

OBSERVACIONES DE LA PRÁCTICA

Manuel Bermúdez - 202121953

Jorge Bustamante -202210581

Jairo Garavito - 202111499

	Máquina 1	Máquina 2
Procesadores	Intel(R) Core(TM) i5-1035G4 CPU @ 1.10GHz 1.50 GHz	Ryzen 5 4600 h
Memoria RAM (GB)	4.00 GB	8.00 GB
Sistema Operativo	Windows 10 Pro - 64 bits	Windows 10 Pro - 64 bits

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

Máquina 1

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAYLIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1148	48.05	32,12	39,47	30,47	33,46
20.00%	4598	356,46	331,94	326,04	345,35	359,78
30.00%	6898	859,90	888,04	925,32	887,12	857.69
50.00%	11498	2741,01	2741,69	2720,98	2696,78	2694,25
100.00%	22998	13178,26	13264,60	13111,11	13302,02	13374,96

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

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKEDLIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1148	33,01	33,09	61,23	36,09	33,66
20.00%	4598	384,72	342,59	369,02	356,96	328,59
30.00%	6898	848.89	929.68	859.38	862.23	865.45
50.00%	11498	2854.92	2777.87	2776.44	2854.92	2802.08
100.00%	22998	13791.36	13968.26	13653.65	14012.27	13988.42

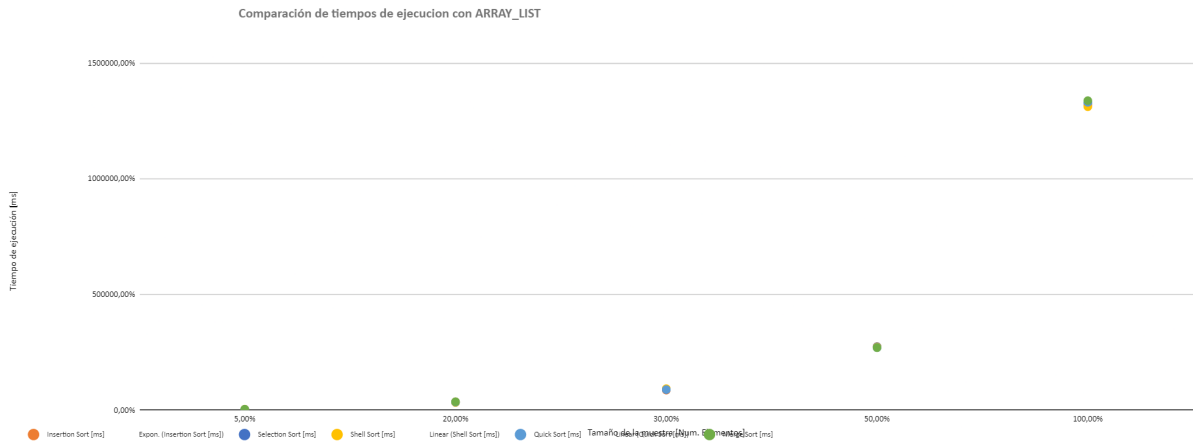
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKEDLIST)
Merge sort	3464.03	3603.64
Quick sort	3452.35	3624.49

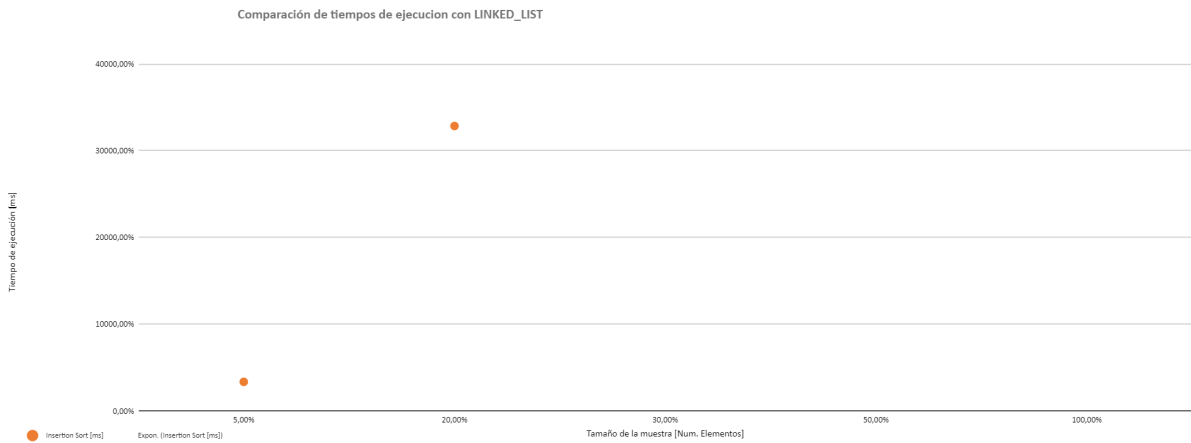
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamiento y estructuras de datos utilizadas.

Gráficas

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



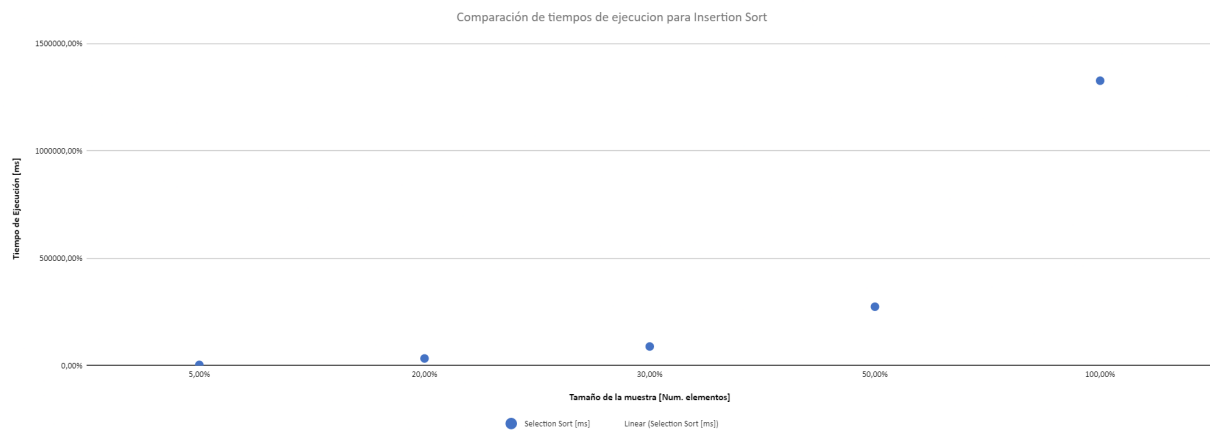
- Comparación de rendimiento LINKEDLIST.



- Comparación de rendimiento para Insertion Sort.



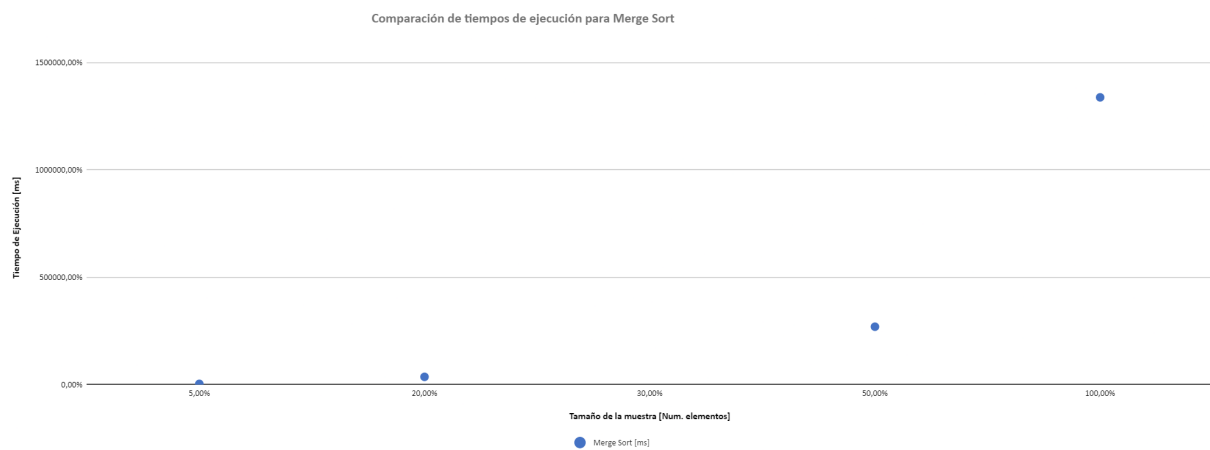
- Comparación de rendimiento para Selection Sort.



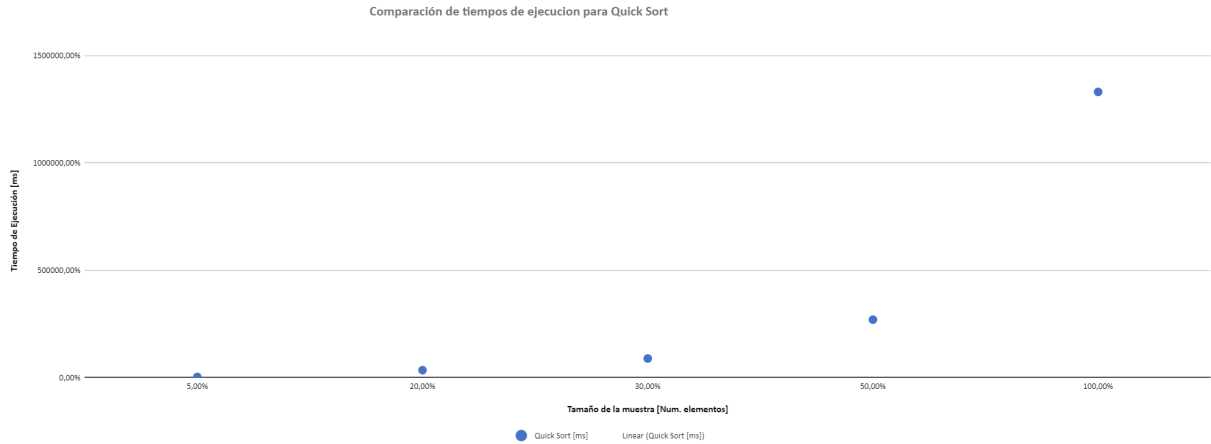
o Comparación de rendimiento para Shell Sort.



o Comparación de rendimiento para MergeSort.



o Comparación de rendimiento para QuickSort.



Máquina 2

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAYLIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1148	49,07	44,31	121,15	0,72	0,75
20.00%	4598	888,91	889,5	1006,96	15,19	14,78
30.00%	6898	2352,41	2382,83	2451,47	901,27	915,78
50.00%	11498	8181,61	8270,56	8368,13	2961,46	2878,68
100.00%	22998	40512,58	41415,54	39766,36	12517,23	13171,77

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

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKEDLIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
5.00%	1148	62,54	58,22	54,21	0,73	1,26
20.00%	4598	936,66	1035,78	902,52	15,97	15,22
30.00%	6898	2503,34	2531,97	2456,11	787,89	770,54
50.00%	11498	4012,54	4235,89	4321,12	2864,47	2824,33
100.00%	22998	42513,25	43145,12	40214,15	13459,94	13491,58

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

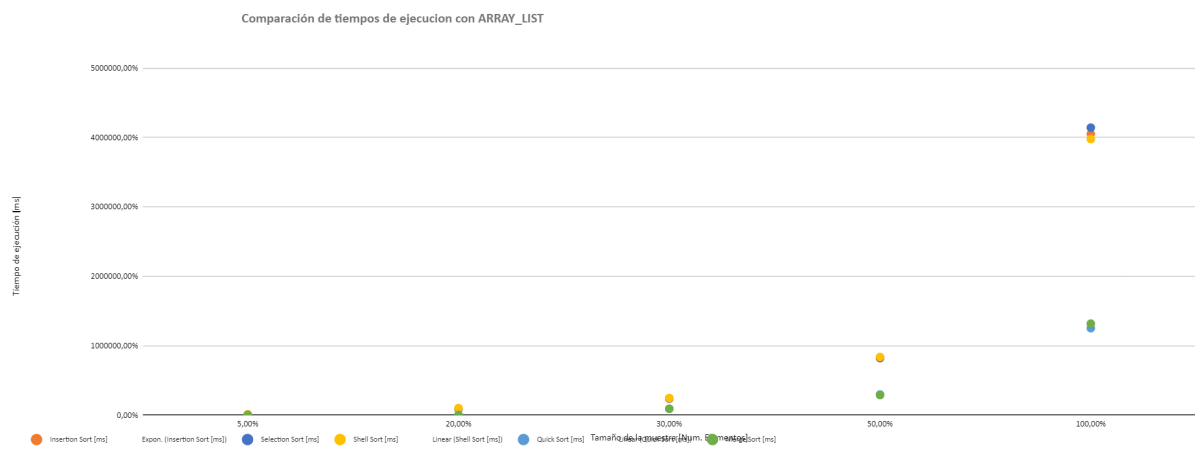
Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKEDLIST)
Merge sort	3396,35	3420,59
Quick sort	3279,17	3425,80

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

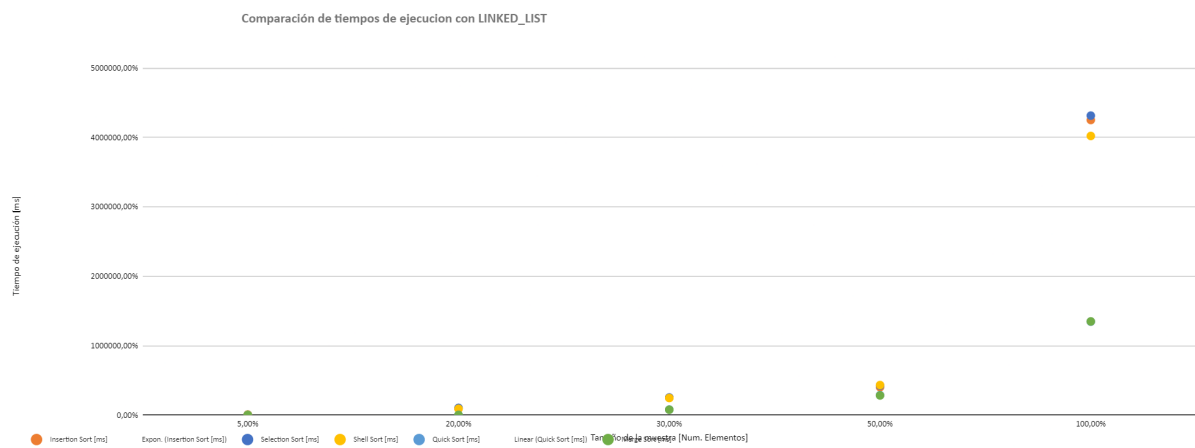
Gráficas

- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la Máquina 2.

o Comparación de rendimiento ARRAYLIST.



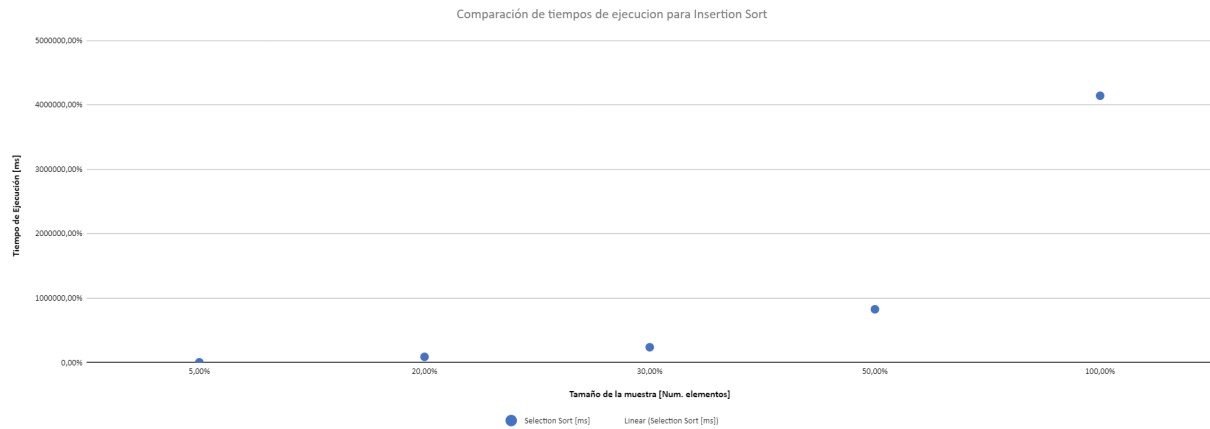
o Comparación de rendimiento LINKEDLIST.



o Comparación de rendimiento para Insertion Sort.



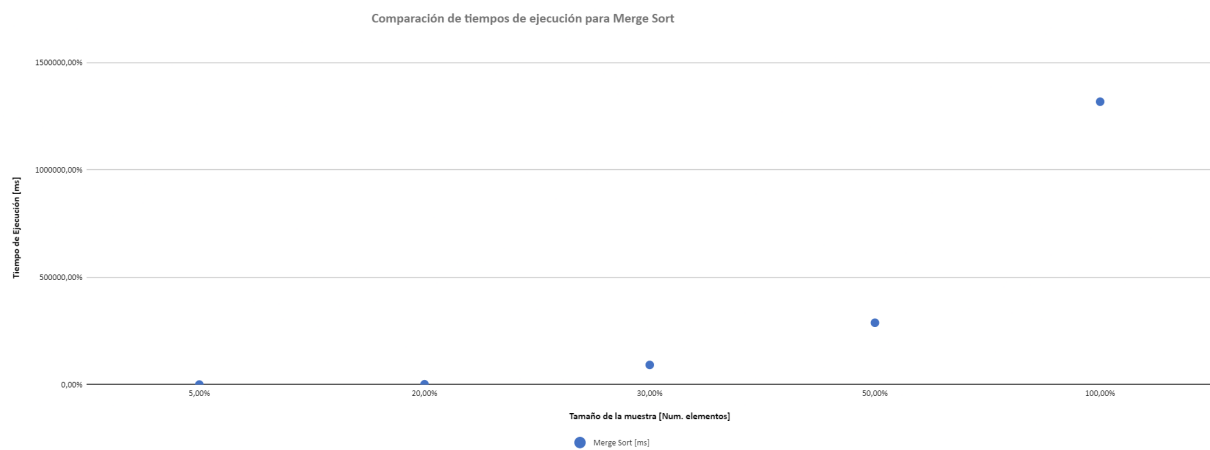
o Comparación de rendimiento para Selection Sort.



o Comparación de rendimiento para Shell Sort.



o Comparación de rendimiento para MergeSort.



o Comparación de rendimiento para QuickSort.



Preguntas de análisis

- 1) ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?
 - No. Quicksort fue un poco más eficiente que Mergesort, inclusive con datos más numerosos. En teoría debería haber ocurrido lo opuesto, pues Mergesort es más eficiente con números más altos de datos, mientras que Quicksort es más eficiente con pocos datos.
- 2) ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
 - Si existe alguna diferencia entre las máquinas. Aunque las diferencias sean un poco mínimas, se puede evidenciar que la máquina 2 ejecuta el programa más rápido.
- 3) De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?
 - Las diferencias pueden ser debido a diferencias en la capacidad de memoria de las máquinas.
- 4) ¿Qué estructura de datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?
 - Depende del tipo de ordenamiento. Para una Arraylist es más rápido el Quicksort, pero para una Singly Linked List es más efectivo usar Merge sort.
- 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.
 - 1. Quicksort
 - 2. Mergesort
 - 3. Shell
 - 4. Insertion
 - 5. Selection

