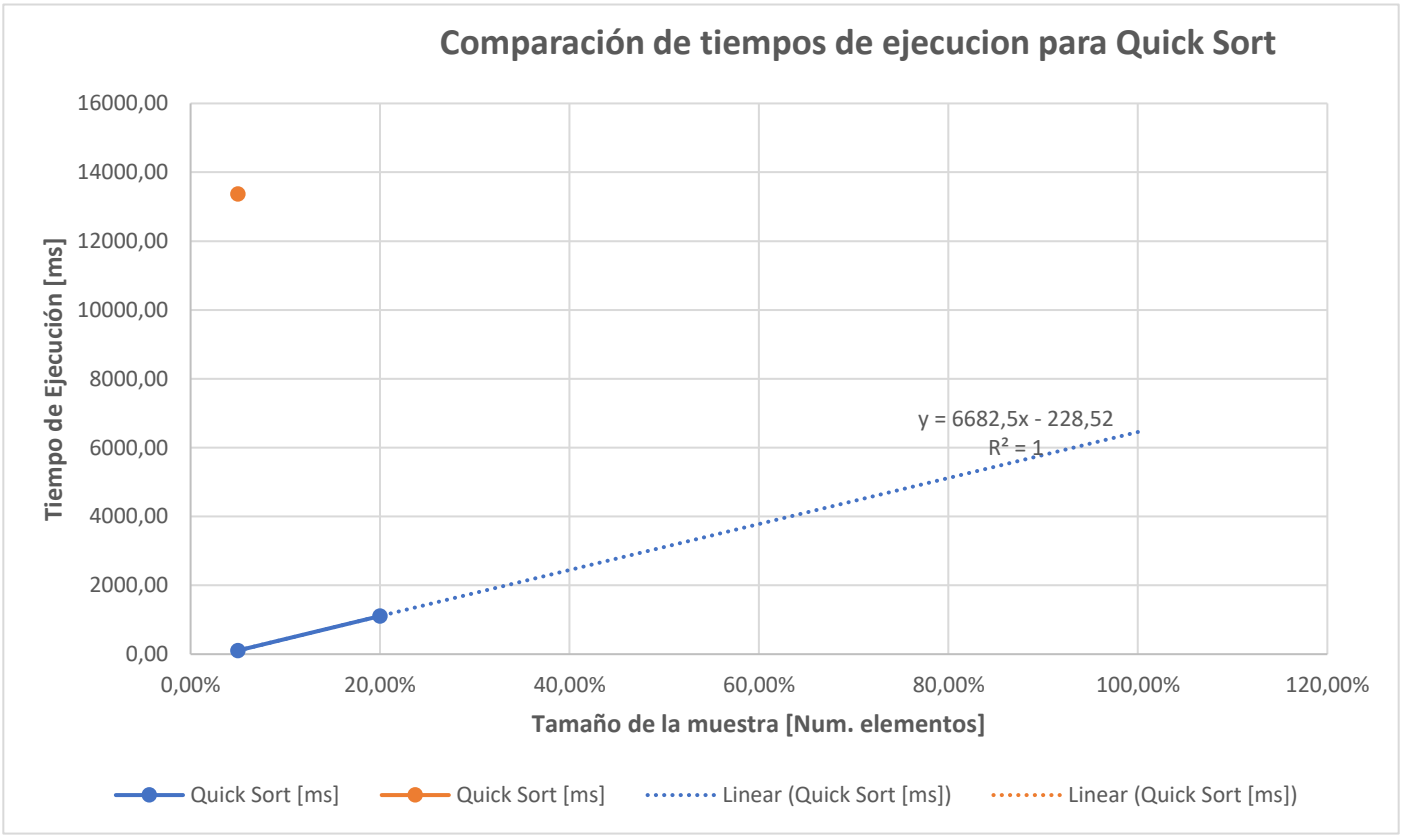
- Comparación de rendimiento para Shell Sort.



- Comparación de rendimiento para MergeSort.



- Comparación de rendimiento para QuickSort.



Maquina 2

Resultados

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]
5.00%	1148	3.251	590.76	57.71	103.15	21.86
20.00%	4589	23.20	12747.29	98.09	755.56	113.10
30.00%	6898	30.46	30266.87	123.51	1451.83	181.97
50.00%	11498	54.57	86062.43	220.69		245.10
100.00%	22998	97.92	401559.44	522.73		491.63

Tabla 5. 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]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1148	201.15	40664.01	956.77	7671.48	221.06
20.00%	4589	3722.38		26935.16	427346.38	2802.62
30.00%	6898	9027.23		50991.24	1977997.51	6138.96
50.00%	11498	24170.64		151083.12		17411.38
100.00%	22998	102265.94		687119.77		70183.89

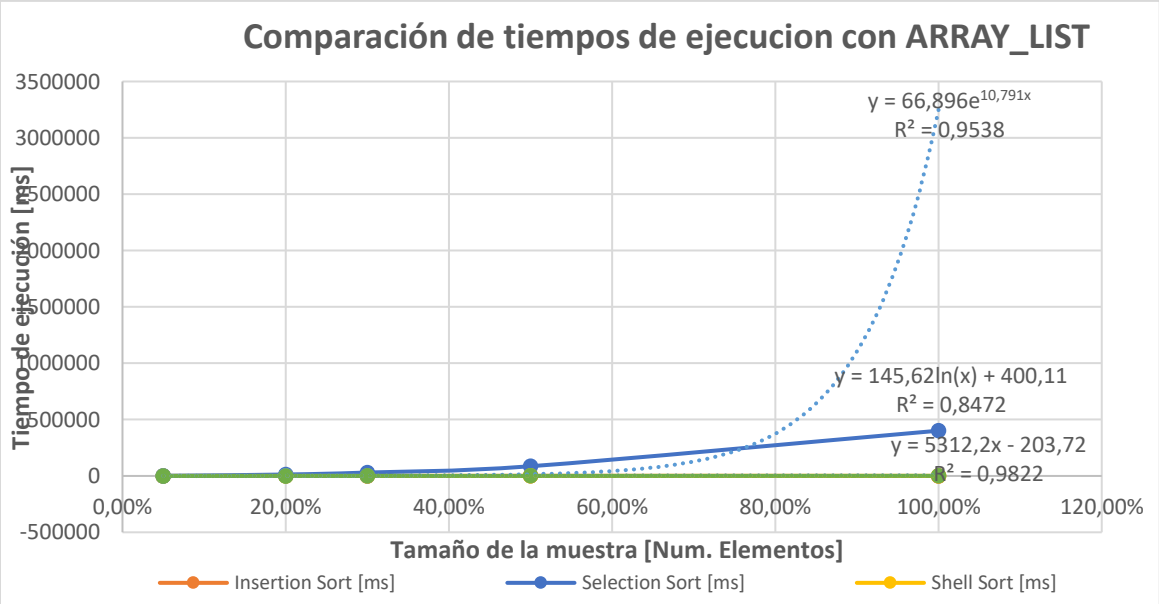
Tabla 6. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

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

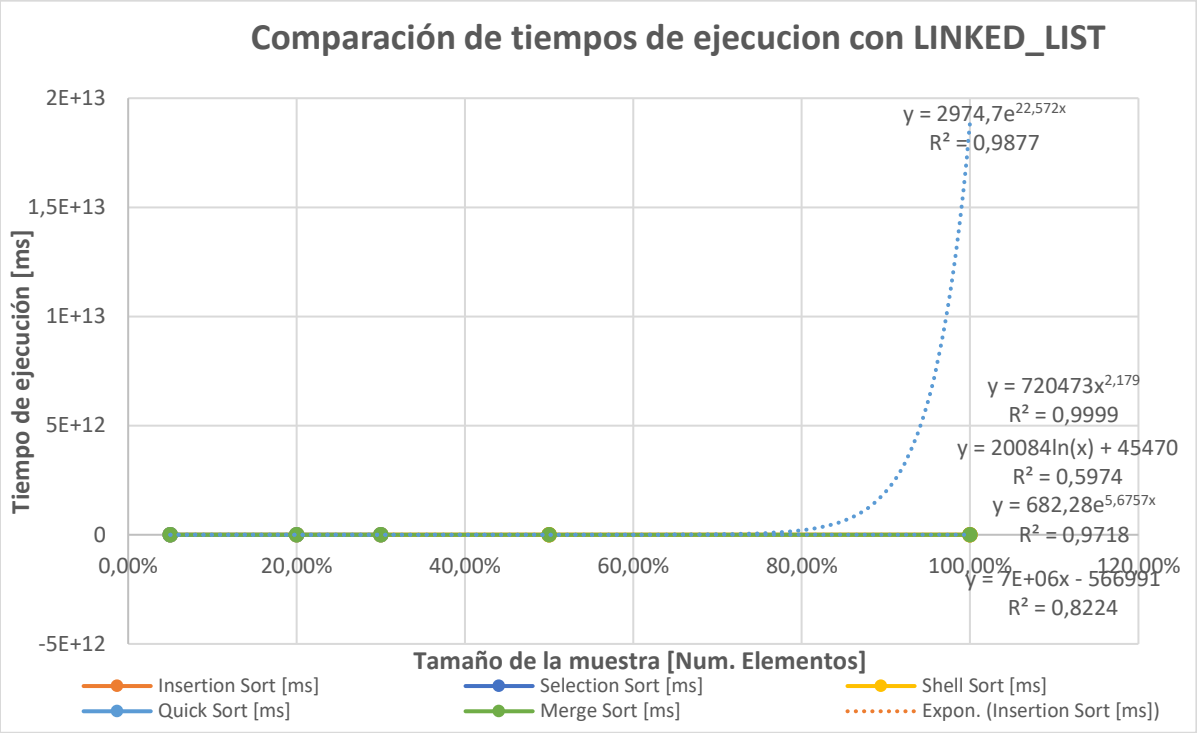
Tabla 7. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Graficas

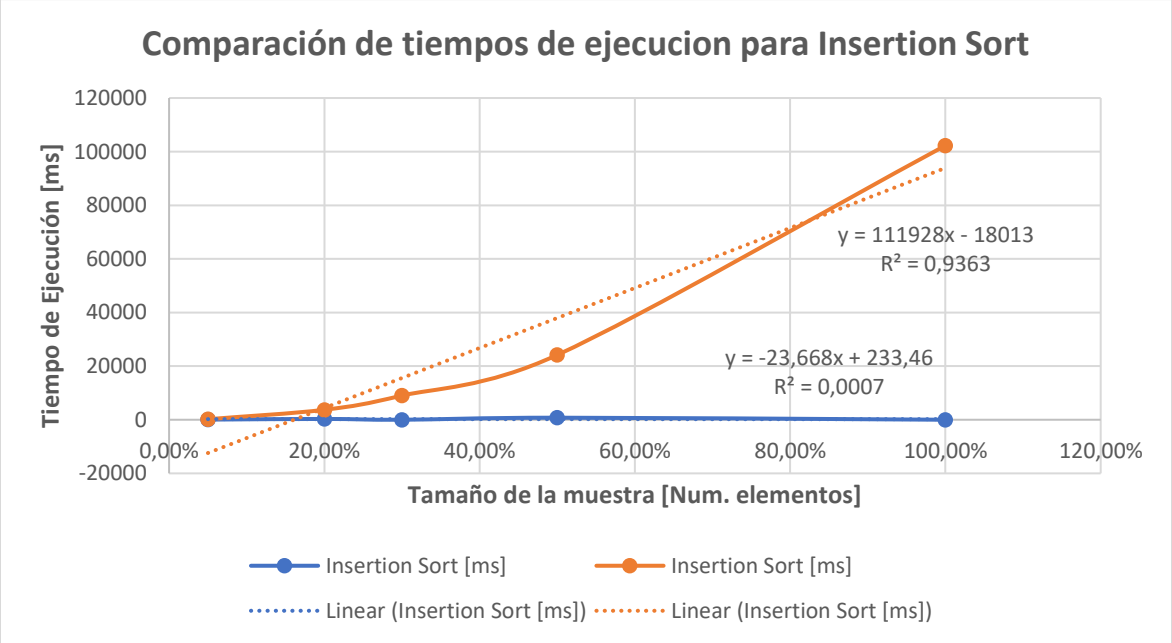
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  - Comparación de rendimiento ARRAYLIST.



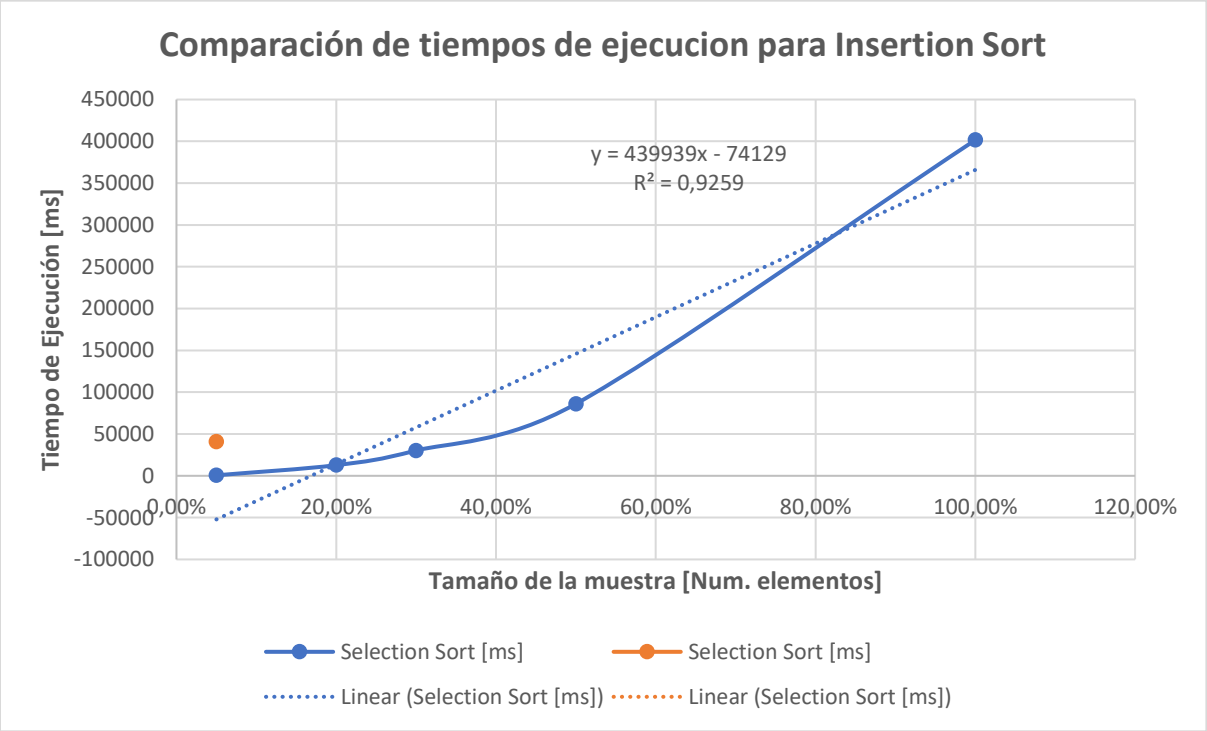
- Comparación de rendimiento LINKED_LIST.



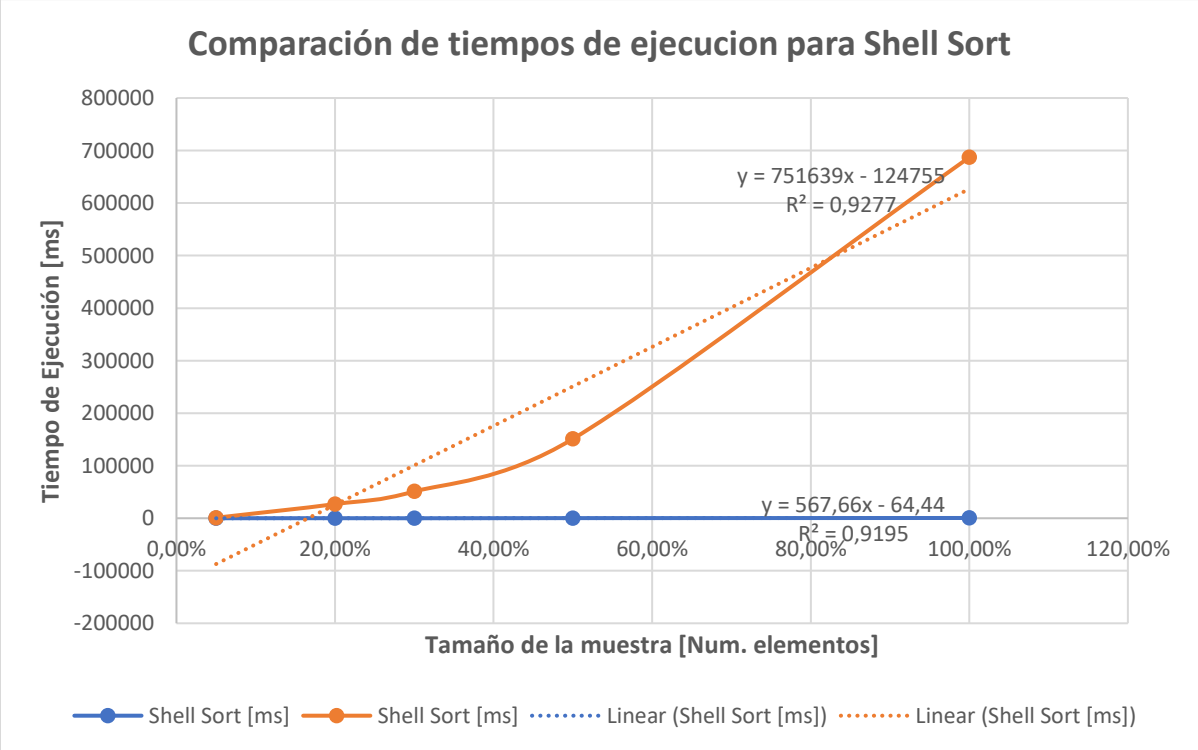
- Comparación de rendimiento para Insertion Sort.



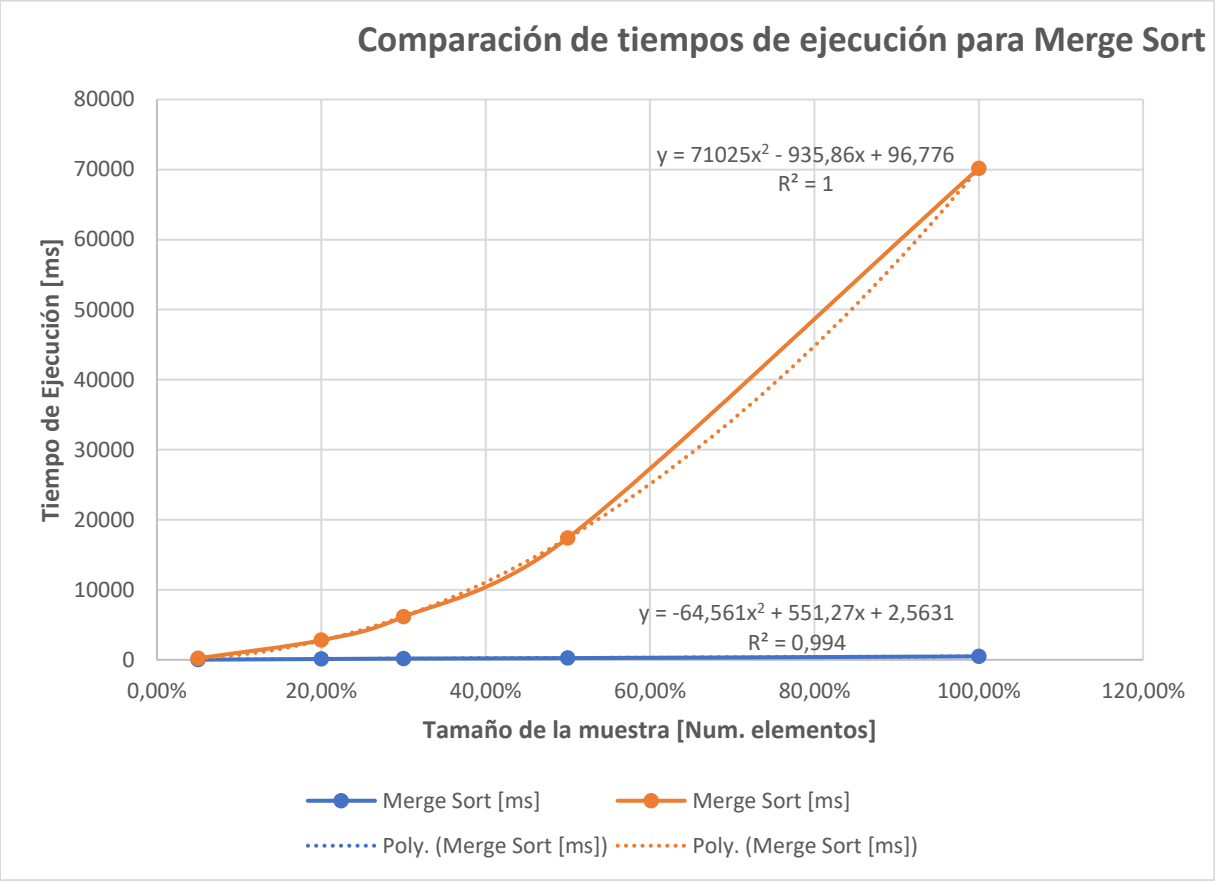
- Comparación de rendimiento para Selection Sort.



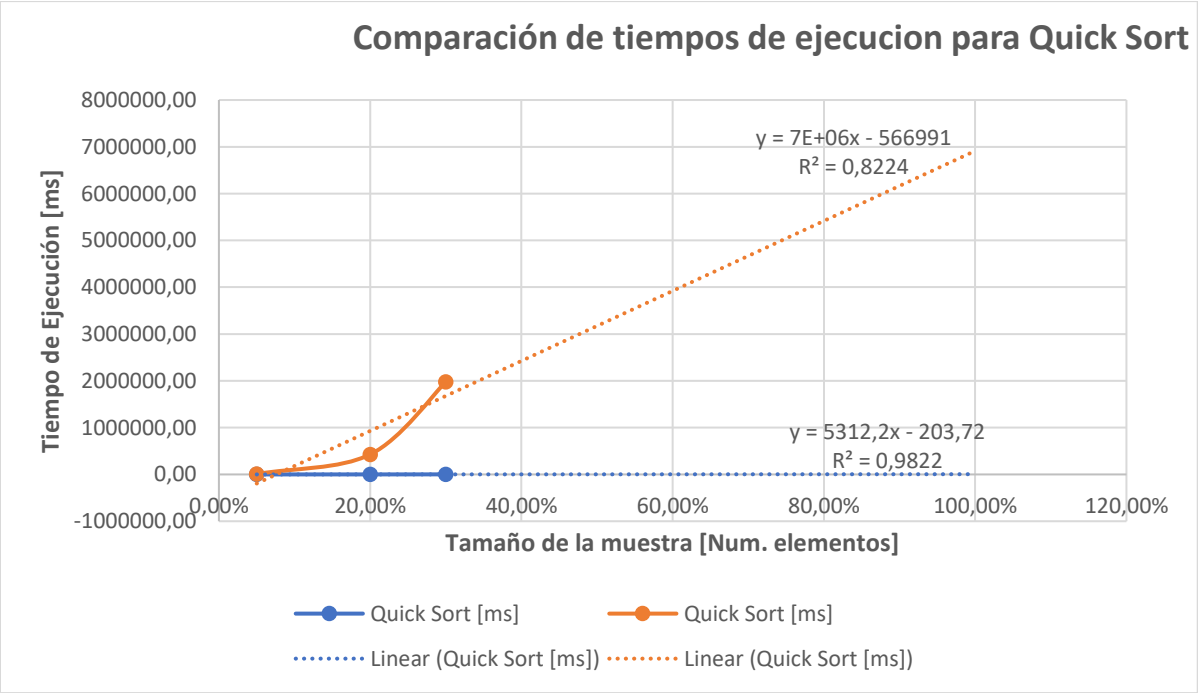
- Comparación de rendimiento para Shell Sort.



- Comparación de rendimiento para MergeSort.



- Comparación de rendimiento para QuickSort.



Maquina 3

Resultados

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]
5.00%	1148	3.365	699.211	21.219	61.788	20.574
20.00%	4598	21.076	11665.773	97.479	648.333	73.518
30.00%	6898	31.627	26795.412	146.526	1373.148	109.624
50.00%	11498	43.001	87030.601	257.697		190.513
100.00%	22998	77.297	322363.2328	543.485		458.836

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]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1148	753.360	44353.284	1422.937	11616.802	174.623
20.00%	4598	7991.697		27578.399	592658.684	2706.628
30.00%	6898	17125.406		70187.035	1977997.52	6076.724
50.00%	11498	33499.753		208712.025		16797.290
100.00%	22998	141160.655				68604.434

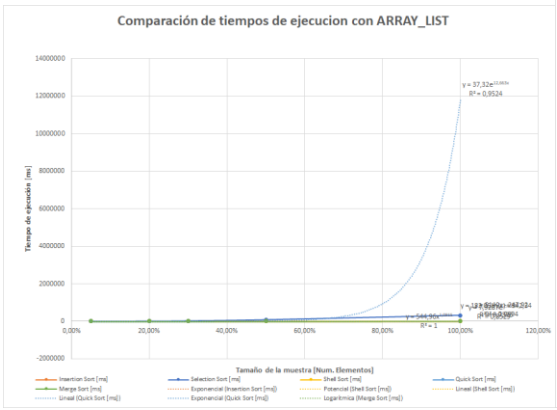
Tabla 9. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

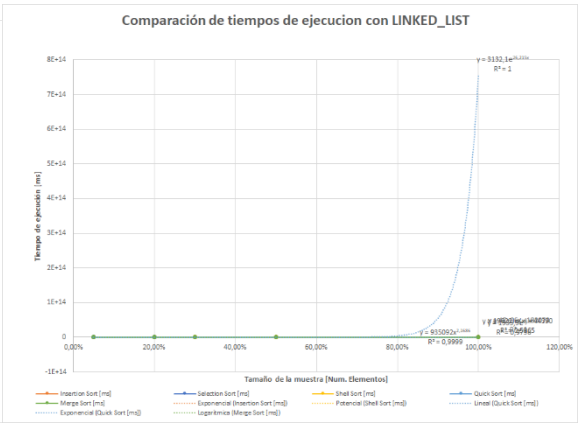
Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	✓	✓
Quick sort		

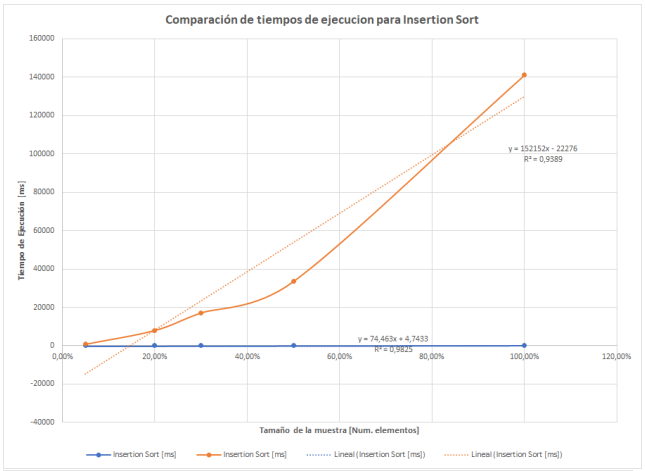
Tabla 10. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

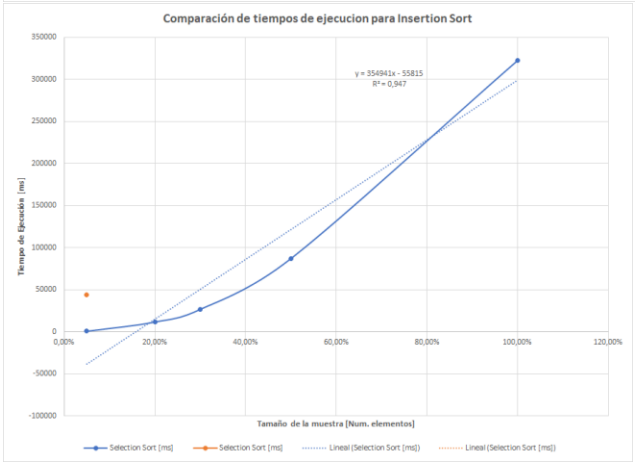
Graficas

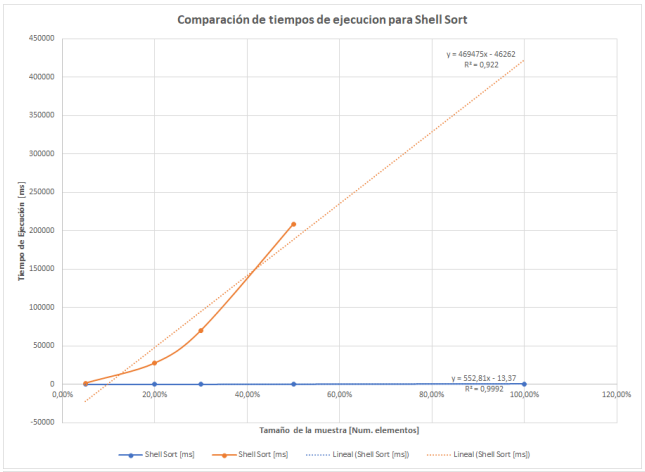
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 3.**
  - Comparación de rendimiento ARRAYLIST.
  - Comparación de rendimiento LINKED_LIST.
  - Comparación de rendimiento para Insertion Sort.
  - Comparación de rendimiento para Selection Sort.
  - Comparación de rendimiento para Shell Sort.
  - Comparación de rendimiento para MergeSort.
  - Comparación de rendimiento para QuickSort.

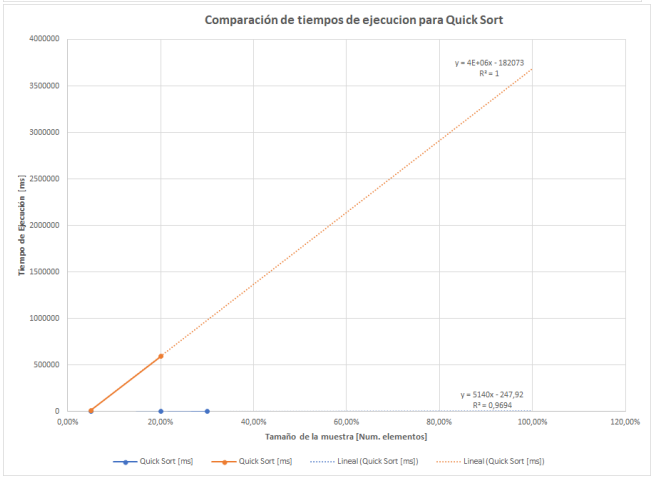

Comparación de tiempos de ejecucion con ARRAY_LIST
Tiempo de ejecución [ms]
Tamaño de la muestra [Num. Elementos]
Insertion Sort [ms]
Selection Sort [ms]
Shell Sort [ms]
Quick Sort [ms]
Merge Sort [ms]
Exponencial (Insertion Sort [ms])
Potencial (Shell Sort [ms])
Lineal (Shell Sort [ms])
Lineal (Quick Sort [ms])
Exponencial (Quick Sort [ms])
Logarítmica (Merge Sort [ms])

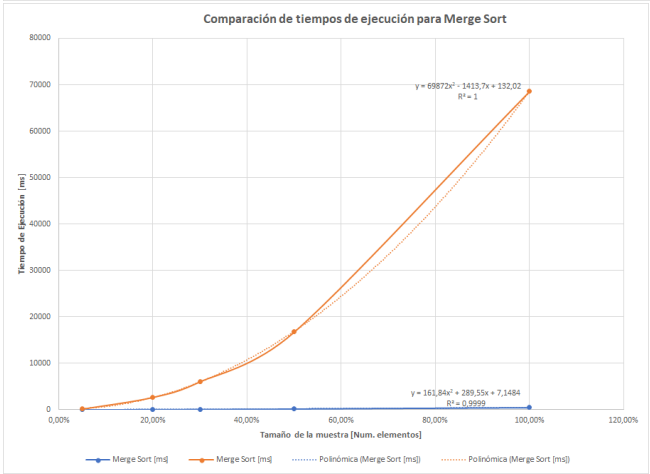

Comparación de tiempos de ejecucion con LINKED_LIST
Tiempo de ejecución [ms]
Tamaño de la muestra [Num. Elementos]
Insertion Sort [ms]
Selection Sort [ms]
Shell Sort [ms]
Quick Sort [ms]
Merge Sort [ms]
Exponencial (Insertion Sort [ms])
Potencial (Shell Sort [ms])
Lineal (Quick Sort [ms])
Exponencial (Quick Sort [ms])
Logarítmica (Merge Sort [ms])


Comparación de tiempos de ejecucion para Insertion Sort
y = 152152x - 22276
R² = 0,9389
y = 74,463x + 4,7433
R² = 0,9825
Tiempo de Ejecución [ms]
Tamaño de la muestra [Num. elementos]
Insertion Sort [ms]
Insertion Sort [ms]
Lineal (Insertion Sort [ms])
Lineal (Insertion Sort [ms])


Comparación de tiempos de ejecucion para Insertion Sort
y = 354941x - 55815
R² = 0,947
Tiempo de Ejecución [ms]
Tamaño de la muestra [Num. elementos]
Selection Sort [ms]
Selection Sort [ms]
Lineal (Selection Sort [ms])
Lineal (Selection Sort [ms])







Preguntas de análisis

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

Si, en cuanto a insertion sort las tres gráficas cumplieron con el mejor caso esperado O(n), así pues, insertion sort con ARRAY_LIST es el mejor algoritmo en tiempos de ejecución, estabilidad y con la cualidad in-Place.

Ahora bien, para todos los tipos de ordenamientos corroboramos que los tiempos con LINKED_LIST tienen una gran diferencia con ARRAY_LIST, siendo los tiempos de ejecución de los últimos, menores que los primeros.

Para evidenciar la estabilidad de Quick las tres máquinas no pudieron completar pruebas de 50.00% ni 100.00%. Con Merge las máquinas ejecutaron todos los porcentajes de muestra propuestos en tiempos relativamente cortos a comparación de Quick Sort.

Así pues, las pruebas realizadas nos permitieron realizar una evaluación de ordenamientos según su efectividad en tiempo y comportamiento, resultados que consignamos en la pregunta 5.

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

Si, evidenciamos que la máquina 3 tiene en general menores tiempos de ejecución (particularmente en array_list), a su vez también sus gráficas coinciden con el comportamiento esperado del tipo de ordenamiento. La máquina 2, también tiene un buen desempeño, y en algunos casos tiene menor tiempo de ejecución que la maquina 3, en particular en Linked_list. Por otra parte la máquina 1 para Quick sort, únicamente arrojó la primera muestra, las otras dos máquinas alcanzaron dos resultados más.

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

Las diferencias se deben a los procesadores que tiene las maquinas, así como la memoria RAM que tienen disponible. Por ejemplo, la maquina 2 tiene más memoria RAM disponible en comparación con las otras 2, lo que podría explicar los mejores tiempos de ejecución con Linked_list en las operaciones que más memoria requieren.

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

En el análisis completo con todos los sorts tanto iterativos como recursivos, el mejor es Insertion Sort con la estructura de datos ARRAY_LIST, pues consitentemente obtiene los menores resultados en tiempo de ejecución.

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

TIPO	TIEMPO MÍNIMO ALCANZADO
1. Insertion Sort – ARRAY_LIST	3,365
2. Merge sort – ARRAY_LIST	20,574
3. Shell sort – ARRAY_LIST	21,219
4. Quick sort – ARRAY_LIST	61,788
5. Merge Sort – SINGLE_LINKED	174,62
6. Insertion Sort – SINGLE_LINKED	201,15
7. Selection Sort – ARRAY_LIST	509,76
8. Shell Sort – SINGLE_LINKED	956,77
9. Quick sort – SINGLE_LINKED	7671,48
10. Selection Sort – SINGLE_LINKED	40664,01