OBSERVACIONES DE LA PRACTICA

Lindsay Vanessa Pinto Morato Cod 202023138

Maicol Yojan Antonio Rincón Cod 202027329

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10GHZ | Intel Pentium Quad Core Processor N4200 1.10 GHz |
| Memoria RAM (GB) | 8,00GB (6.94 utilizable) | 4,00GB (3.84 Utilizable) |
| Sistema Operativo | Windows 10 Home Single languaje | Windows 10 Pro |

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 | 3531,25 | 3500,00 | 3312,50 | 2640,62 | 2609,37 |
| 2000 | 17718,75 | 17937,50 | 17953,12 | 12531,2 | 12109,37 |
| 4000 | 84328,12 | 83093,75 | 84843,75 | 50937,5 | 47281,25 |
| 8000 | 371156,25 | 370265,62 | 370265,62 | 267562,5 | 227187,5 |
| 16000 | 1221328,12 | 1240093,75 | 1242968,75 | 988531,25 | 984781,25 |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 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 |  |  |  |  |  |

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 | 3531,25 | 3437,50 | 3515,62 | 2828,12 | 2656,25 |
| 2000 | 19250,50 | 18609,37 | 18484,37 | 12500,0 | 12453,12 |
| 4000 | 87343,75 | 88640,62 | 87937,50 | 54500,0 | 50756,62 |
| 8000 | 277406,25 | 272875,00 | 277484,37 | 273750,0 | 227406,25 |
| 16000 | 1508234,37 | 1470437,50 | 1352640,62 | 1195687,5 | 993609,37 |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 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 |

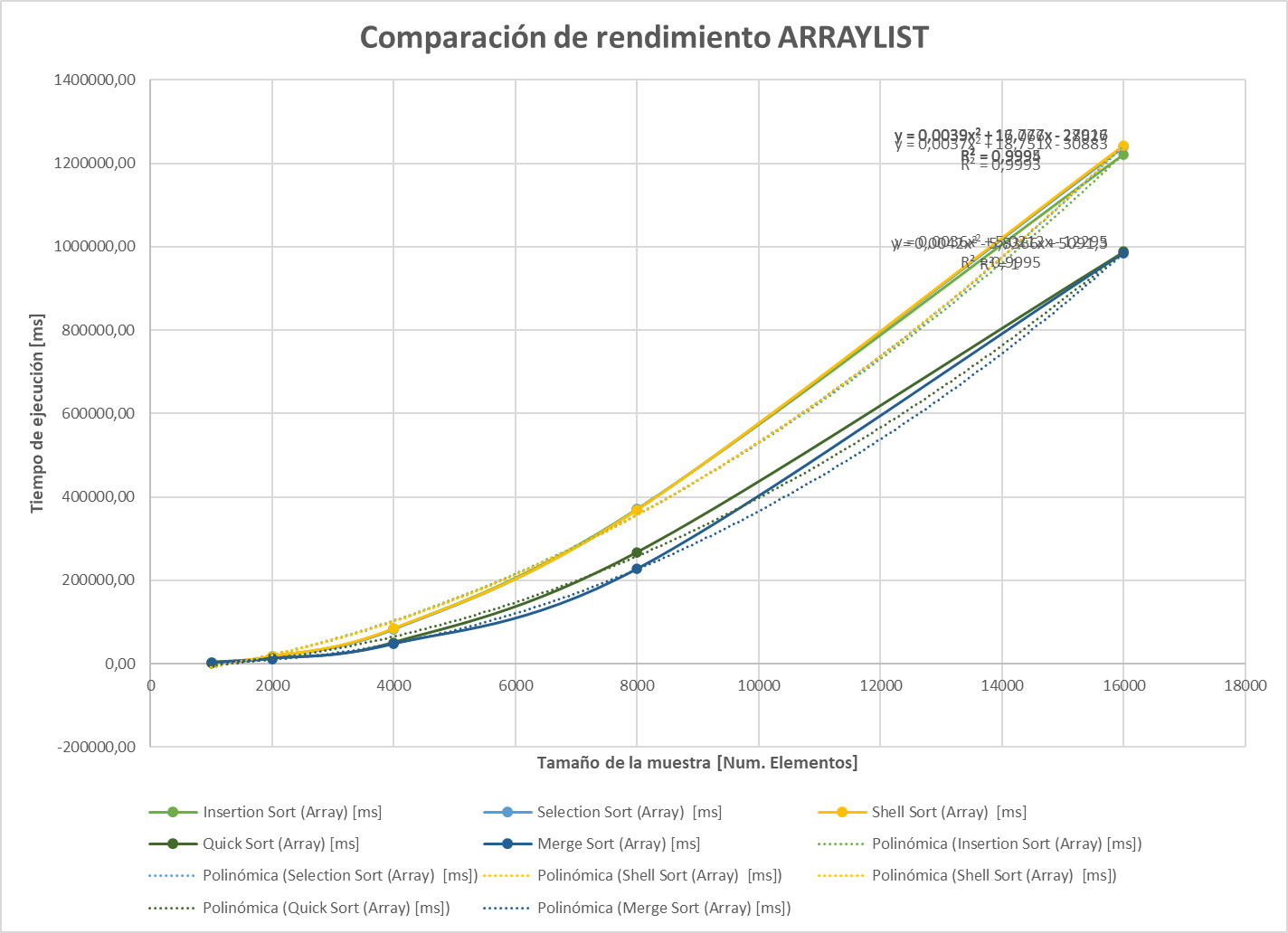
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | Presenta tiempos de ejecución menores, aunque similares al Linked\_list. | Los tiempos de ejecución son ligeramente mayores a los del Arraylist. |
| Quick sort | El comportamiento es mejor que en el linked\_list y esta diferencia se acentúa a medida que crece el tamaño de los datos. | El comportamiento es significativamente menor que con el Arraylist pues los tiempos de ejecución aumentan significativamente a medida que aumenta el tamaño de los datos. |

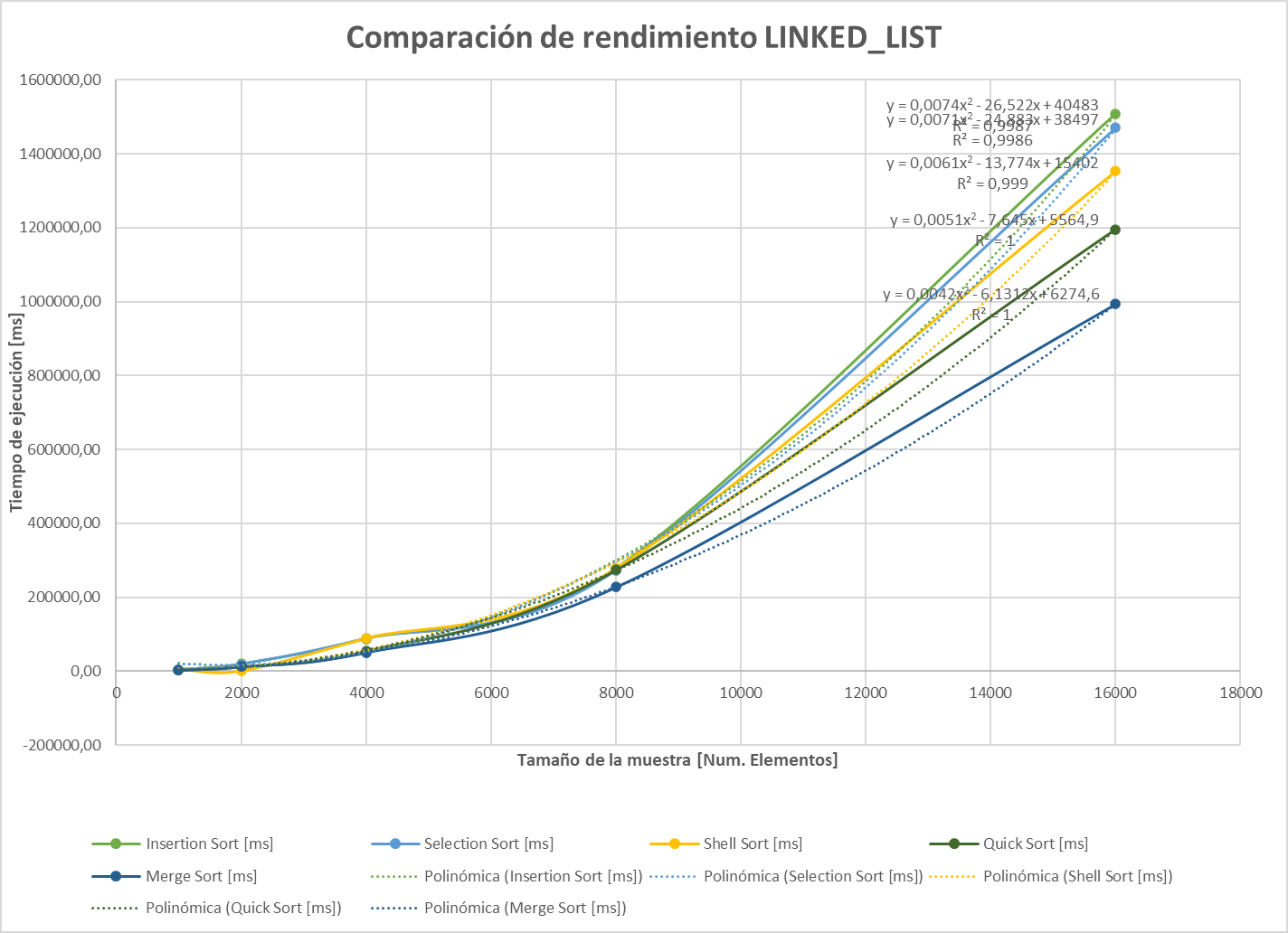
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

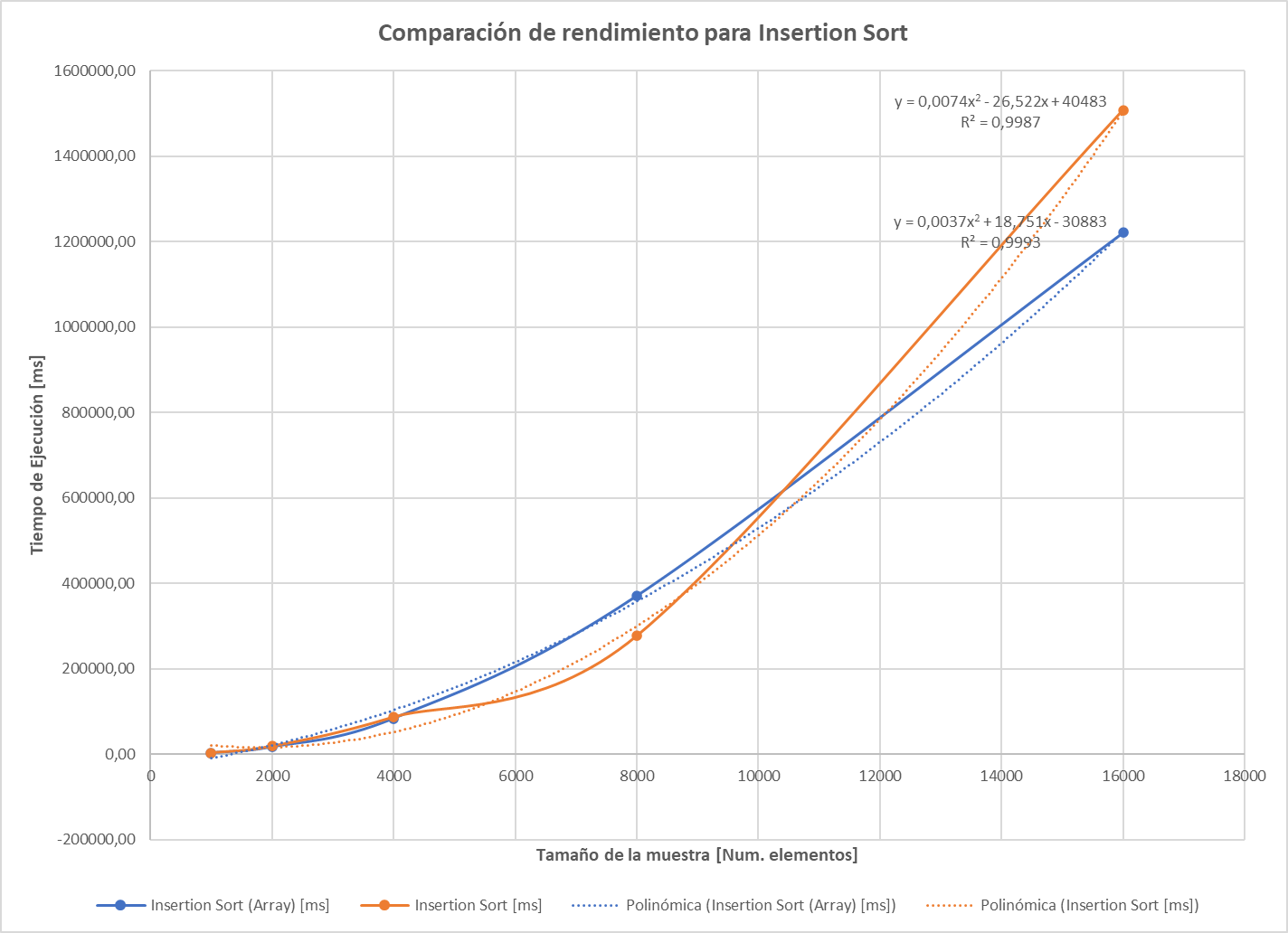
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.



* + Comparación de rendimiento LINKED\_LIST.



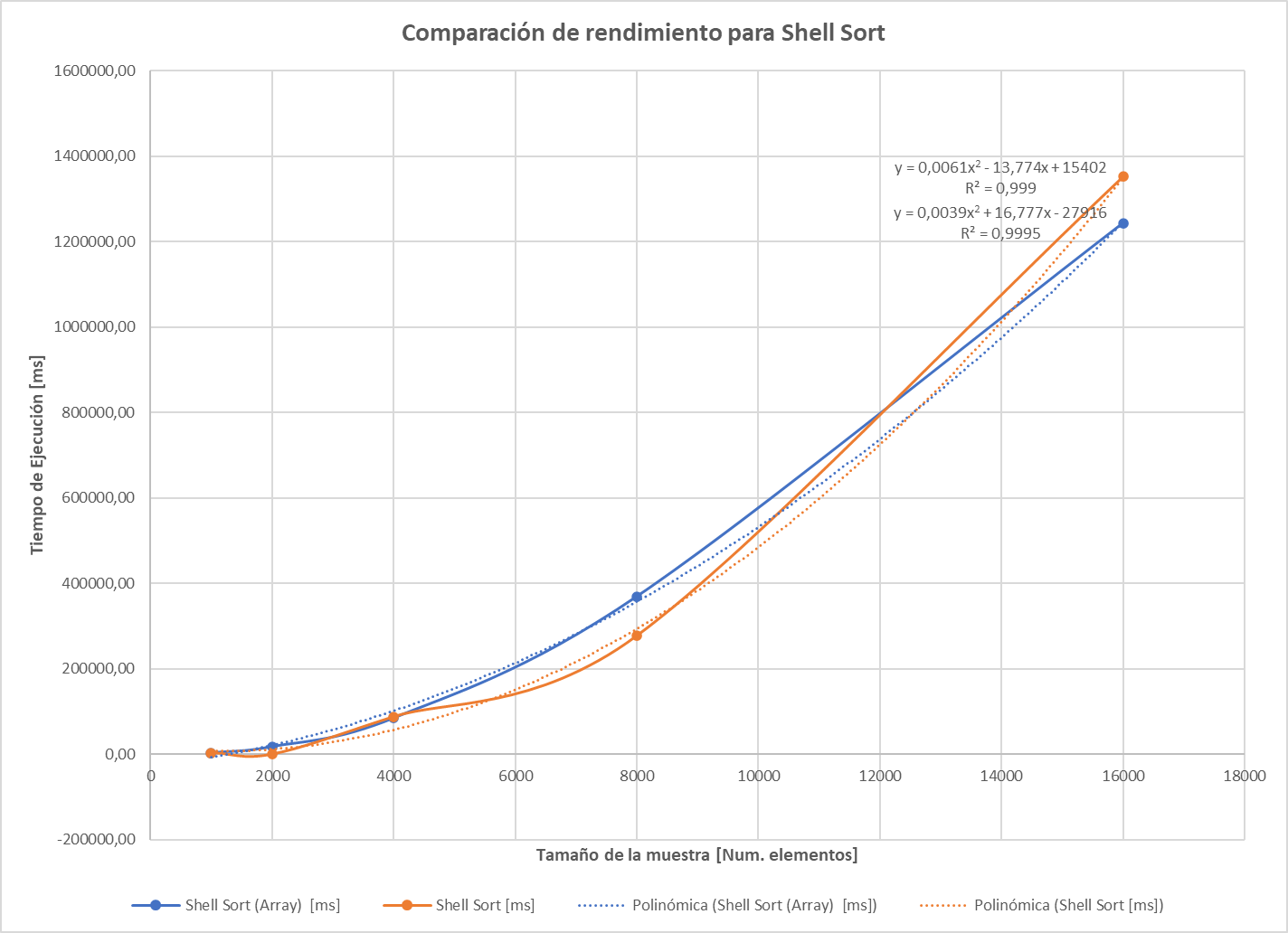
* + Comparación de rendimiento para Insertion Sort.



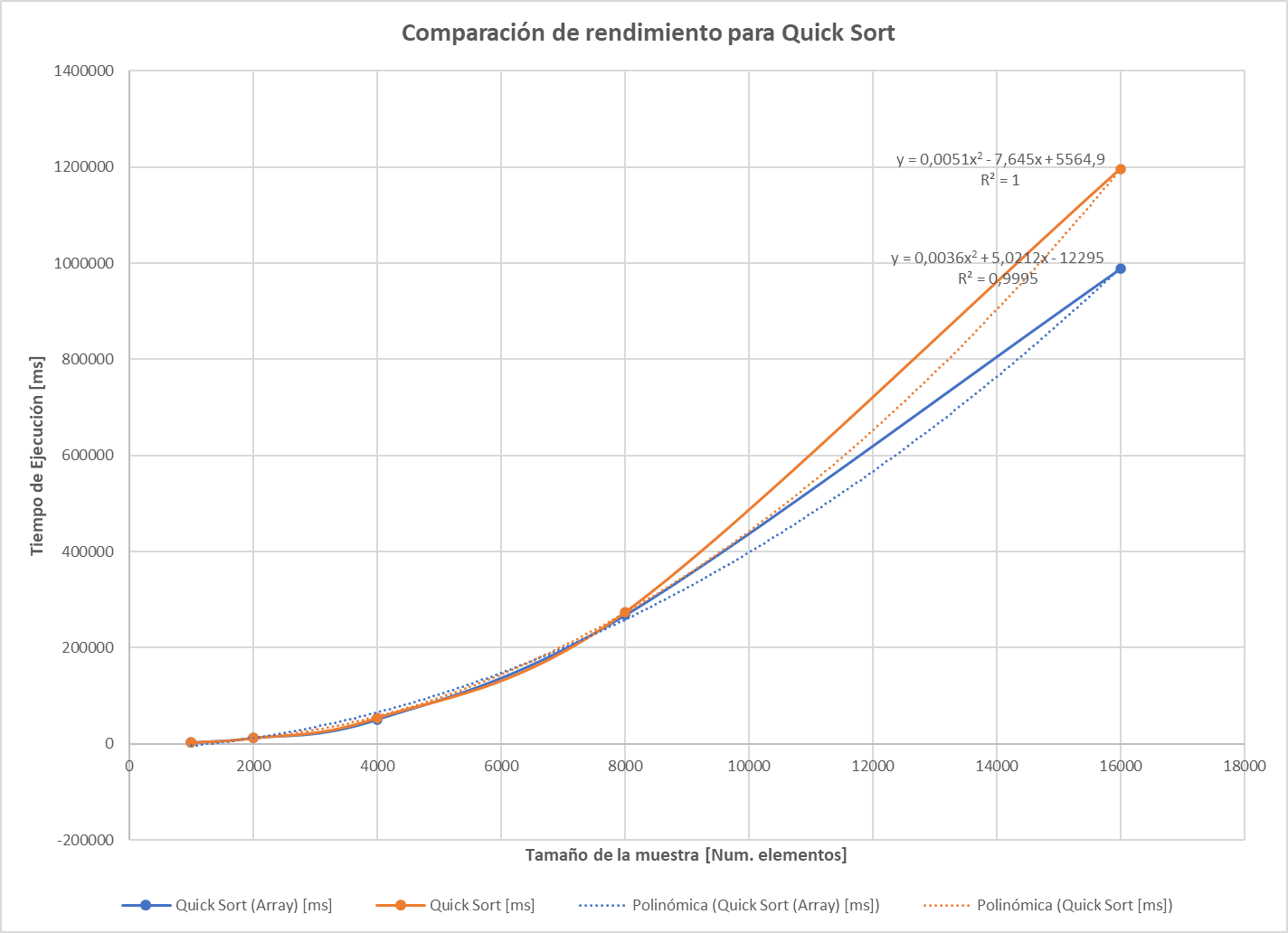
* + Comparación de rendimiento para Selection Sort.



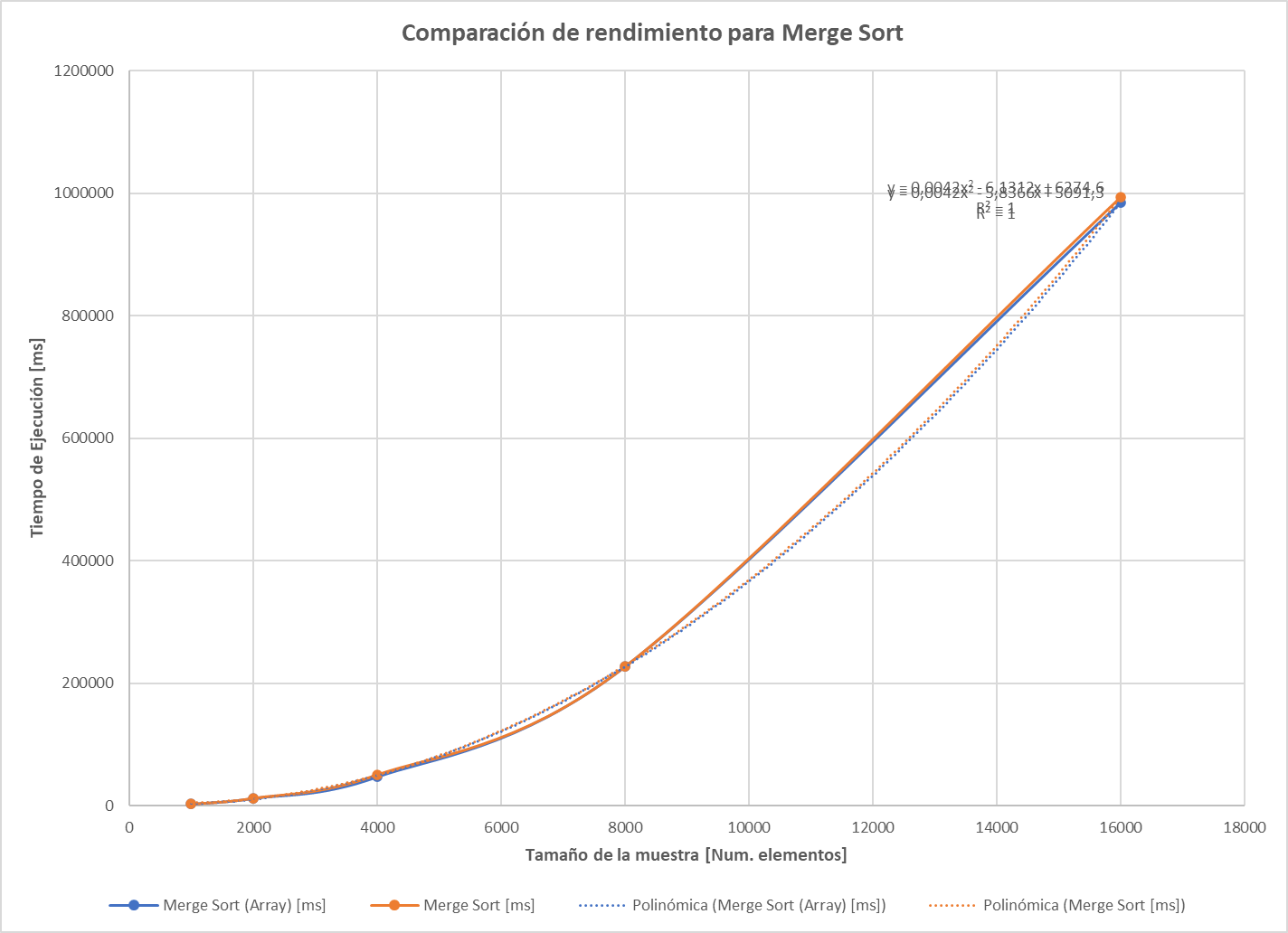
* + Comparación de rendimiento para Shell Sort.



* + 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 |  |  |  |  |  |
| 2000 |  |  |  |  |  |
| 4000 |  |  |  |  |  |
| 8000 |  |  |  |  |  |
| 16000 |  |  |  |  |  |
| 32000 |  |  |  |  |  |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

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 |  |  |  |  |  |
| 2000 |  |  |  |  |  |
| 4000 |  |  |  |  |  |
| 8000 |  |  |  |  |  |
| 16000 |  |  |  |  |  |
| 32000 |  |  |  |  |  |
| 64000 |  |  |  |  |  |
| 128000 |  |  |  |  |  |
| 256000 |  |  |  |  |  |
| 512000 |  |  |  |  |  |

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 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?

Teóricamente se explica que los tiempos de ejecución entre el Inserion Sort, Selection Sort y Shell Sort deberían ser mejores para este último, sin embargo, en las pruebas se vieron tiempos de ejecución muy similares para estos 3 algoritmos. Por otro lado, aunque el comportamiento del Quick Sort en el peor de los casos es de O(n^2) y para el Merge Sort de O(n log (n)), el promedio para ambos debería ser de O(n log (n)), no obstante, las pruebas realizadas siempre arrojaron tiempos menores (en promedio) para el Merge Sort tanto en Array List como en Linked List.

1. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
2. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?
3. ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

Si se tienen en cuenta únicamente los tiempos de ejecución, se prefiere el Array List sobre el Linked List pues presenta tiempos menores de ejecución en todas sus pruebas.

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. Selection Sort
6. Insertion Sort