OBSERVACIONES DE LA PRACTICA

Javier Cerino Cod 202020873

Marco Zuliani Cod 202022412

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core (TM)i5-1035G1 CPU @ 1.00GHz | AMD Ryzen 3 3300U with Radeon Vega Mobile Gfx 2.10 GHz |
| Memoria RAM (GB) | 8.0 GB (7.76 GB Utilizable) | 12.0 GB (9.92 GB Utilizable) |
| Sistema Operativo | Windows 10 Home Single Language | Windows 10 Home Single Language |

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 | 762.50 | 953.12 | 31,25 | 31,25 | 31,25 |
| 2000 | 3038.12 | 4109.37 | 78.12 | 78,12 | 78,12 |
| 4000 | 12348.95 | 16312.34 | 218.75 | 168,64 | 140,62 |
| 8000 | 48505.20 | 70187.26 | 531.25 | 367,18 | 312,5 |
| 16000 | 206859.37 | 272296.87 | 1187,5 | 703,12 | 668,75 |
| 32000 | 1180406.25 | 1649125.08 | 2765.62 | 1428,12 | 1421,87 |
| 64000 | 5079984.37 | 5065421.87 | 7059.35 | 3265,62 | 2993,74 |
| 128000 | Límite de tiempo excedido | Límite de tiempo excedido | 16109.37 | 7065,62 | 6525.0 |
| 256000 | Límite de tiempo excedido | Límite de tiempo excedido | 43296.87 | 17337,90 | 13968,74 |
| 512000 | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra |

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 | 41368.27 | 45468.75 | 2000 | 1468,75 | 218,75 |
| 2000 | 376529.25 | 398734.37 | 9562.5 | 8375,0 | 781,25 |
| 4000 | 3286927.5 | 3054406.25 | 44046.87 | 35156,25 | 3328,12 |
| 8000 | Límite de tiempo excedido | Límite de tiempo excedido | 220109.37 | 159062,5 | 12968,75 |
| 16000 | Límite de tiempo excedido | Límite de tiempo excedido | 989375.0 | 568718,75 | 52078,12 |
| 32000 | Límite de tiempo excedido | Límite de tiempo excedido | 4870421.87 | 2595703,12 | 203625,0 |
| 64000 | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | 859093,75 |
| 128000 | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido |
| 256000 | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido |
| 512000 | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra |

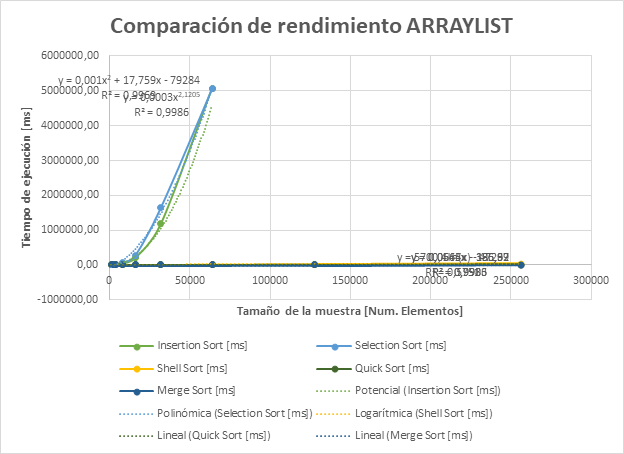
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

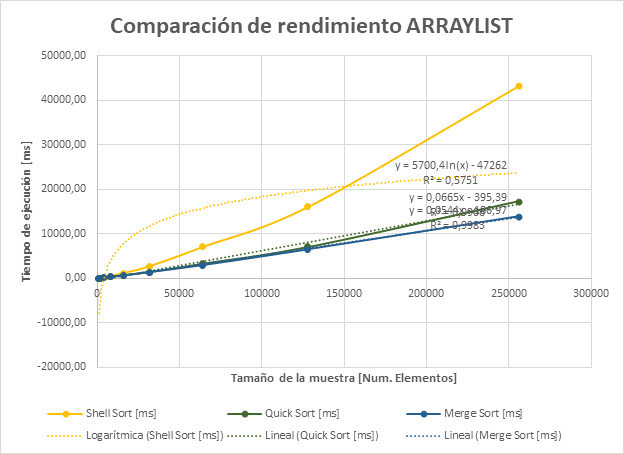
|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | X |  |
| Quick sort | X |  |

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

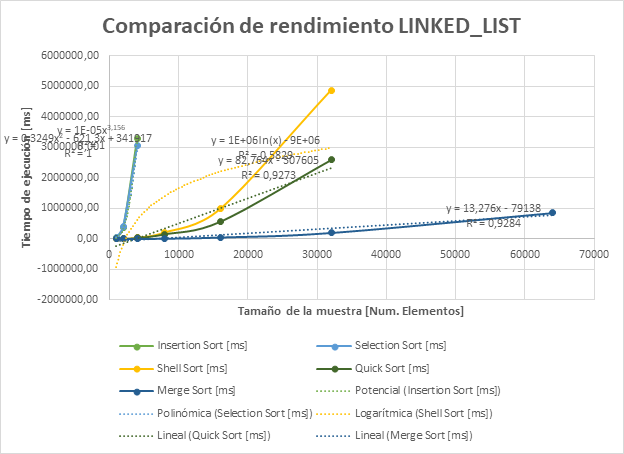
## **Graficas**

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

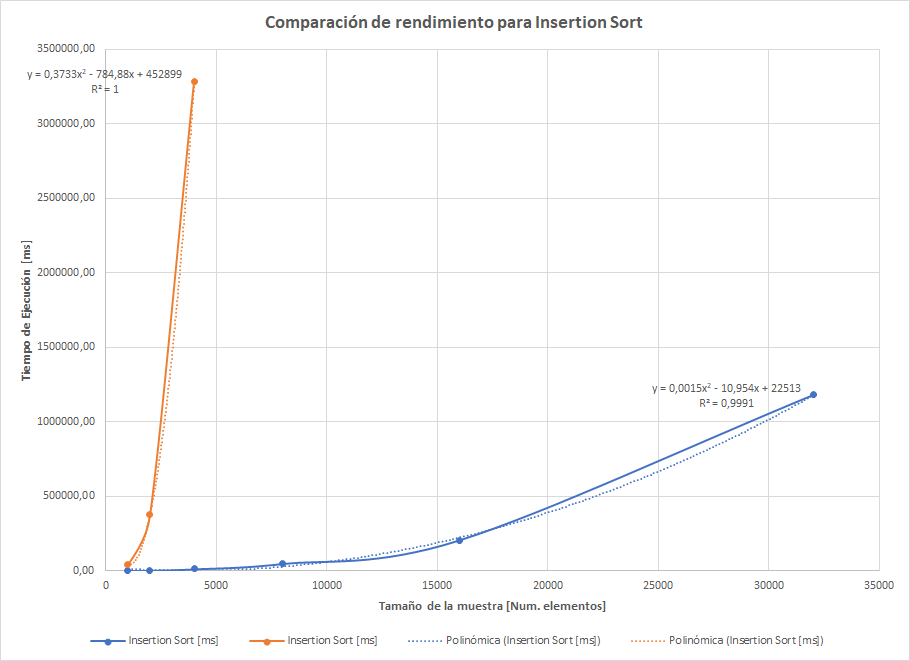




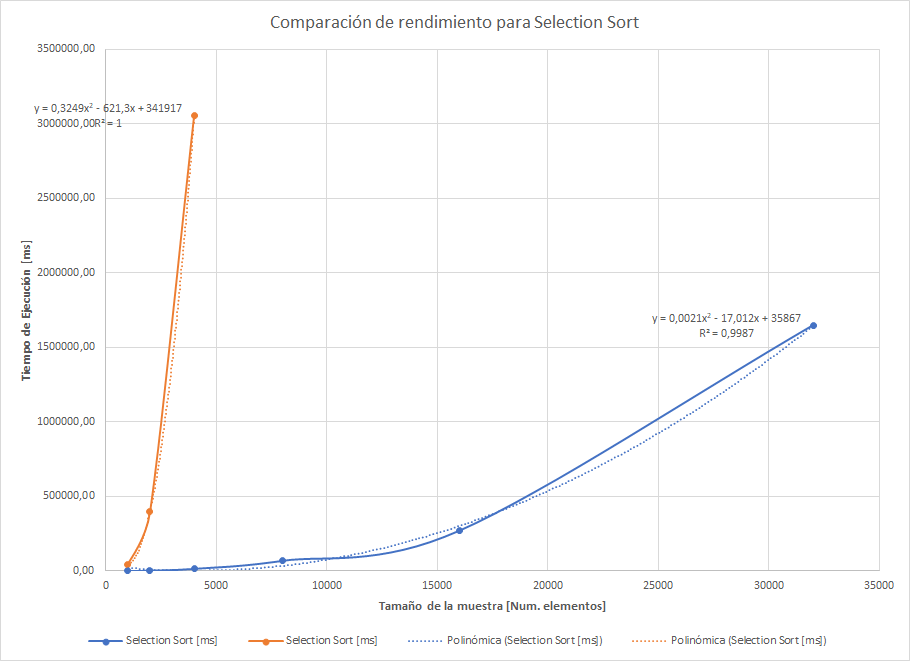
* + Comparación de rendimiento LINKED\_LIST.



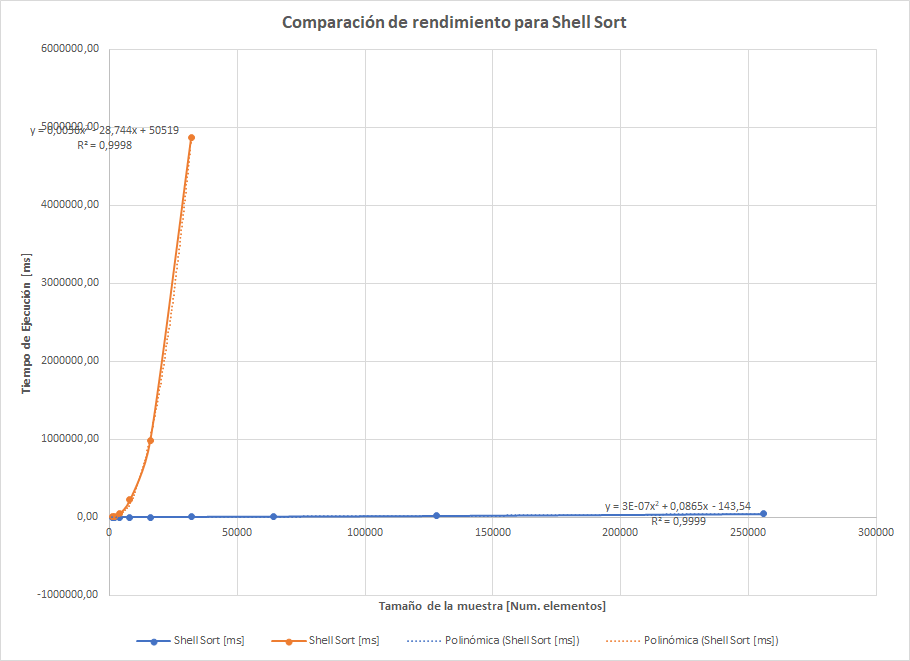
* + Comparación de rendimiento para Insertion Sort.



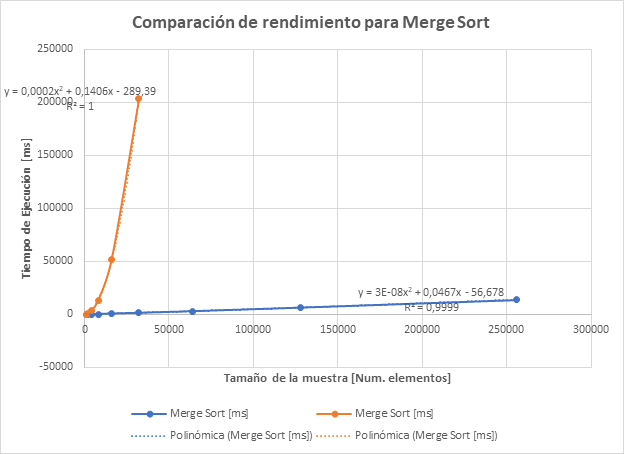
* + Comparación de rendimiento para Selection Sort.



* + Comparación de rendimiento para Shell Sort.



* + Comparación de rendimiento para MergeSort.



* + Comparación de rendimiento para Quick Sort.



# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Selection Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 1000 | 790,62 | 921,87 | 46,87 | 46,87 | 46,87 |
| 2000 | 3168,75 | 3703,12 | 109,37 | 109,37 | 109,37 |
| 4000 | 13057,29 | 14796,87 | 234,37 | 218,75 | 208,3 |
| 8000 | 53348,95 | 64109,37 | 593,75 | 515,62 | 453,12 |
| 16000 | 229156,25 | 269843,75 | 1265,62 | 942,7 | 1010,41 |
| 32000 | 914734,37 | 1639343,75 | 2984,37 | 2098,95 | 2161,45 |
| 64000 | 4346218,75 | 4762843,75 | 7125,0 | 4765,62 | 4640,62 |
| 128000 | Límite de tiempo excedido | Límite de tiempo excedido | 17187,5 | 10265,62 | 9885,41 |
| 256000 | Límite de tiempo excedido | Límite de tiempo excedido | 44218,75 | 24156,24 | 21244,79 |
| 512000 | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra |

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 | 51843,75 | 56343,75 | 3875,0 | 1838,54 | 255.20 |
| 2000 | 421015,62 | 449390,62 | 17265,62 | 9264.62 | 1031.24 |
| 4000 | 4110875,0 | 3265046,87 | 83531,25 | 43046.87 | 4026.04 |
| 8000 | Límite de tiempo excedido | Límite de tiempo excedido | 467437,5 | 197062.5 | 16317.70 |
| 16000 | Límite de tiempo excedido | Límite de tiempo excedido | 2745734,37 | 688500.5 | 65968.75 |
| 32000 | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | 3439140,63 | 268593.75 |
| 64000 | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | 1093687.5 |
| 128000 | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido |
| 256000 | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido | Límite de tiempo excedido |
| 512000 | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra | No hay suficientes datos para la muestra |

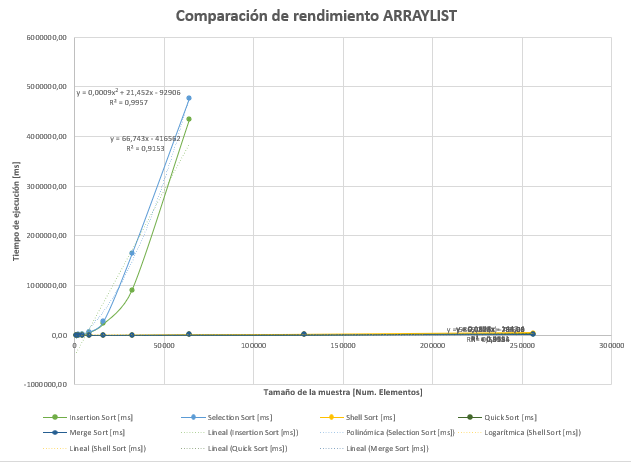
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | X |  |
| Quick sort | X |  |

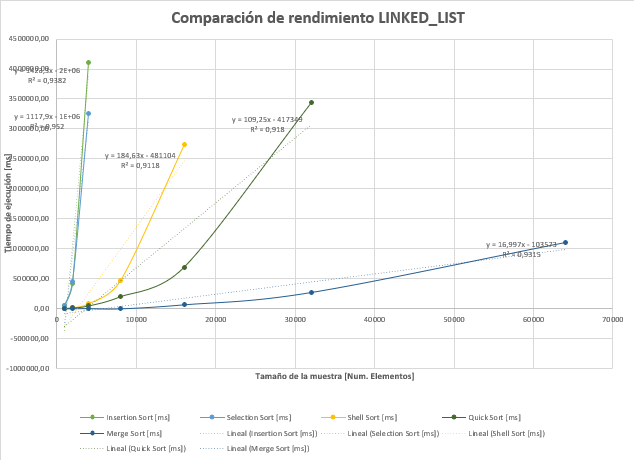
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

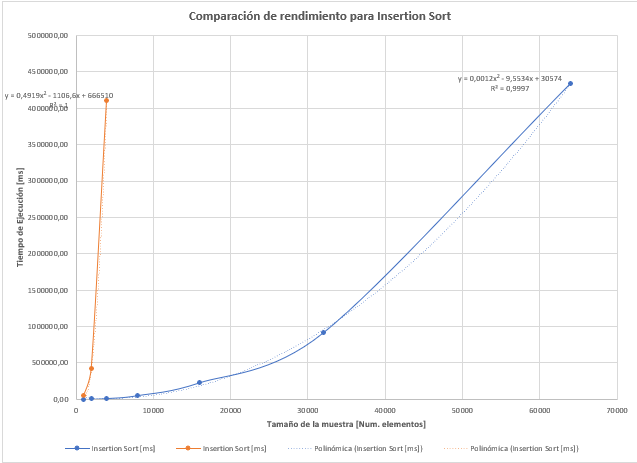
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.



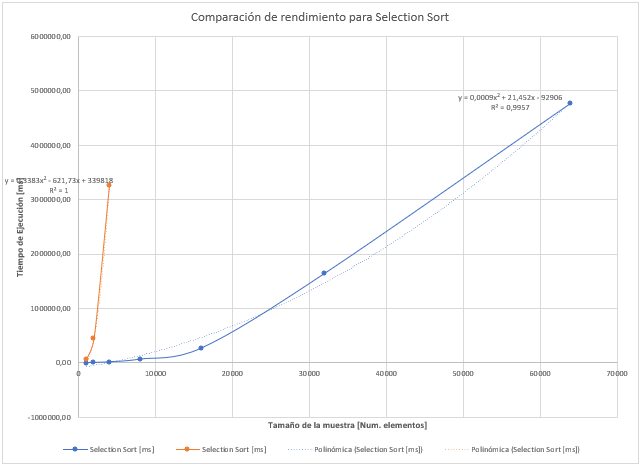
* + Comparación de rendimiento LINKED\_LIST.



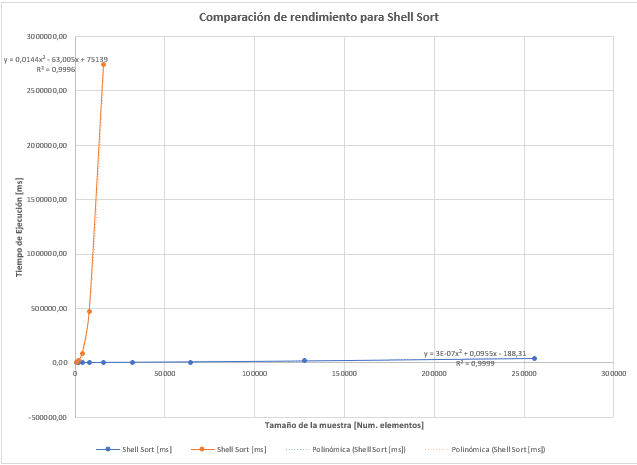
* + Comparación de rendimiento para Insertion Sort.



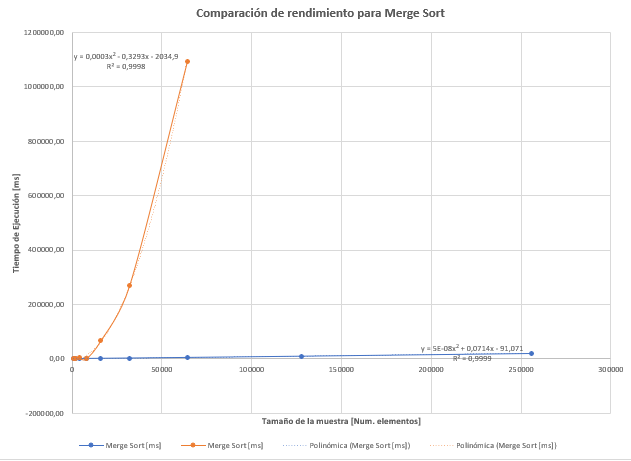
* + Comparación de rendimiento para Selection Sort.



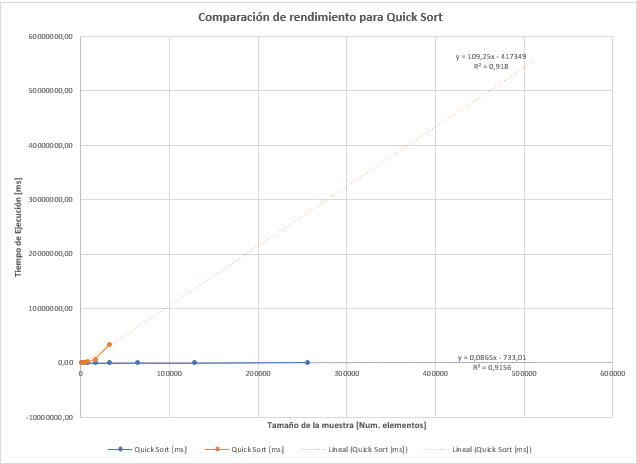
* + Comparación de rendimiento para Shell Sort.



* + Comparación de rendimiento para Merge Sort.



* + Comparación de rendimiento para Quick Sort.



# **Preguntas de análisis**

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

Si, el comportamiento de los algoritmos es acorde con lo enunciado teóricamente. Esto se puede observar en las tablas de datos donde se realiza la comparación entre los tipos de ordenamientos y la cantidad de datos a ordenar. La eficiencia de los códigos de ordenamiento Quick y Merge, al momento de ordenar pocas o grandes cantidades de datos, tienen un tiempo relativamente menor en comparación a los otros tipos de ordenamiento donde el tiempo aumenta en gran medida con el aumento de los datos. Por lo tanto se entiende perfectamente porqué ordenamientos recursivos como Quick y Merge, debido a las operaciones que realizan y al como las realizan, son los tipos de ordenamiento más rápidos.

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

Sí, existen varias diferencias entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas ya que la segunda necesita de un mayor tiempo para realizar el mismo procedimiento que la primera.

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

Estas diferencias se deben a que cada máquina tiene características bien específicas, como memoria RAM y procesador, que le dan una mayor o menor capacidad al momento de realizar ciertas acciones. Por lo tanto, analizando las características de las maquinas utilizadas (ver tabla 1) y los resultados obtenidos sobre los tiempos de ejecución se puede evidenciar que la máquina que tiene un mejor procesador utiliza un menor tiempo para realizar los ordenamientos. Además, se puede resaltar que aunque la maquina 2 tiene una mayor cantidad de memoria RAM utilizable, esto no influye en gran medida sobre el tiempo utilizado al momento de ordenar los datos.

1. **¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?**La Estructura de Datos que es mejor utilizar si solo se tienen en cuenta los tiempos de ejecución es el Arraylist. Además, para ordenar este tipo de estructura es evidente que el tiempo es menor al momento de utilizar el ordenamiento recursivo Merge, por lo que estos dos son una gran combinación.
2. **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.**

El ranking sería:

1. *Merge sort* en primer lugar con unos tiempos de ejecución un tanto mejores que los del segundo lugar.
2. *Quick sort* en segundo lugar con unos tiempos de ejecución bastante mejores del tercer lugar.
3. *Shell sort* en tercer lugar con unos tiempos eficientes, pero no siendo rivales para los algoritmos de ordenamiento recursivo.
4. *Insertion sort* en cuarto lugar con una diferencia bastante grande con el tercer puesto.
5. *Selection sort* en último lugar con una muy poca diferencia junto con el cuarto puesto.