OBSERVACIONES DE LA PRACTICA

Santiago Pardo - 202013025

Juan José Ramírez - 202013228

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | 2,4 GHz Intel Core i5 de cuatro núcleos | Intel (R) Core (TM) i3-8145U CPU @2.10 GHz 2.30 GHz |
| Memoria RAM (GB) | 8 GB | 4 GB |
| Sistema Operativo | MacOS Catalina | Windows |

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 | 659.66 | 708.78 | 49.68 | 33.97 | 36.24 |
| 2000 | 2766.22 | 2952.30 | 92.53 | 71.45 | 66.03 |
| 4000 | 11596.46 | 12072.17 | 192.60 | 128.43 | 128.01 |
| 8000 | 44995.56 | 54425.11 | 474.38 | 266.61 | 268.53 |
| 16000 | 185267.84 | 236881.24 | 1113.22 | 572.18 | 571.60 |
| 32000 | 820793.35 | Tiempo excedido | 2545.26 | 1194.91 | 1274.73 |
| 64000 | Tiempo excedido | Tiempo excedido | 6164.48 | 2662.14 | 2792.88 |
| 128000 | Tiempo excedido | Tiempo excedido | 14872.75 | 6445.28 | 5705.52 |
| 256000 | Tiempo excedido | Tiempo excedido | 36125.95 | 12362.07 | 12374.87 |
| 512000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |

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 | 42499.89 | 36960.79 | 2285.80 | 1768.95 | 223.16 |
| 2000 | 349214.70 | 313236.85 | 9617.58 | 8128.33 | 809.47 |
| 4000 | Tiempo excedido | Tiempo excedido | 45595.77 | 32631.12 | 3285.85 |
| 8000 | Tiempo excedido | Tiempo excedido | 231993.82 | 146093.88 | 13021.16 |
| 16000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | 624926.17 | 54777.80 |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 228544.35 |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 512000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |

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 |  |

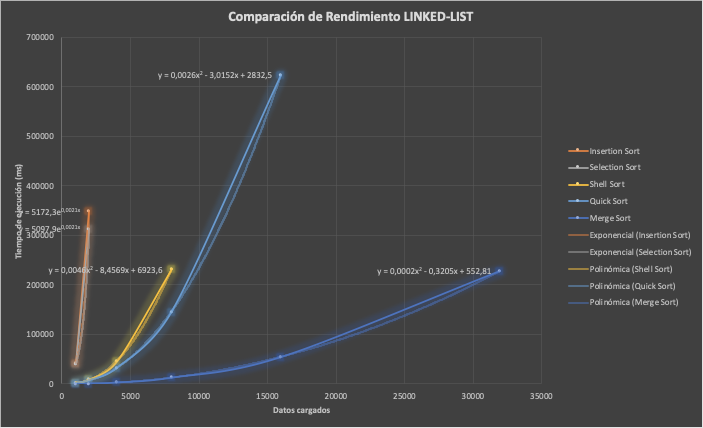
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

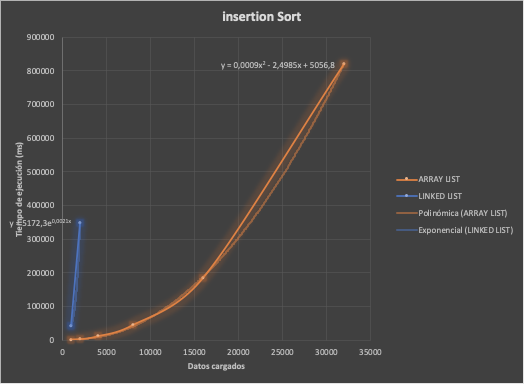
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.



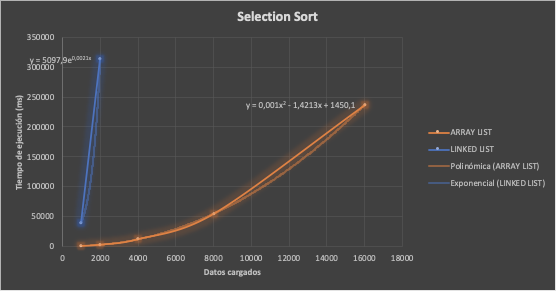
* + Comparación de rendimiento LINKED\_LIST.



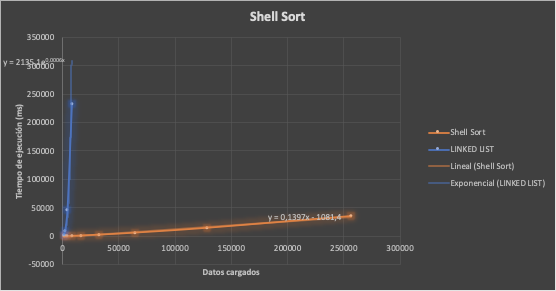
* + Comparación de rendimiento para Insertion Sort.



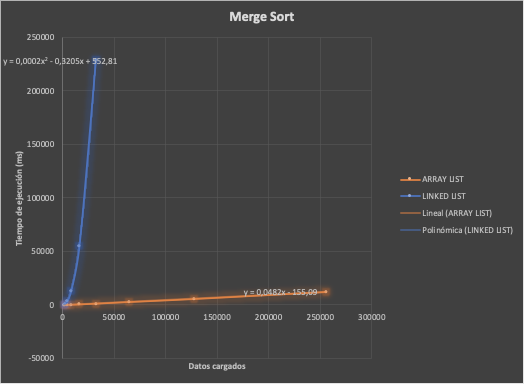
* + Comparación de rendimiento para Selection Sort.



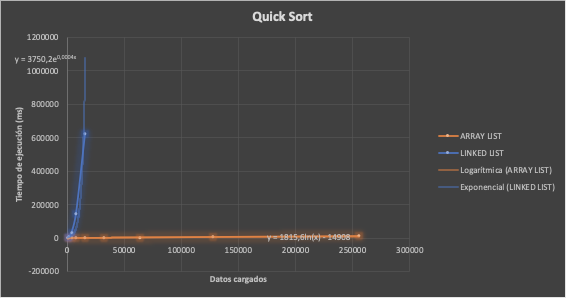
* + Comparación de rendimiento para Shell Sort.



* + 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 | 953,12 | 1031,25 | 62,5 | 15,625 | 31,25 |
| 2000 | 2828,12 | 3171,87 | 104,5 | 78,125 | 62,5 |
| 4000 | 12046,87 | 13609,37 | 203,12 | 109,375 | 140,625 |
| 8000 | 45859,37 | 60697,5 | 687,5 | 421,875 | 296,875 |
| 16000 | 196890,62 | 239031,25 | 1046,87 | 703,125 | 875 |
| 32000 | 843890,62 | 1012796,87 | 3312,5 | 1343,75 | 1359,375 |
| 64000 | Tiempo excedido | Tiempo excedido | 6203,12 | 2984,375 | 2687,5 |
| 128000 | Tiempo excedido | Tiempo excedido | 16187,5 | 6593,75 | 5687,5 |
| 256000 | Tiempo excedido | Tiempo excedido | 46546,87 | 20968,75 | 12406,25 |
| 512000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | No calculado | No calculado |

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 | 58062,5 | 49937,5 | 2625 | 1718,75 | 203,125 |
| 2000 | 42762,5 | 407718,75 | 10875 | 8312,5 | 1000 |
| 4000 | Tiempo excedido | Tiempo excedido | 57875 | 43453,125 | 3718,75 |
| 8000 | Tiempo excedido | Tiempo excedido | 286750 | 180296,875 | 15750 |
| 16000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 110046,87 |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | 450596,875 |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo  excedido |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 512000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |

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 | x |
| Quick sort |  |  |

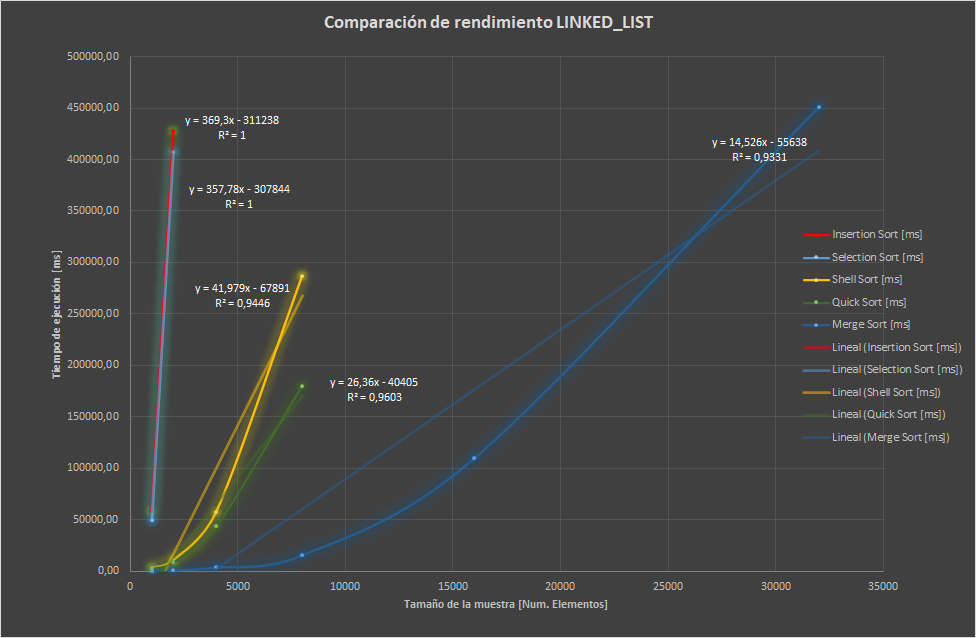
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

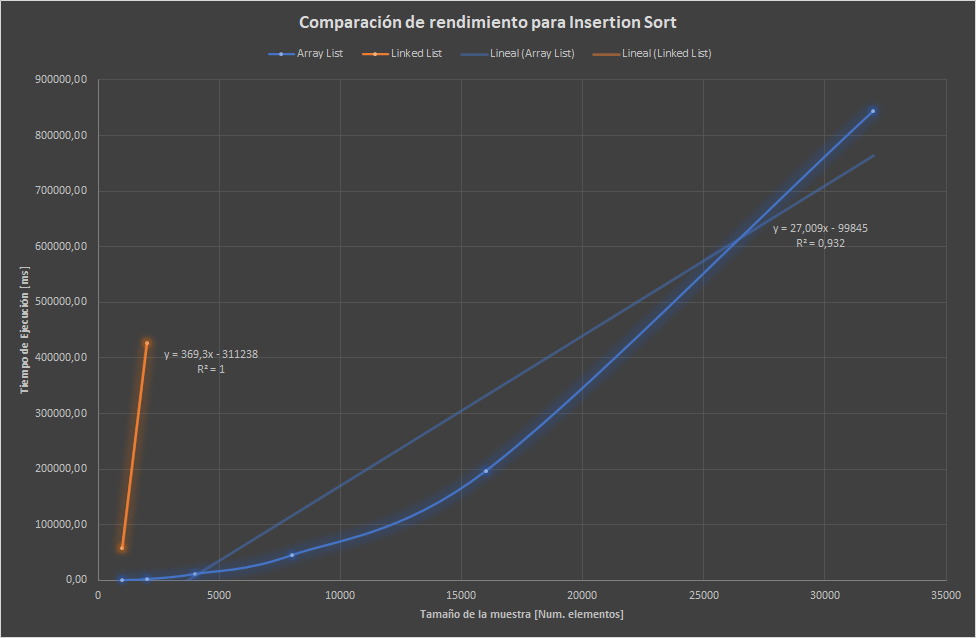
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.



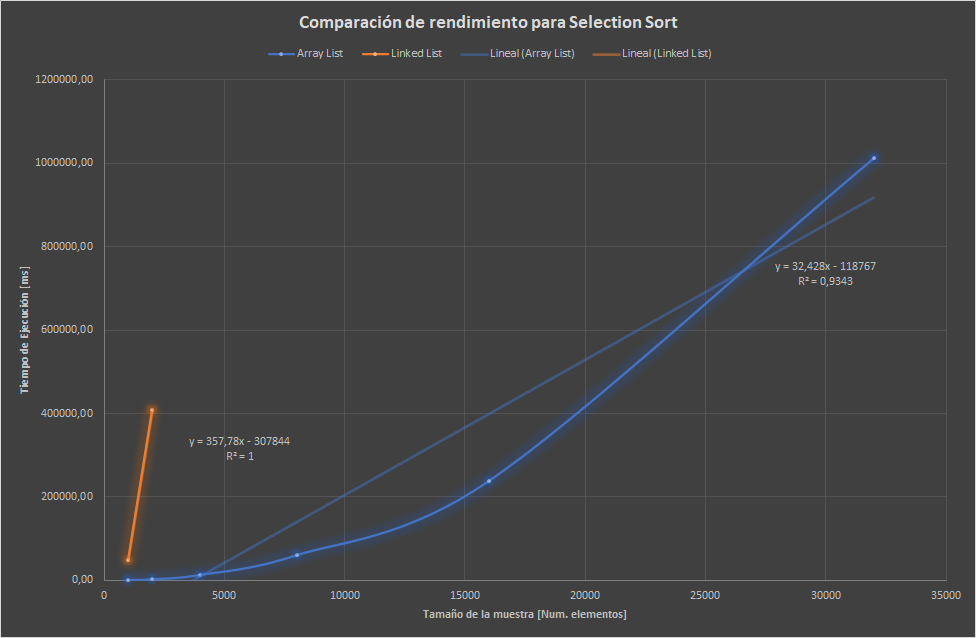
* + Comparación de rendimiento LINKED\_LIST.



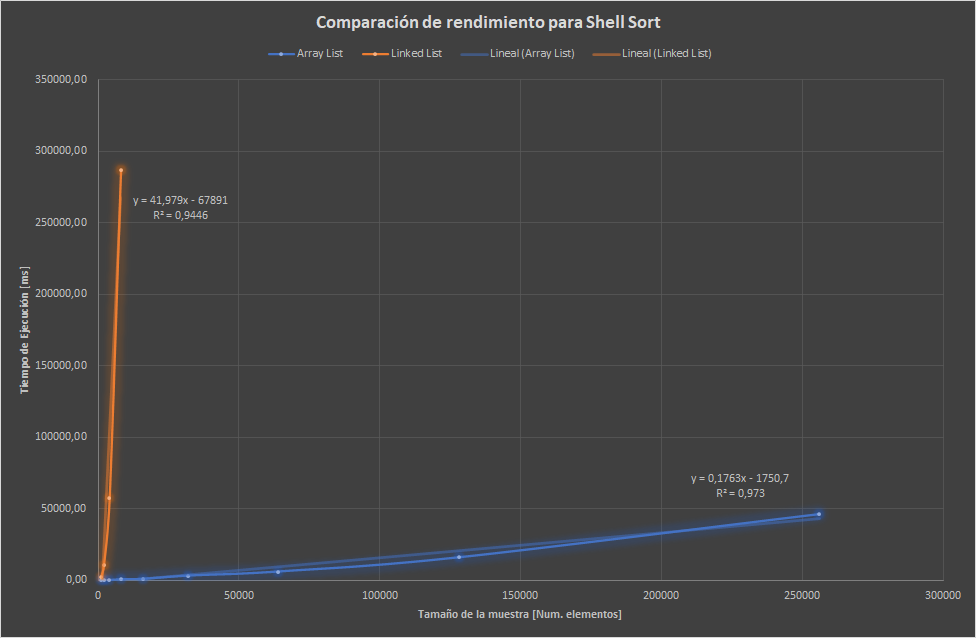
* + Comparación de rendimiento para Insertion Sort.



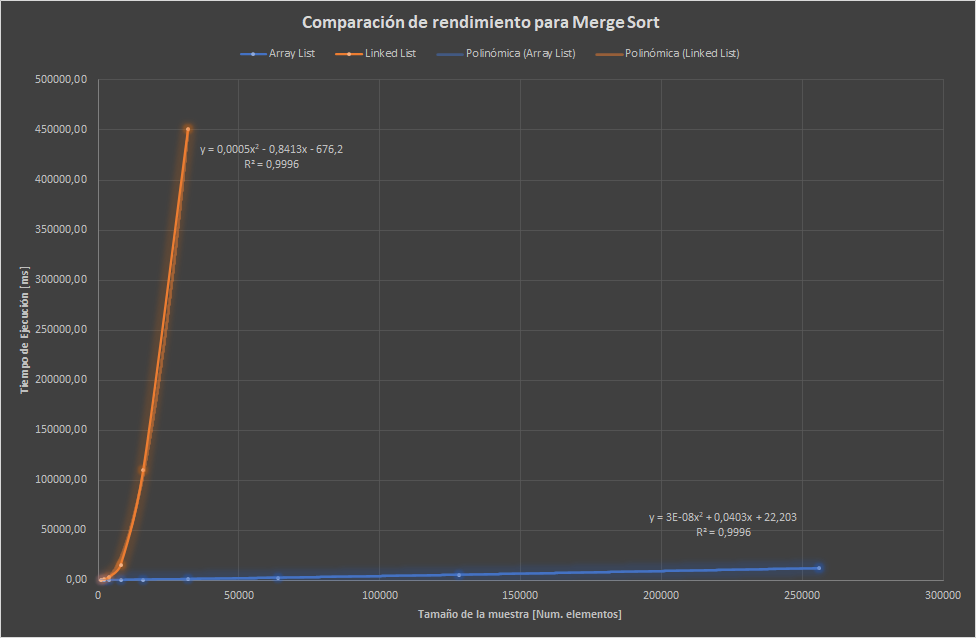
* + Comparación de rendimiento para Selection Sort.



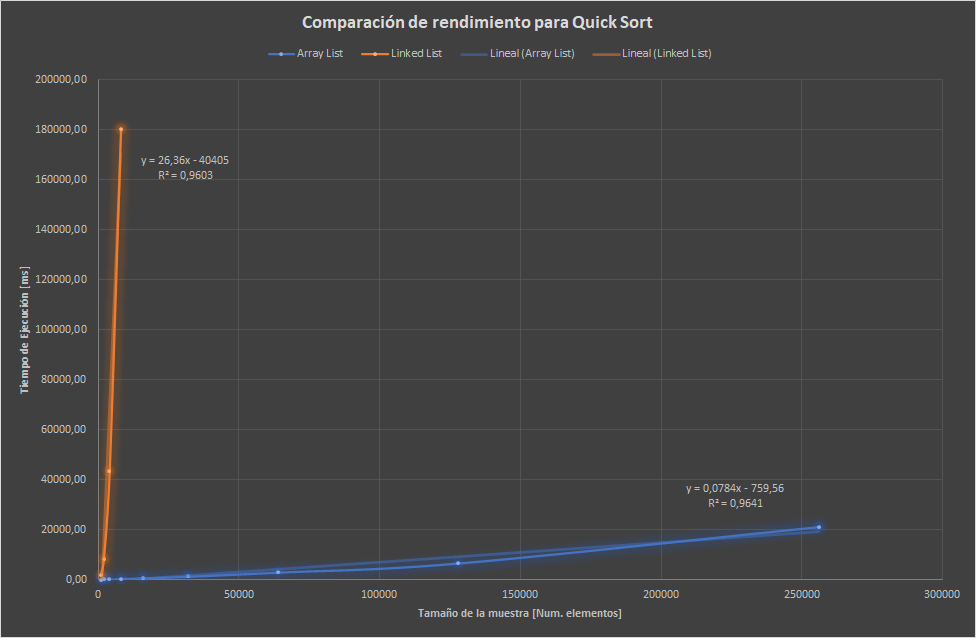
* + Comparación de rendimiento para Shell Sort.



* + 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?

Si se evalúa cada algoritmo individualmente, podemos observar que si coincide con los resultados teóricos. Sin embargo, en las gráficas se ven complejidades diferentes debido a que podemos relacionar el uso de arreglos como el mejor de los casos, o un buen caso, y el uso de las listas enlazadas como los peores casos, por lo que se ve una diferencia enorme, afectando las complejidades de cada uno, y causando discrepancias. Un ejemplo claro de esto es con el MergeSort, ya que su complejidad teórica es NLogN, y en la gráfica se muestra un comportamiento logarítmico y exponencial respectivamente.

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

La única diferencia encontrada fue en un pequeño delay con respecto al tiempo de carga de todos los datos. Sin embargo, esto no tuvo una incidencia significativa dentro de los resultados, evidenciado en el comportamiento de las gráficas y los algoritmos, que son similares.

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

Se tienen dos razones principales

* La memoria RAM de un computador es superior a la otra, por lo que esto se evidenció en que la máquina 1 tuvo un ligeramente menor tiempo de carga que la máquina 2 en la mayoría de los casos.
* Un usuario pudo haber sido más cuidadoso con la toma de datos, al asegurarse que todas las aplicaciones y servicios estuvieran cerrados, con el objetivo de que el computador sólo se enfocara en cargar los datos.

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

Los arreglos son más útiles debido a que cargan una mayor cantidad de datos en menos tiempo, si se compara cada algoritmo con respecto al tipo de estructura, podemos observar que los arreglos siempre tienen una complejidad favorable (lineal, constante, logaritmica, y en el peor de los casos, cuadrada)

1. 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.
2. Merge Sort
3. Quick Sort
4. Shell Sort
5. Insertion Sort
6. Selection Sort