OBSERVACIONES DE LA PRACTICA

Felipe Garzón 202021161

Pablo Ortega 202021700

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core (TM) i5-8265 CPU @  1.60 GHz 1.80GHz | 1,6 GHz Dual-Core Intel Core i5 |
| Memoria RAM (GB) | 12.0 GB | 8.0GB |
| Sistema Operativo | Windows 10 Pro | Mac 10.15.6 |

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

# **Maquina 1**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 421.875 | 453.125 | 31.25 |
| 2000 | 1671.875 | 1812.5 | 46.875 |
| 4000 | 7171.875 | 7609.375 | 156.25 |
| 8000 | 29375.0 | 30234.375 | 312.5 |
| 16000 | 124765.625 | 142656.25 | 718.75 |
| 32000 | 537625.0 | 601390.625 | 1718.75 |
| 64000 | Tiempo excedido | Tiempo excedido | 3859.375 |
| 128000 | Tiempo excedido | Tiempo excedido | 9718.75 |
| 256000 | Tiempo excedido | Tiempo excedido | 22937.5 |
| 512000 | N/A | N/A | N/A |

*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) |
| 1000 | 437.5 | 484.375 | 31.25 |
| 2000 | 1890.625 | 1953.125 | 62.5 |
| 4000 | 7906.25 | 8765.625 | 140.625 |
| 8000 | 34546.875 | 39437.5 | 312.5 |
| 16000 | 121093.75 | 168406.25 | 750.0 |
| 32000 | 630671.875 | Tiempo excedido | 1765.625 |
| 64000 | Tiempo excedido | Tiempo excedido | 4187.5 |
| 128000 | Tiempo excedido | Tiempo excedido | 10515.625 |
| 256000 | Tiempo excedido | Tiempo excedido | 32687.5 |
| 512000 | N/A | N/A | N/A |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Insertion sort | x |  |
| Selection sort | x |  |
| Shell 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.

# **Maquina 2**

## **Resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (ARRAY\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 522,61 | 559,32 | 44,14 |
| 2000 | 2174,08 | 2514,22 | 94,56 |
| 4000 | 8703,96 | 10385,94 | 173,05 |
| 8000 | 37283,29 | 44881,91 | 388,39 |
| 16000 | 171431,50 | 210160,17 | 906,73 |
| 32000 | 741596,15 | 839912,27 | 2356,08 |
| 64000 | Tiempo excedido | Tiempo excedido | 5227,87 |
| 128000 | Tiempo excedido | Tiempo excedido | 12726,48 |
| 256000 | Tiempo excedido | Tiempo excedido | 30423,18 |
| 512000 | N/A | N/A | N/A |

*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) |
| 1000 | 525,25 | 563,09 | 44,61 |
| 2000 | 2261,00 | 2648,87 | 89,31 |
| 4000 | 9118,87 | 11765,56 | 177,84 |
| 8000 | 39659,20 | 45778,91 | 386,24 |
| 16000 | 174429,62 | 204057,46 | 956,62 |
| 32000 | 726944,88 | 839008,08 | 2328,25 |
| 64000 | Tiempo excedido | Tiempo excedido | 5073,82 |
| 128000 | Tiempo excedido | Tiempo excedido | 14849,55 |
| 256000 | Tiempo excedido | Tiempo excedido | 33834,54 |
| 512000 | N/A | N/A | N/A |

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

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Insertion sort | x |  |
| Selection sort | x |  |
| Shell sort | x |  |

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

## **Graficas**

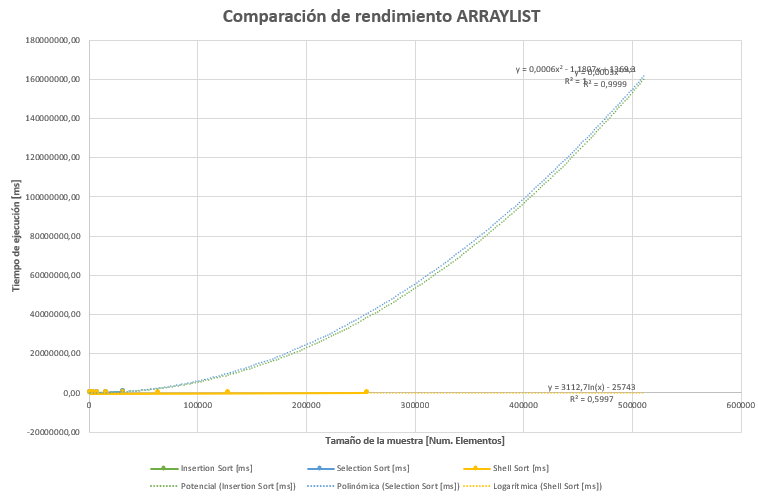
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.
  + Comparación de rendimiento LINKED\_LIST.
  + Comparación de rendimiento para Insertion Sort.
  + Comparación de rendimiento para Selection Sort.
  + Comparación de rendimiento para Shell Sort.

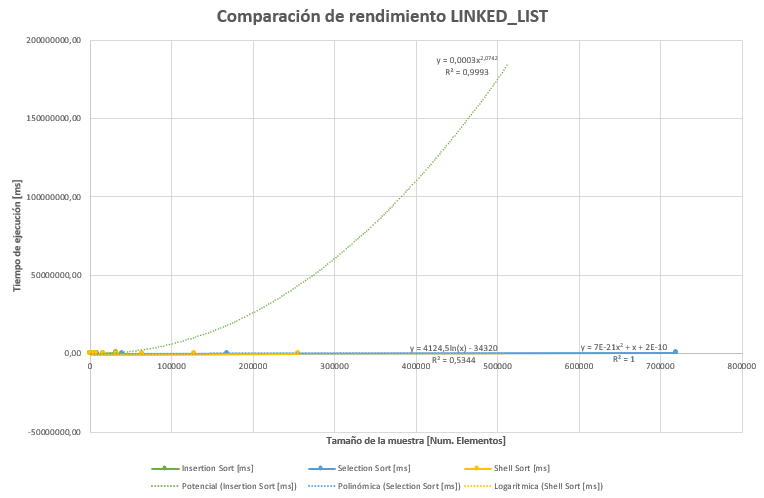
# **Preguntas de análisis**

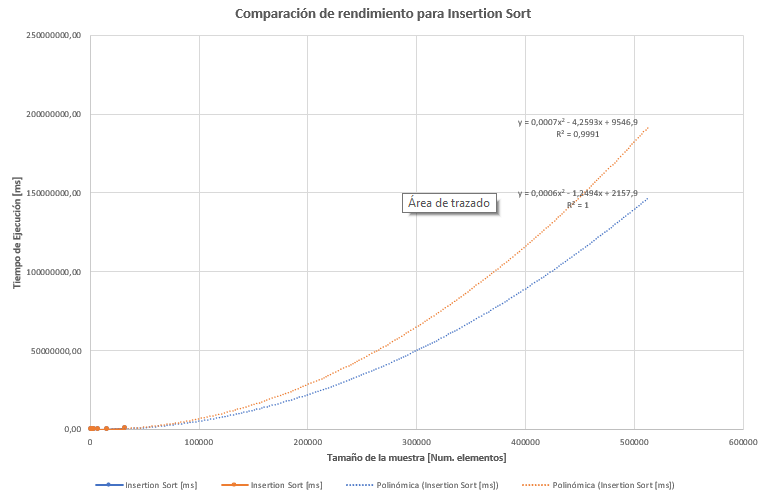
1. ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?
   * Efectivamente se demuestra para los dos casos que tanto teóricamente como prácticamente el arraylist es más eficiente en cuanto al tiempo y al rendimiento que el linked-list.
2. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?
   * En promedio, la maquina 1 logró organizar los datos levemente más rápido que la máquina 2.
3. De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?
   * Estas diferencias se deben a que la maquina 1 tiene un ligeramente mejor procesador y más capacidad en la memoria RAM.
4. ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?
   * Tomando en cuenta únicamente los tiempos de ejecución de los algoritmos se podría decir que la mejor estructura de datos es ARRAYLIST y en esta el mejor algoritmo en cuanto eficiencia de tiempo sería Shell sort.

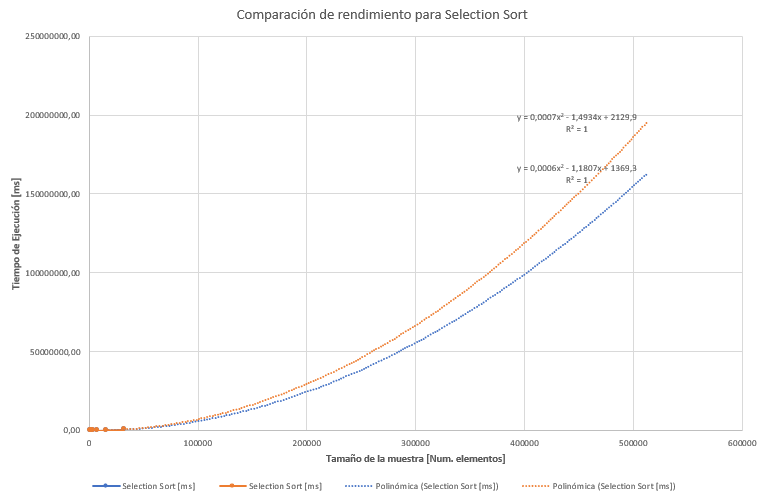
Gráficas:

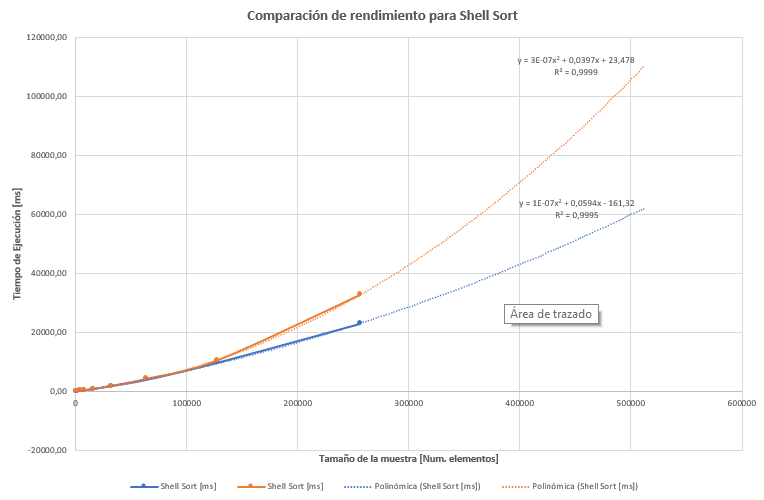
**Maquina 1**











**Maquina 2**

