OBSERVACIONES DE LA PRACTICA

Brainer Jimenez Gonzalez - Cod 202222320
Eric Sebastián Alarcón Dolynko - Cod 202220287
Daniel Felipe Ortiz Vallejo - 202221234

	Máquina 1	Máquina 2	Máquina 3
Procesadores	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, 2592 Mhz, 6 Core(s), 12 Logical Processor(s)	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz, 1800 Mhz, 4 procesadores principales, 8 procesadores lógicos	Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz, 2904 Mhz, 2 procesadores principales, 4 procesadores lógicos
Memoria RAM (GB)	16 GB	8GB	16GB
Sistema Operativo	Windows 11	Windows 11	Windows 10 pro

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]
1.00%	49	11.29	16.71	6.63	4.86	6.83
5.00%	245	134.72	271.48	77.05	70.05	59.93
10.00%	490	480.21	926.24	217.73	145.26	171.61
20.00%	980	1901.95	3471.01	534.57	410.11	301.41
30.00%	1470	4453.74	7945.44	910.21	554.47	533.44
50.00%	2451	12329.42	21427.23	1658.79	1149.79	874.37
80.00%	3922	32511.23	60648.20	3052.80	1893.78	1624.96
100.00%	4903	49950.18	88782.28	4514.55	2562.38	2124.56

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]
1%	49	2.04	3.38	1.18	7.08	6.22
5.00%	245	41.5	68.55	16.15	139.87	68.80
10.00%	490	184	231.097	43.72	554.30	197.11
20.00%	980	590.2	990.164	103.37	2006.58	499.23
30.00%	1470	1444.3	1977.63	182.5	4599.90	1030.04
50.00%	2451	3686.54	5968.38	338.65	14287.77	2335.98

80.00%	3922	10178.65	15941.24	668.36	41762.92	5367.80
100.00%	4903	16736.32	24646.95	878.30	59164.82	7833.50

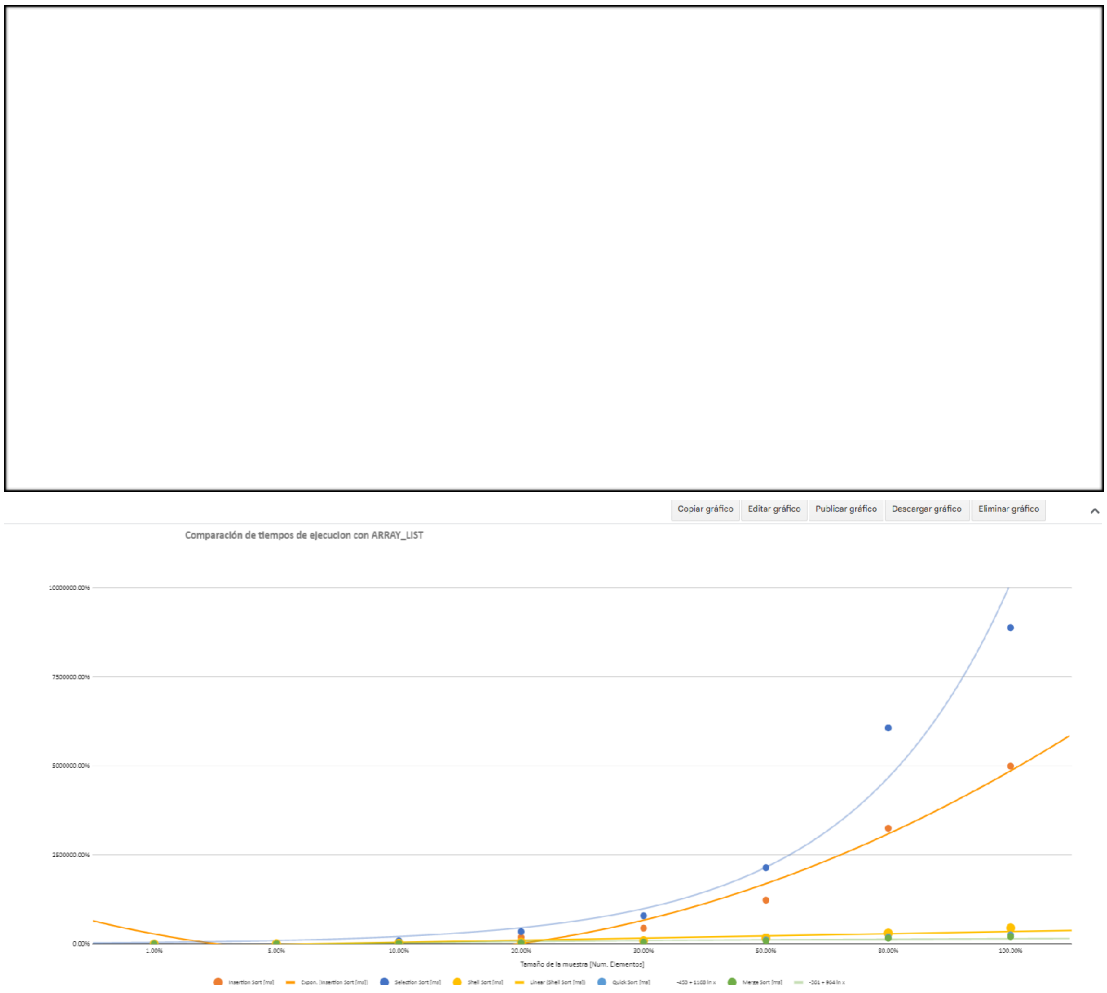
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	-361 + 964ln(x)	-1690 + 3421ln(x)
Quick sort	-453 + 1168 ln(x)	-7844x + 2253x^2

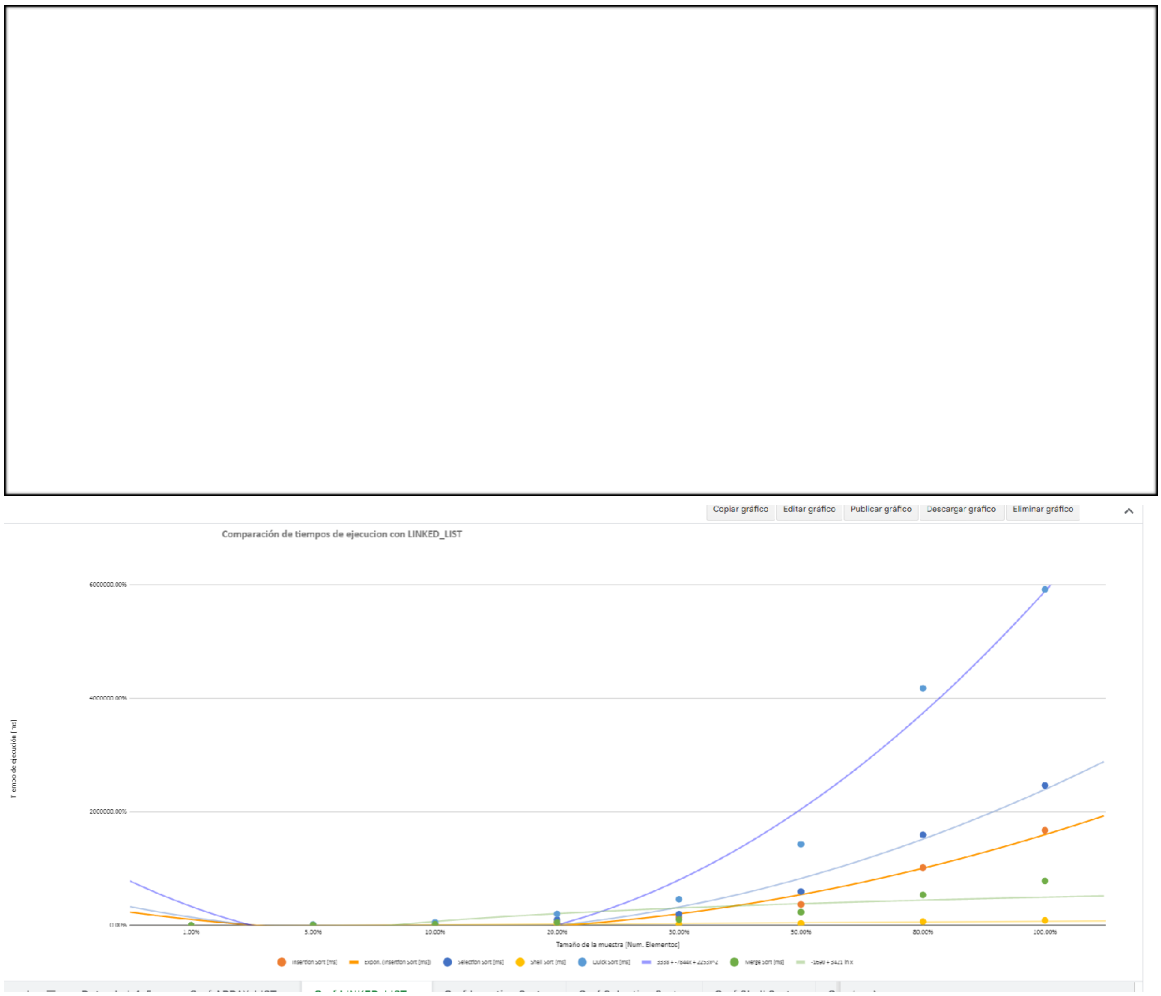
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Graficas

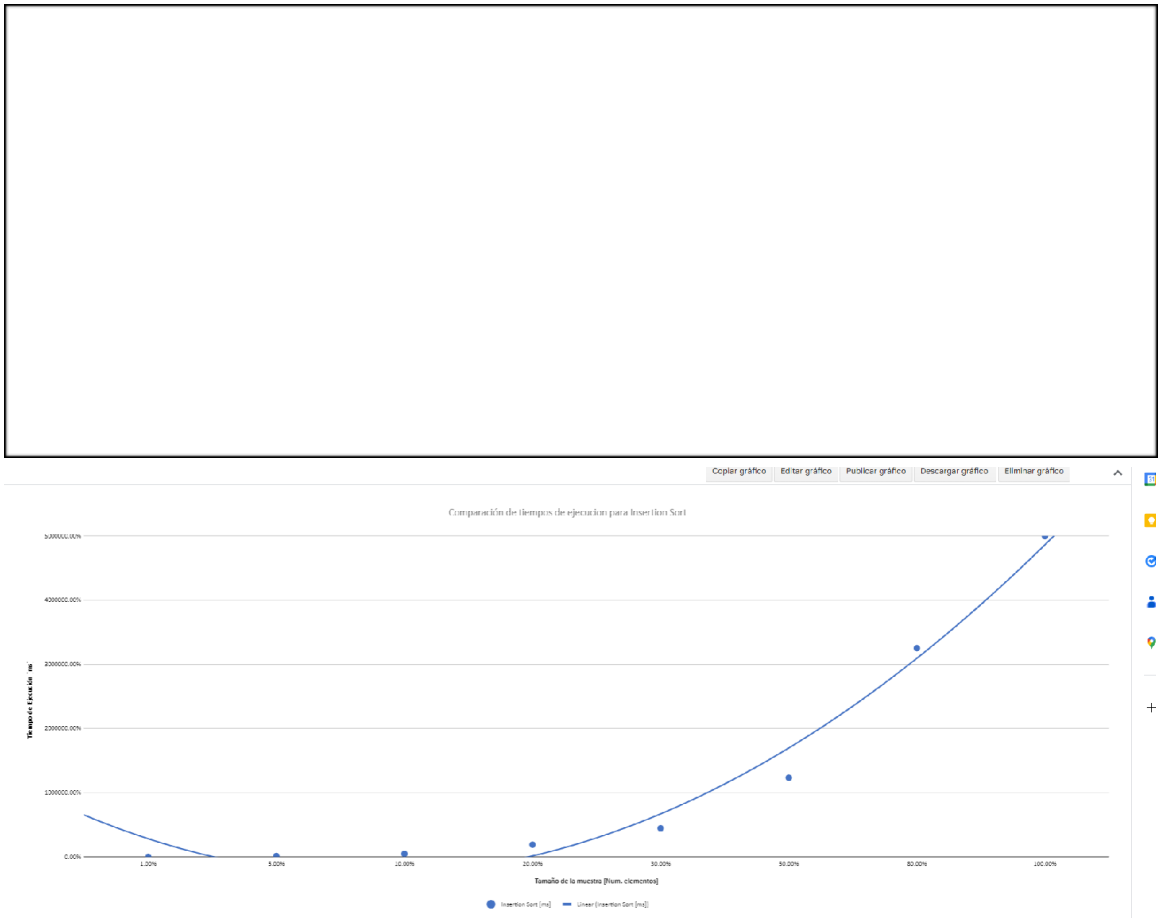
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  - Comparación de rendimiento ARRAYLIST.



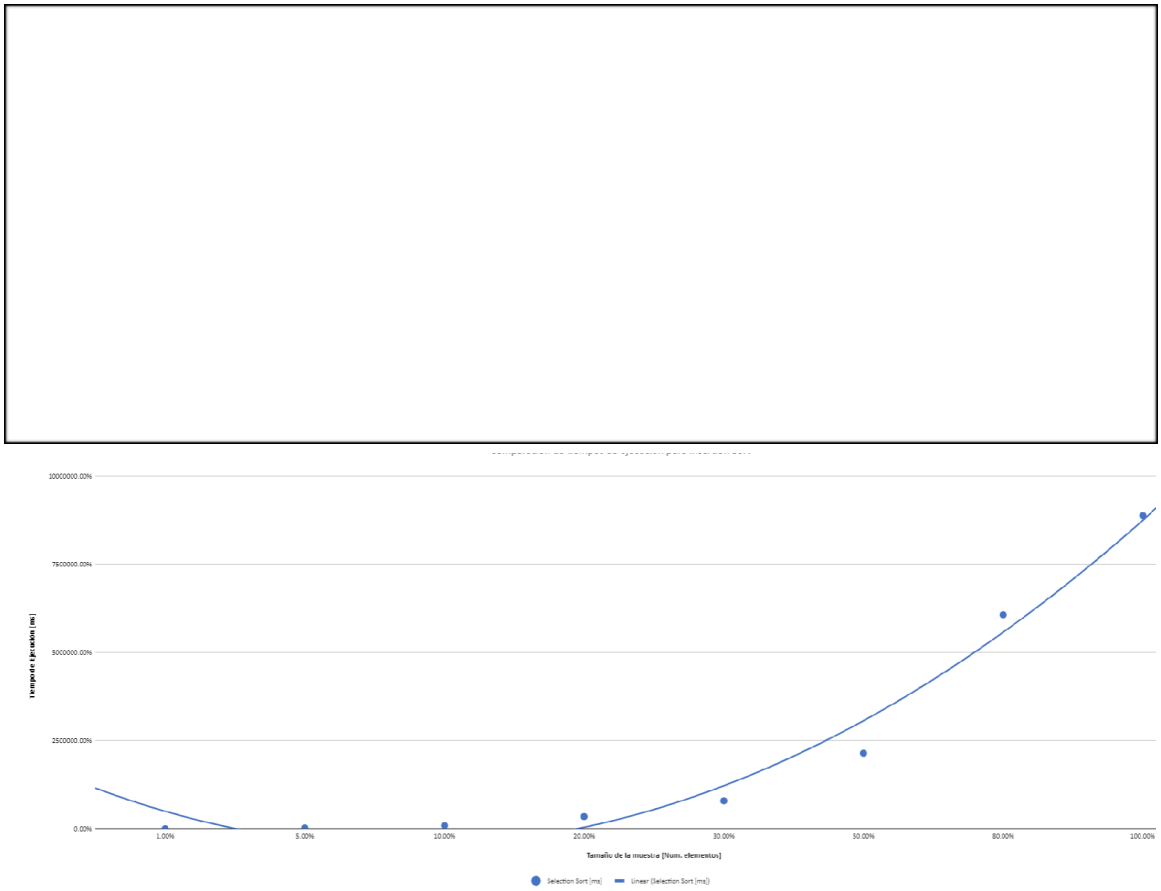
  - Comparación de rendimiento LINKED_LIST.



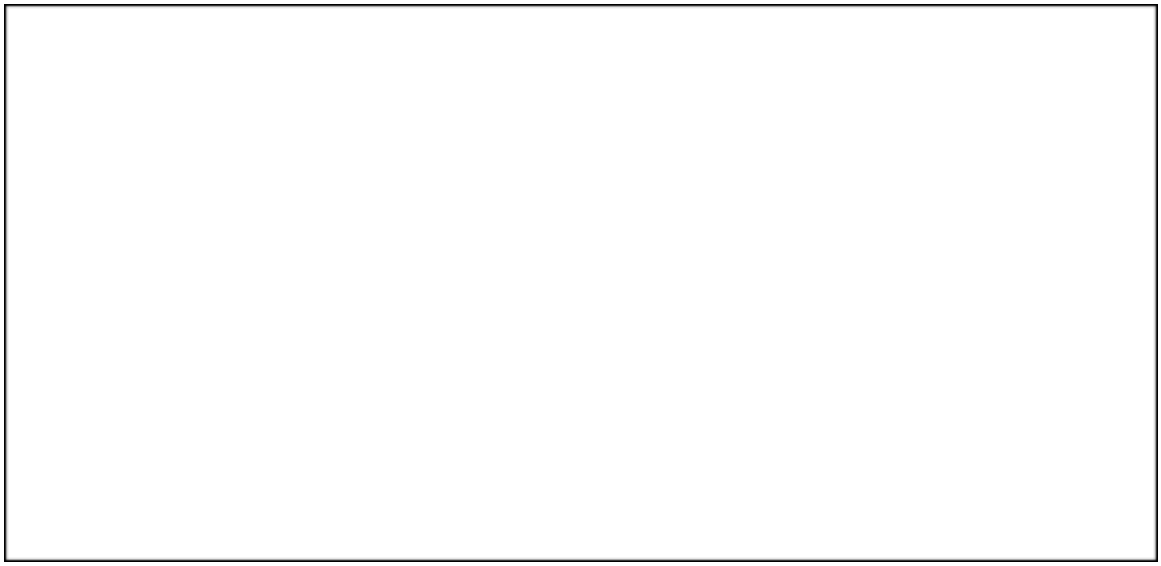
- Comparación de rendimiento para Insertion Sort.

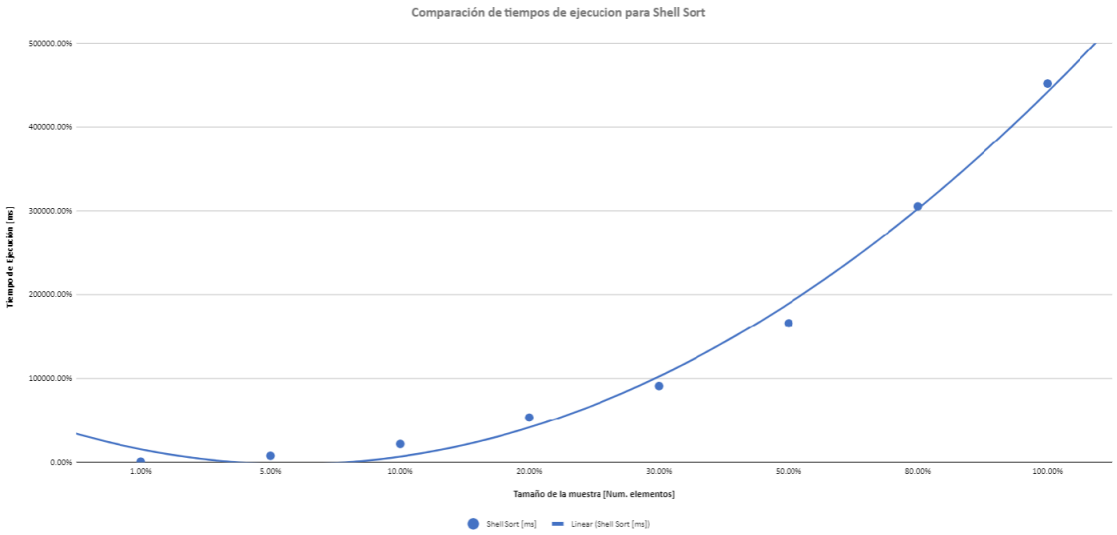


- Comparación de rendimiento para Selection Sort.

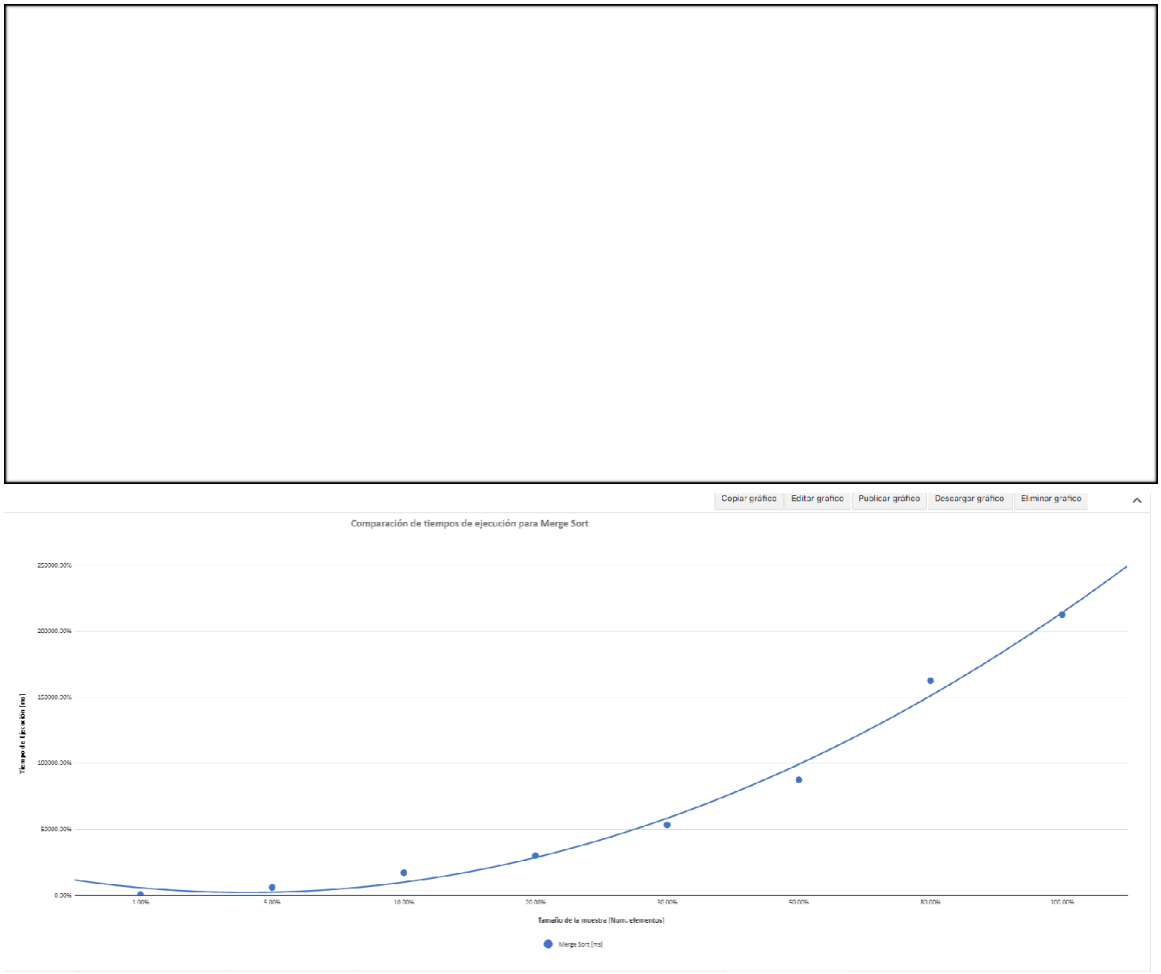


- Comparación de rendimiento para Shell Sort.

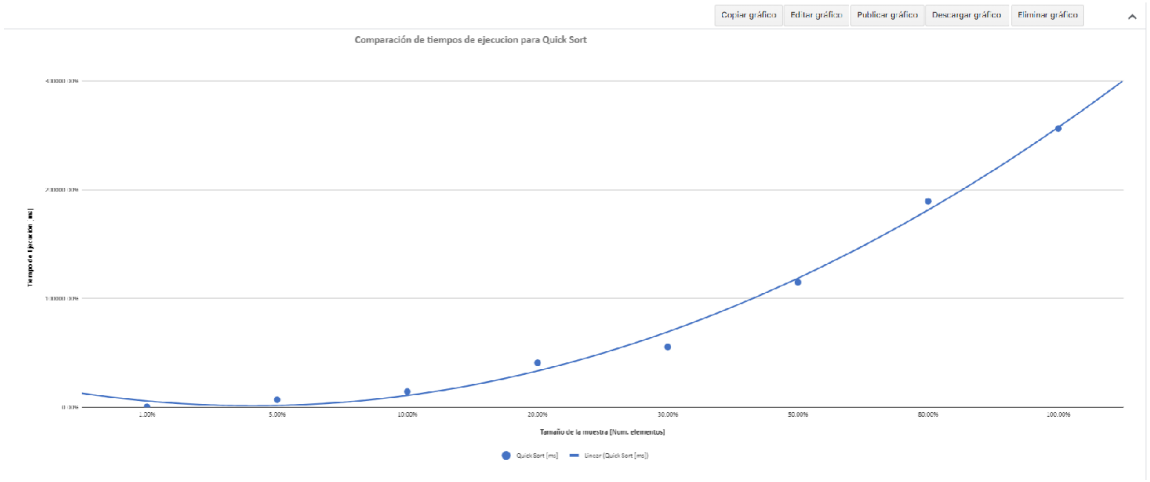




- Comparación de rendimiento para MergeSort.



- Comparación de rendimiento para QuickSort.



Maquina 2

Resultados

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]
1%	49	2.04	3.38	1.18	2.21	5.51
5.00%	245	41.5	68.55	16.15	30.1	25.4
10.00%	490	184	231.097	43.72	68.10	107.83
20.00%	980	590.2	990.164	103.37	159.82	135.61
30.00%	1470	1444.3	1977.63	182.5	412.87	248.27
50.00%	2451	3686.54	5968.38	338.65	402.87	611.68
80.00%	3922	10178.65	15941.24	668.36	570.16	520.85
100.00%	4903	16736.32	24646.95	878.30	1051.46	721.87

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]
1%	49	6.43	7.62	5.77	5.22	3.81
5.00%	245	476.3	520.19	93.66	182.74	93.97
10.00%	490	4376.06	4927.65	351.02	594.43	134.19
20.00%	980	38189.27	38876.62	2366.82	1680.81	604.69
30.00%	1470	145345.69	152477.11	5426.97	4787.51	893.91
50.00%	2451	>10 min	>10 min	16966.43	16463.90	1618.01
80.00%	3922	>10 min	>10 min	54118.07	47132.86	4310.88
100.00%	4903	>10 min	>10 min	104151.47	69961.91	6016.66

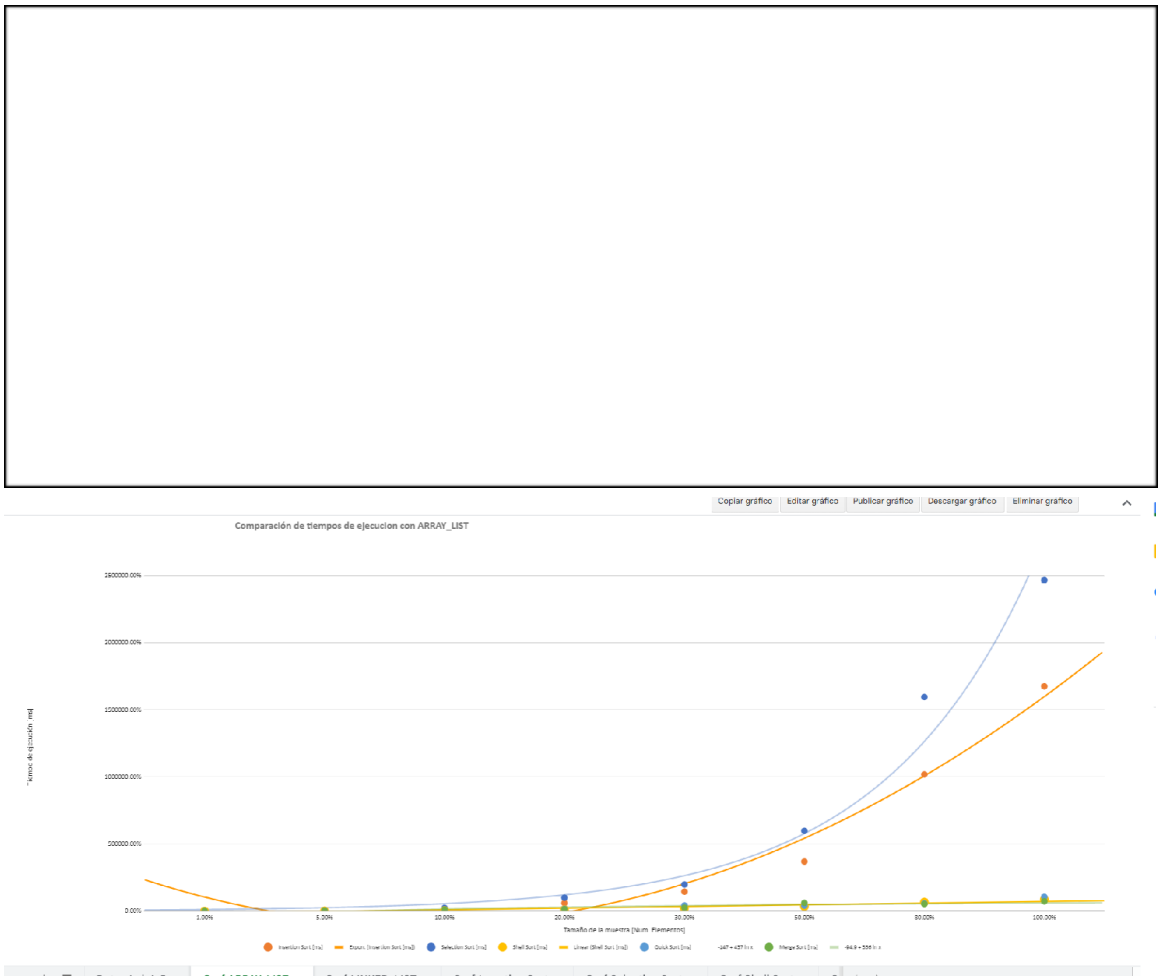
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	-94.9 + 356ln(x)	-1242 + 2623 ln(x)
Quick sort	-147 + 437 ln(x)	-9671x + 2700x^2

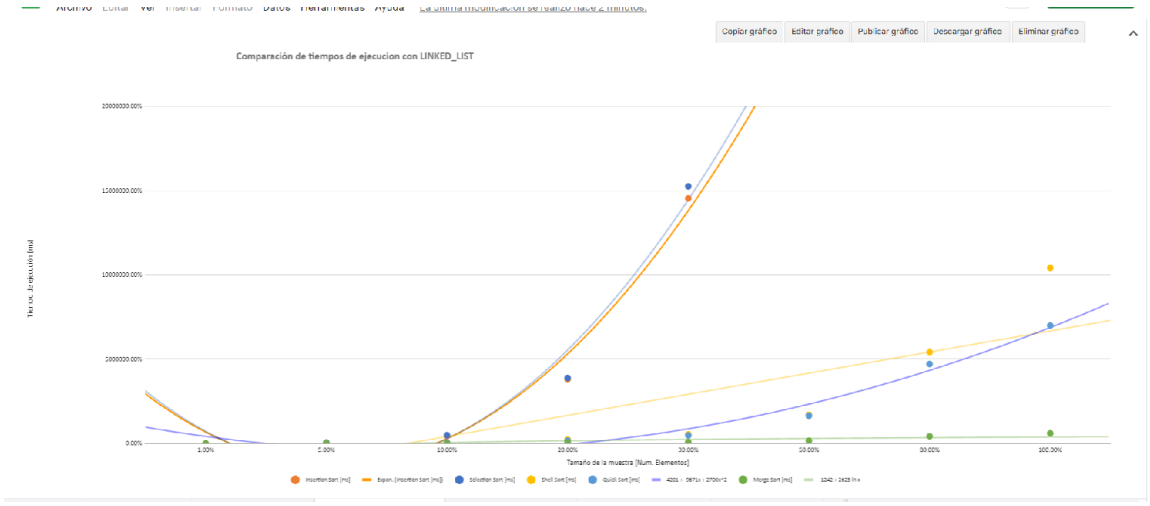
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Graficas

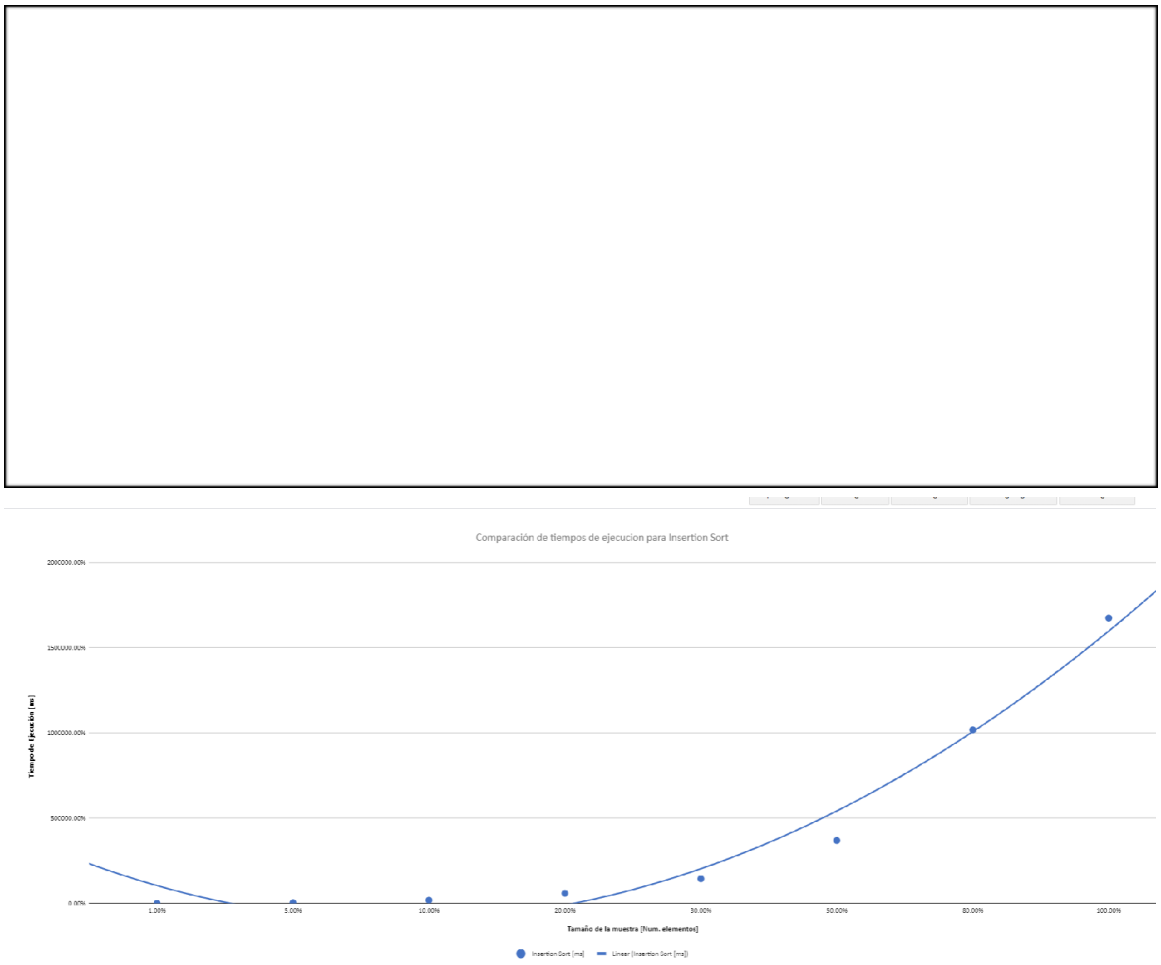
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
    - Comparación de rendimiento ARRAYLIST.



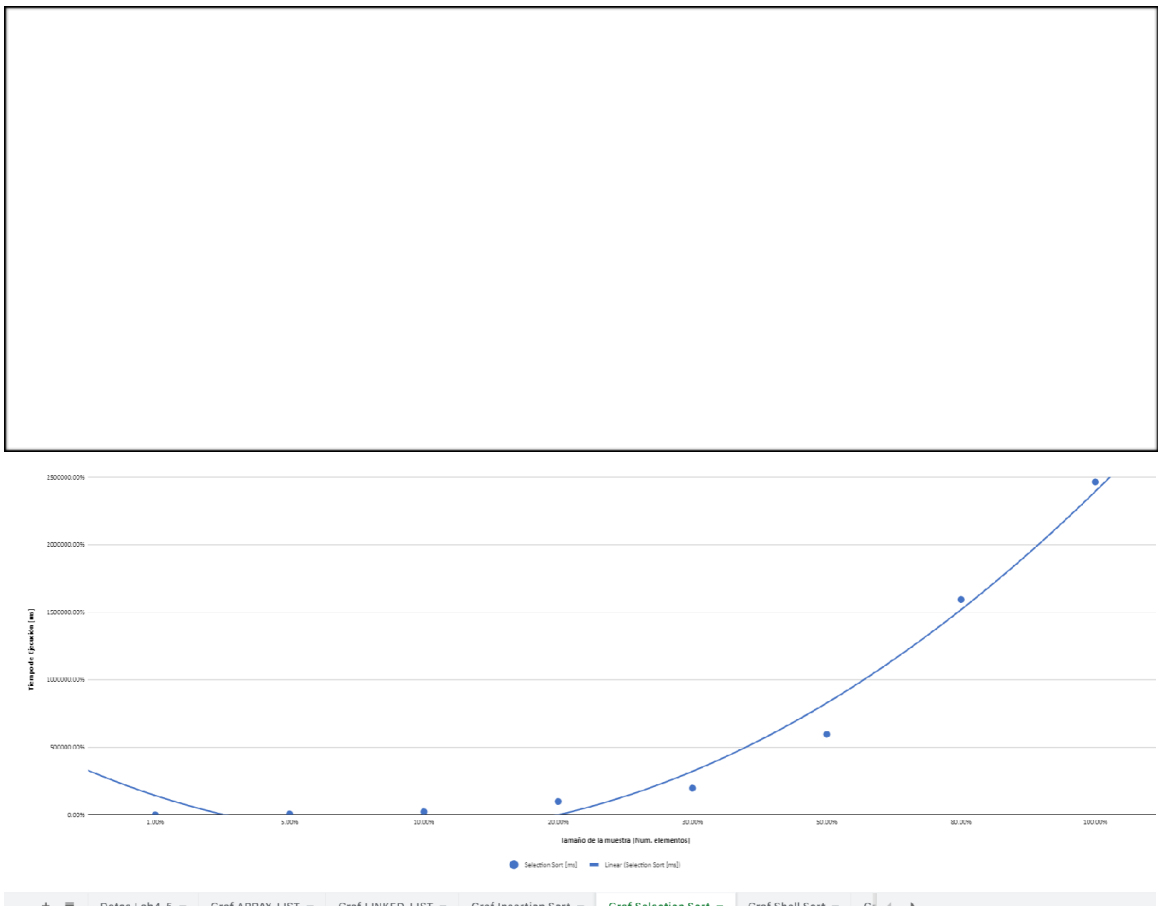
- Comparación de rendimiento LINKED_LIST.



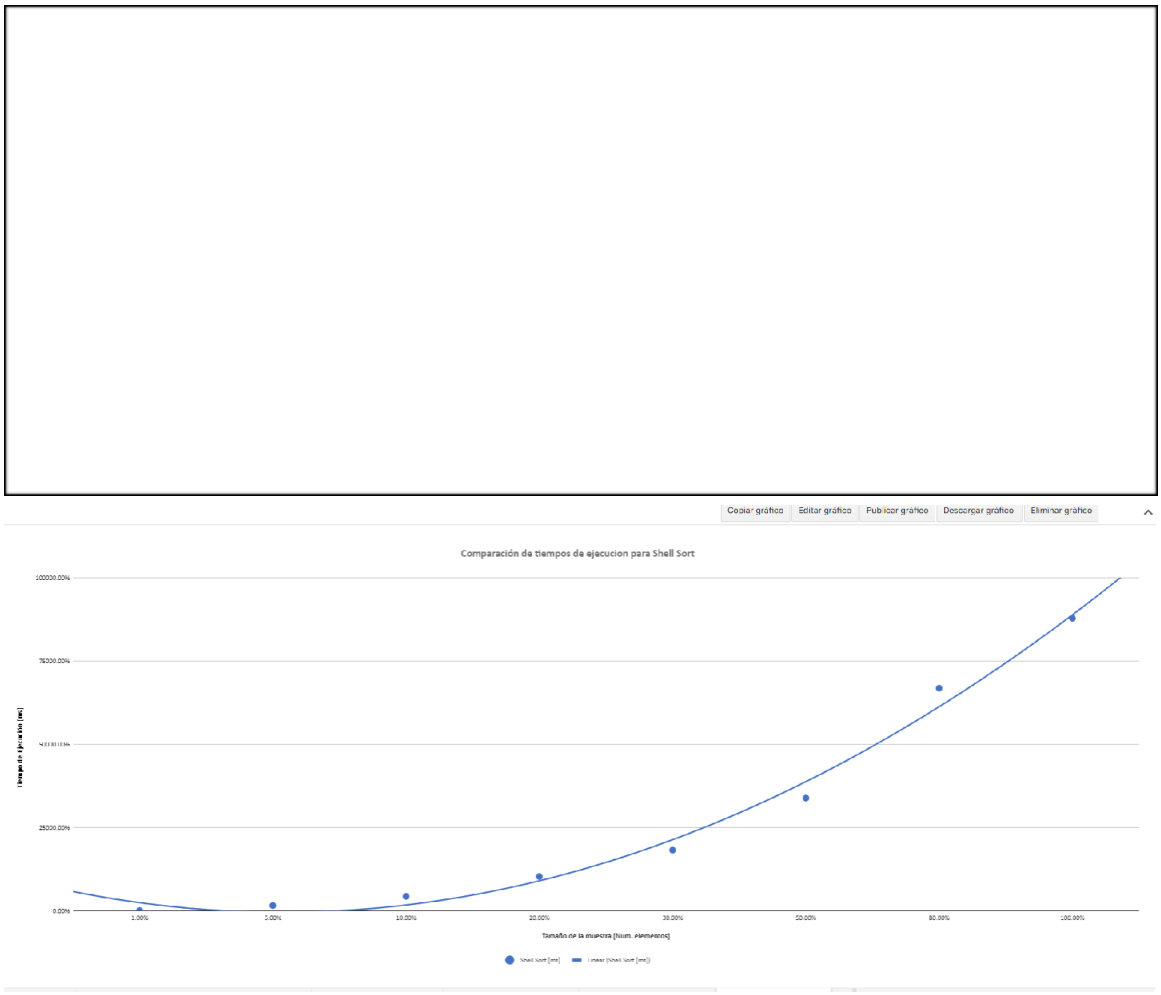
- Comparación de rendimiento para Insertion Sort.



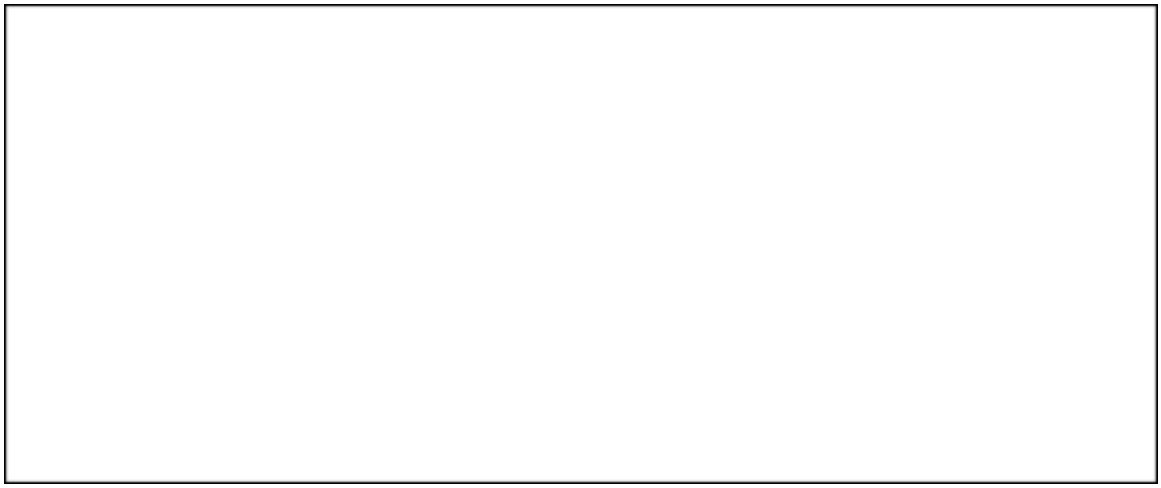
- Comparación de rendimiento para Selection Sort.

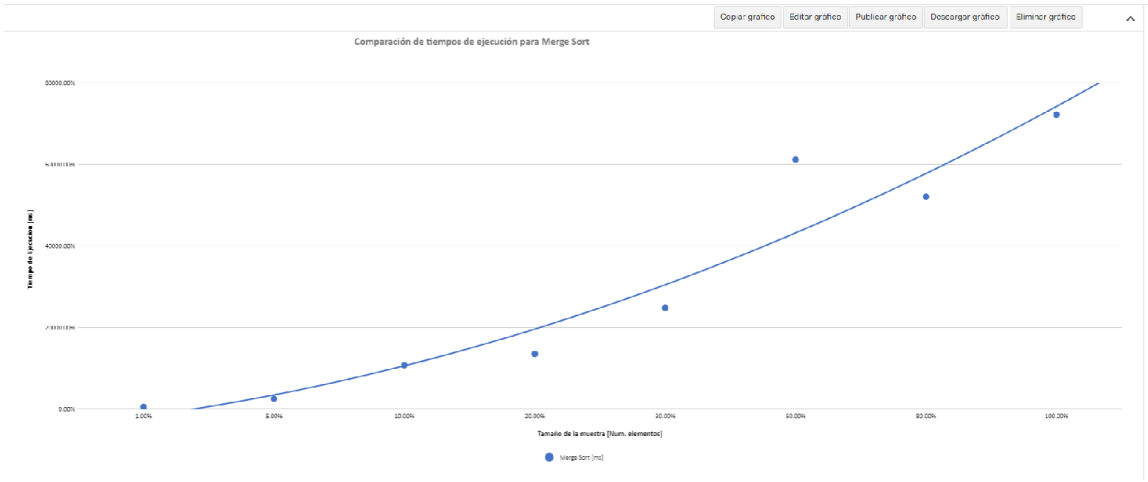


- Comparación de rendimiento para Shell Sort.

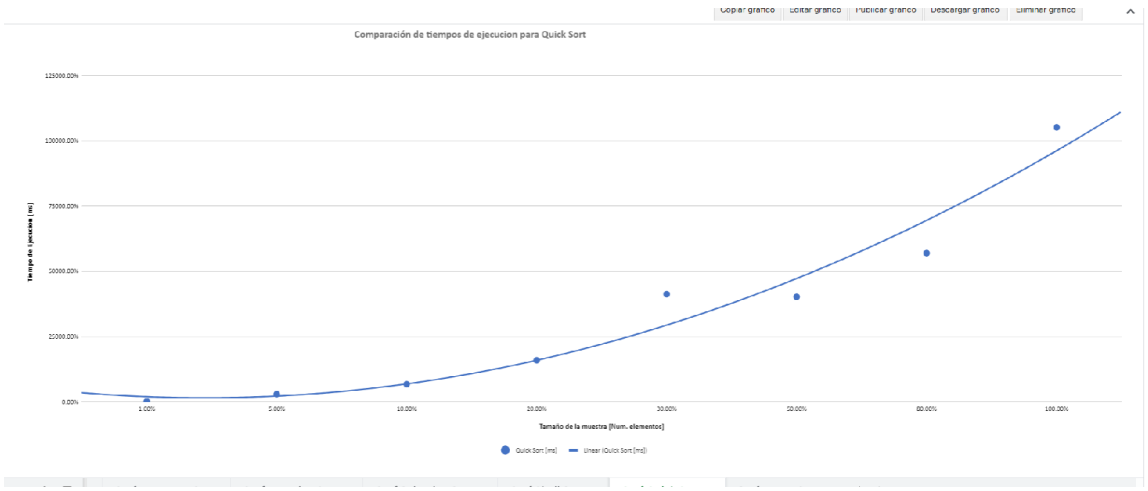


- Comparación de rendimiento para MergeSort.





- 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]
1%	49	7.45	5.93	3.17	4.32	1.36
5.00%	245	215.79	104.37	31.94	19.02	14.79
10.00%	490	456.036	1042.84	88.77	43.09	36.75
20.00%	980	1234.56	3093.55	199.55	85.58	69.51
30.00%	1470	3843.64	5204.79	353.86	198.24	114.38
50.00%	2451	8679.57	13126.63	634.78	262.27	202.58
80.00%	3922	21591.45	34365.41	1541.28	1008.98	611.99
100.00%	4903	34.081.75	56.648.33	2228.14	1365.33	526.30

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

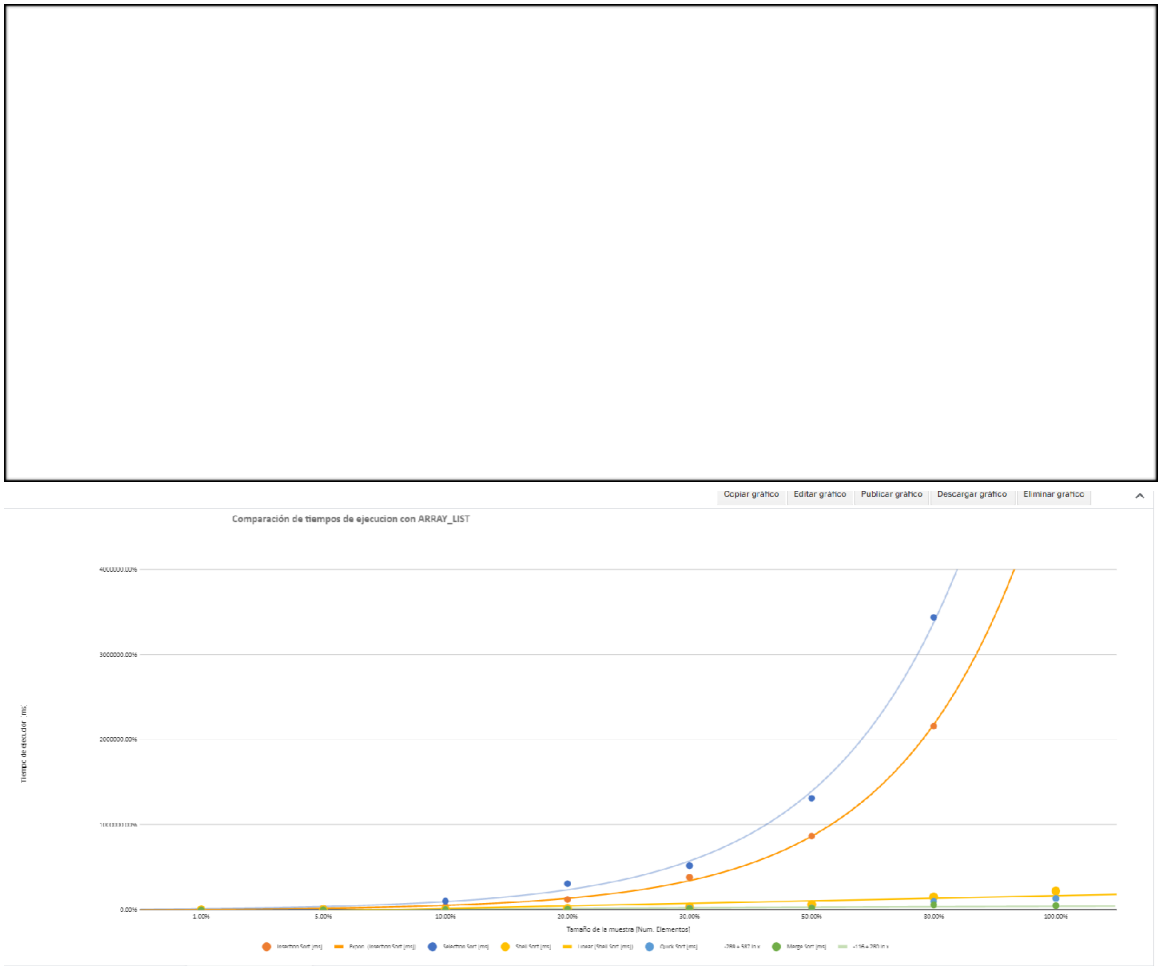
Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
1%	49	13.77	24.95	17.78	7.65	4.13
5.00%	245	1299.60	1646.97	163.37	117.61	31.21
10.00%	490	898.275	12987.70	570.53	525.07	207.30
20.00%	980	88123.87	79528.31	5425.80	1705.79	242.37
30.00%	1470	295476.33	265590.71	15245.27	4292.24	501.27
50.00%	2451	>10 min	>10 min	31895.61	14138.62	1752.98
80.00%	3922	>10 min	>10 min	90381.42	40883.97	4059.83
100.00%	4903	>10 min	>10 min	156925.56	58641.39	6511.26

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

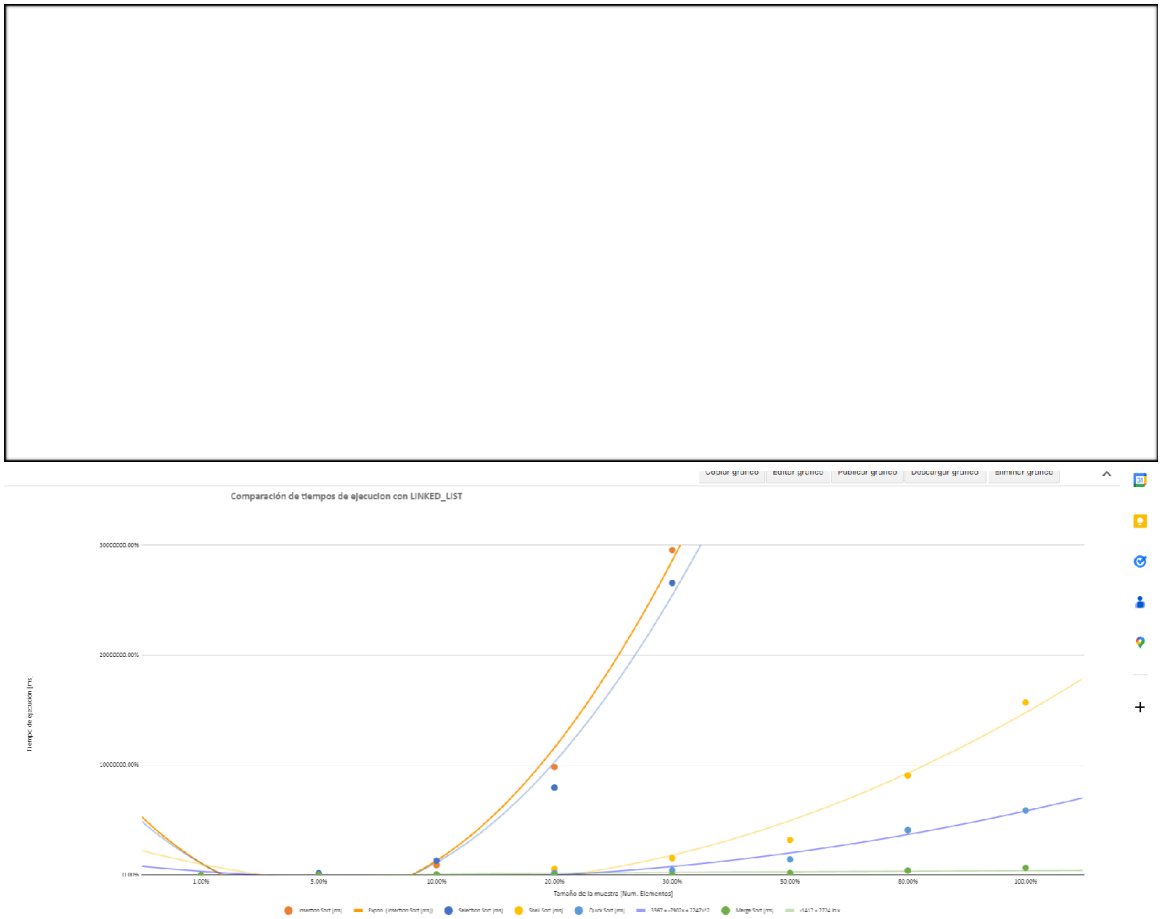
Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	-116 +280ln(x)	-1417 + 2724 ln(x)
Quick sort	-289 + 587ln(x)	-7902x + 2247x^2

Graficas

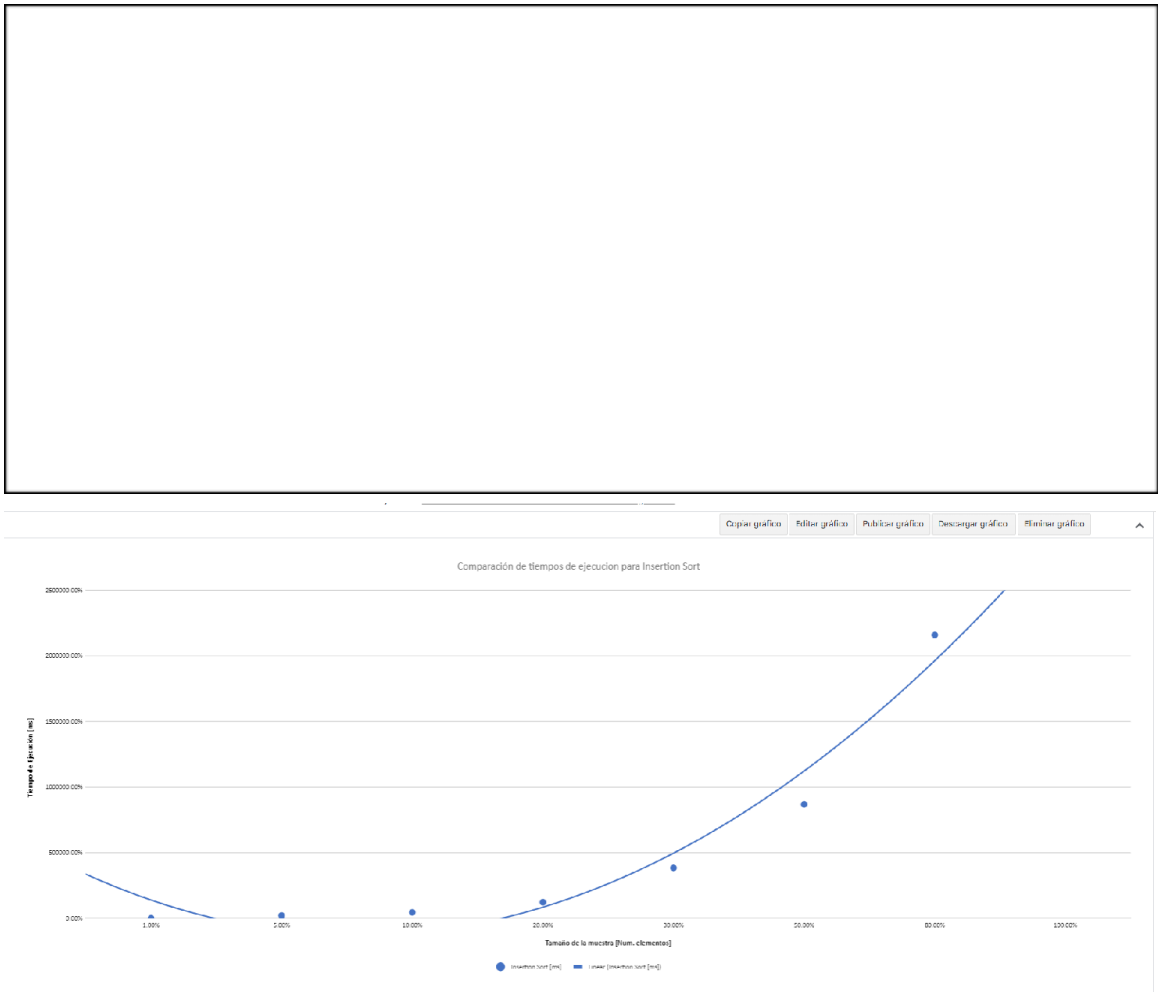
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 3.**
    - Comparación de rendimiento ARRAYLIST.



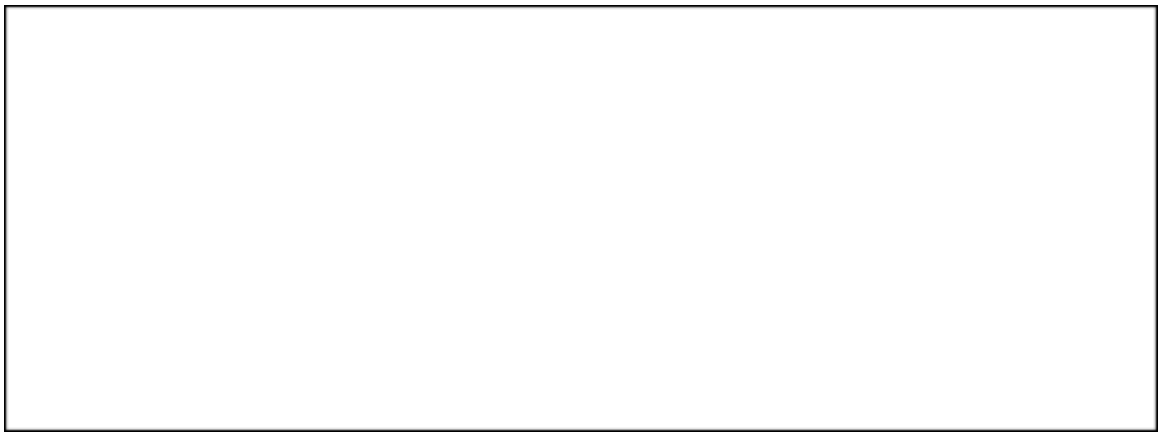
- Comparación de rendimiento LINKED_LIST.

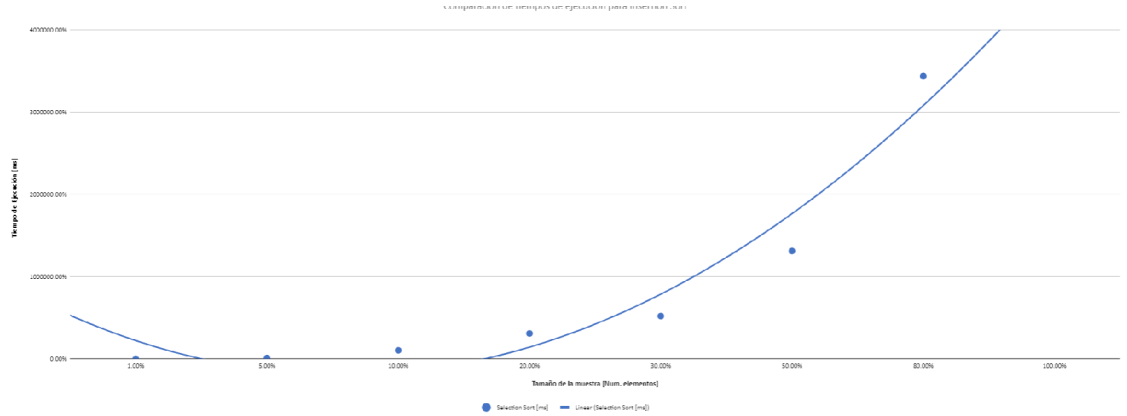


- Comparación de rendimiento para Insertion Sort.

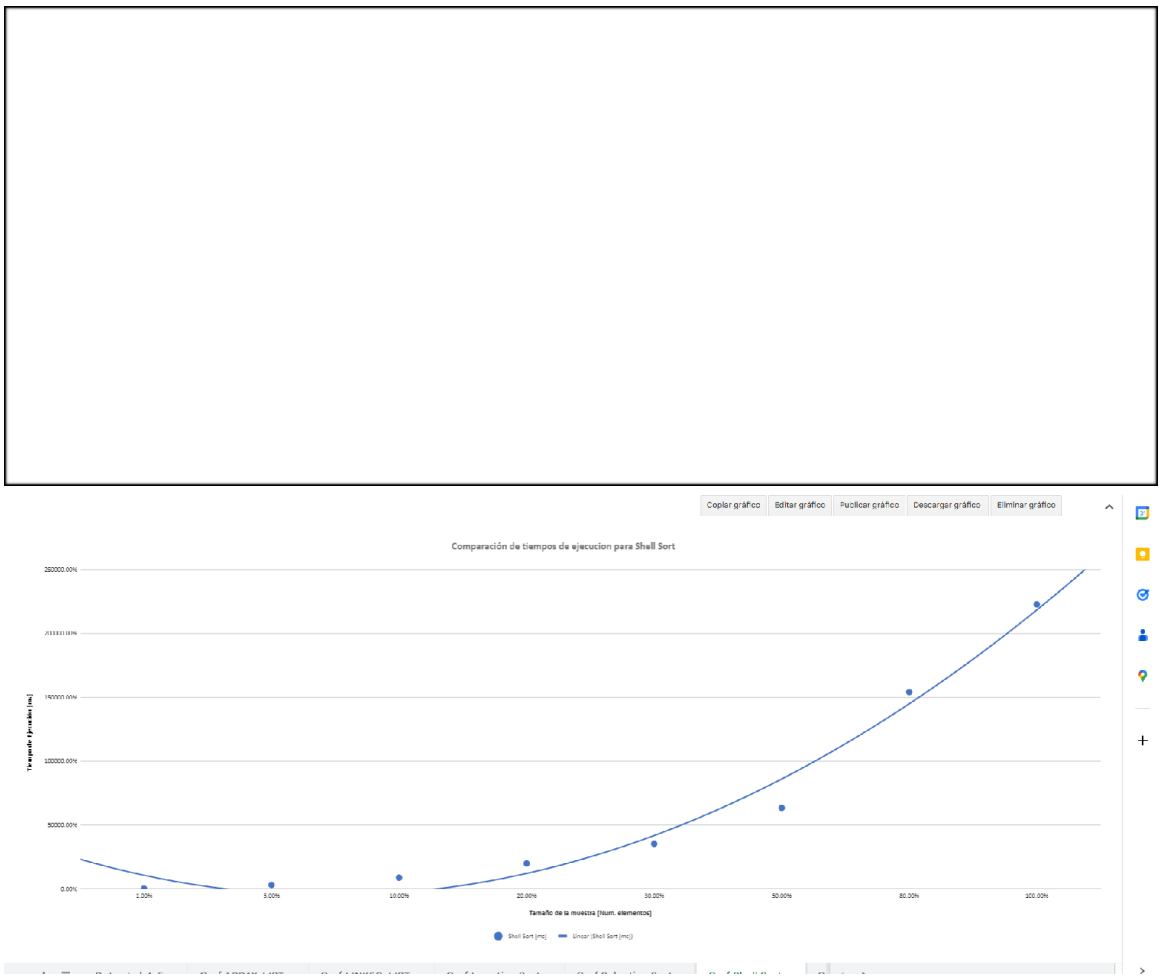


- Comparación de rendimiento para Selection Sort.

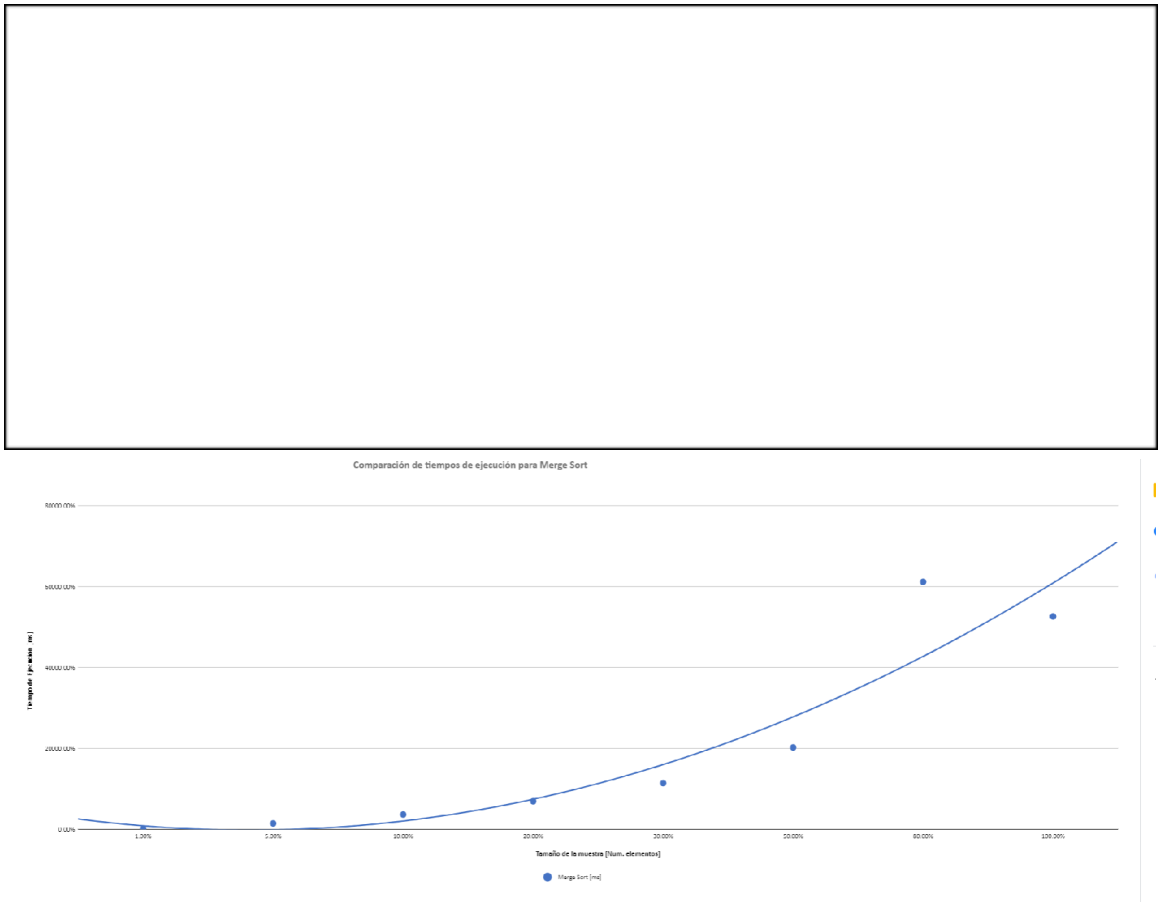




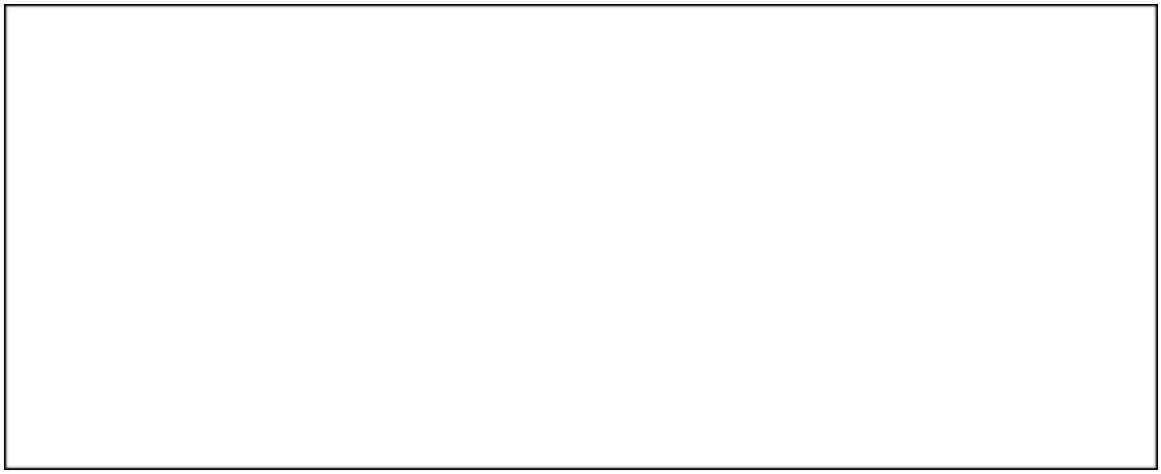
- Comparación de rendimiento para Shell Sort.

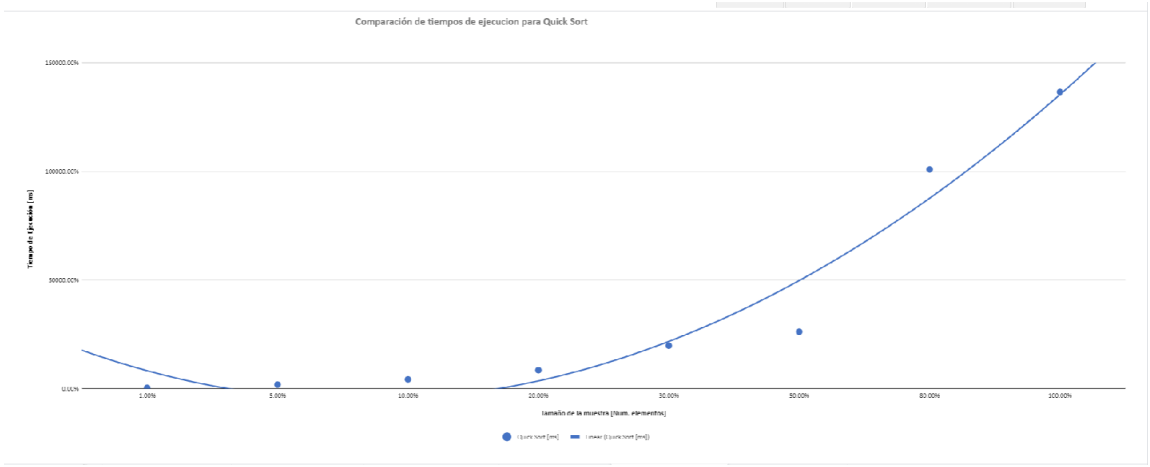


- Comparación de rendimiento para MergeSort.



- Comparación de rendimiento para QuickSort.





Preguntas de análisis

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

En general el comportamiento si es acorde a lo enunciado teóricamente, puesto que el merge sort y el quick sort son los más rápidos en cargar los datos, no obstante cabe recalcar que el tiempo entre estos dos sorts debería ser parecidos ya que ambos en el caso promedio tienen una complejidad de O(nlogn), sin embargo al considerar que en el linked list, fue donde realmente se evidencio un cambio grande entre el quick y el merge se puede llegar a concluir que en un linked list, la elección del pivote siempre hará que se tenga que hacer una iteración en el lado izquierdo de la lista ya que no hay apuntadores para poder ver si la parte que ya se recorrió es menor que el pivot y está organizado. De los demás aspectos, si coincide con lo teórico.

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

Si existe, en el caso de la máquina 3 se puede ver que el quick sort en todo momento dura más tiempo procesando que el merge tanto en array list como en single linked list, mientras que en la máquina 2 y 1, el quick sort en un array list en pocas ocasiones es más eficiente que el merge sort. También el tiempo

de ejecución de la máquina 3 suele ser considerablemente mayor que el de las demás máquinas. No obstante, en general, se puede apreciar que los comportamientos generales de la complejidad de los algoritmos siguen siendo parecidas y congruentes.

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

Las diferencias se pueden deber a la diferencia en memoria RAM y en el CPU que son los encargados de procesar los datos, no obstante considerando que la máquina 3 tiene mejor RAM y una CPU mejor también se puede tener en cuenta el estado físico del computador, puesto que puede estar maltratado o viejo, y a pesar de tener las estadísticas o requerimientos que se muestran en la parte superior del laboratorio, su eficiencia disminuiría considerablemente.

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

Si se tiene en cuenta los tiempos de ejecución un ARRAY LIST es mucho más rápido que el single linked list, por el hecho de que a la hora de organizar un arreglo y se necesite saber la posición anterior, se puede acceder fácilmente por el índice, no obstante en un single linked list si se quiere comparar un elemento con el anterior, se tendría que recorrer toda la lista otra vez hasta llegar al elemento anterior ya que los apuntadores solo van hacia el siguiente elemento y no saben de dónde vienen.

- Para el escenario de ordenamiento de impuestos, 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 relación con los tiempos de ejecución) para ordenar la mayor cantidad de impuestos.

El ranking sería:

1.(Merge Sort) Más eficiente

2.(Quick Sort)

3.(Shell Sort)

4.(Insertion Sort)

5.(Selection Sort) Menos eficiente