OBSERVACIONES DE LA PRACTICA

María Paula Alméciga Moreno 202023369

Andrés Felipe Vargas Cuadros 202013817

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD A10-8700P Radeon R6 | AMD Ryzen 5 3500U |
| Memoria RAM (GB) | 16 GB | 8 GB |
| Sistema Operativo | Windows 8.1 Pro | Windows 10 |

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 | 2000,00 | 1942,71 | 109375,00 |
| 2000 | 8609375,00 | 8593,75 | 296875,00 |
| 4000 | 34468,75 | 35437,50 | 609375,00 |
| 8000 | 133156,25 | 137406,25 | 1312,50 |
| 16000 | 535484,38 | 555296,88 | 3125,00 |
| 32000 | Excede tiempo | Excede tiempo | 7750,00 |
| 64000 | Excede tiempo | Excede tiempo | 17953,13 |
| 128000 | Excede tiempo | Excede tiempo | 41812,50 |
| 256000 | Excede tiempo | Excede tiempo | Excede tiempo |
| 512000 | Excede tiempo | Excede tiempo | Excede tiempo |

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 | 144296,88 | 131500,00 | 7265,63 |
| 2000 | 1169312,50 | 1007343,75 | 35937,50 |
| 4000 | Excede tiempo | Excede tiempo | 169437,50 |
| 8000 | Excede tiempo | Excede tiempo | Excede tiempo |
| 16000 | Excede tiempo | Excede tiempo | Excede tiempo |
| 32000 | Excede tiempo | Excede tiempo | Excede tiempo |
| 64000 | Excede tiempo | Excede tiempo | Excede tiempo |
| 128000 | Excede tiempo | Excede tiempo | Excede tiempo |
| 256000 | Excede tiempo | Excede tiempo | Excede tiempo |
| 512000 | Excede tiempo | Excede tiempo | Excede tiempo |

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

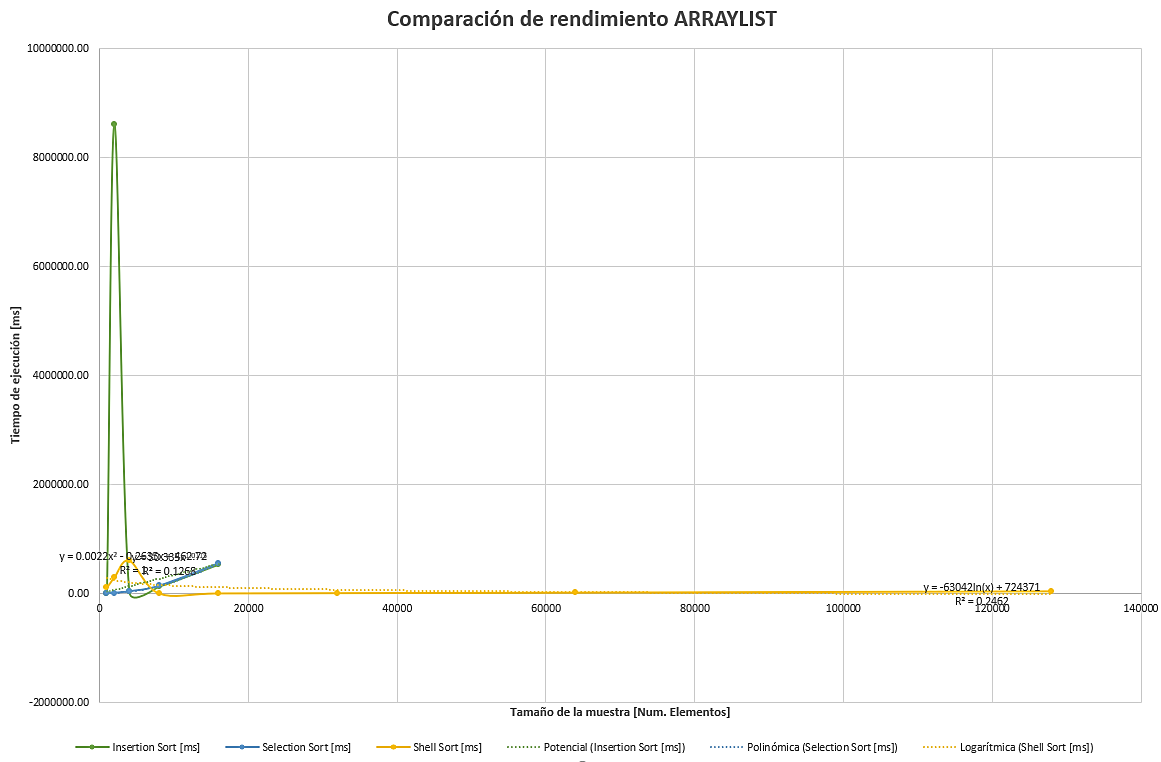
|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Insertion sort |  |  |
| Selection sort |  |  |
| Shell sort | x | x |

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

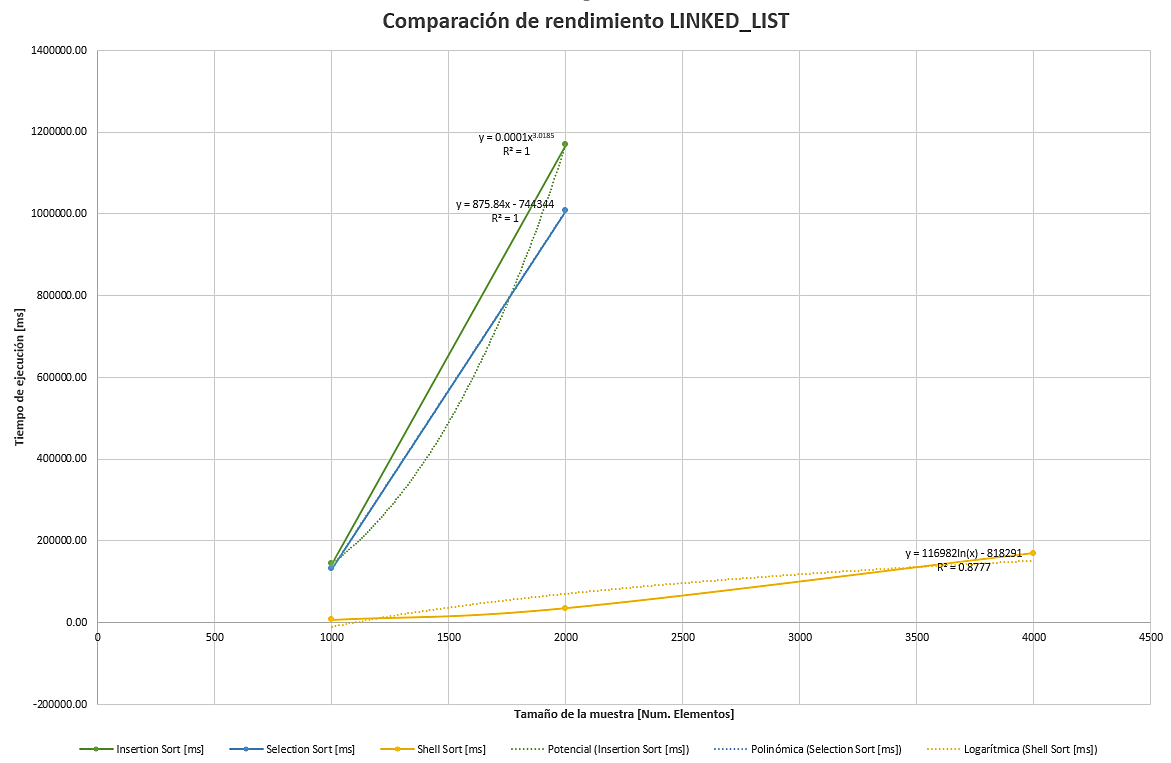
Se ha marcado en la anterior tabla, para cada tipo de estructura de datos, qué tipo de ordenamiento iterativo fue más eficiente en las pruebas. Según los datos, para ambas estructuras fue el Shell Sort el ordenamiento más eficiente, mostrando tiempos significativamente menores al de los demás ordenamientos y, por tanto, alcanzando el tiempo excedido para tamaños de muestra mucho mayores. Además, se ha determinado que el arreglo (array\_list) es el tipo de estructura de datos que exhibió tiempos de ejecución menores, resultando ser también la mejor opción entre las dos posibles.

## **Graficas**

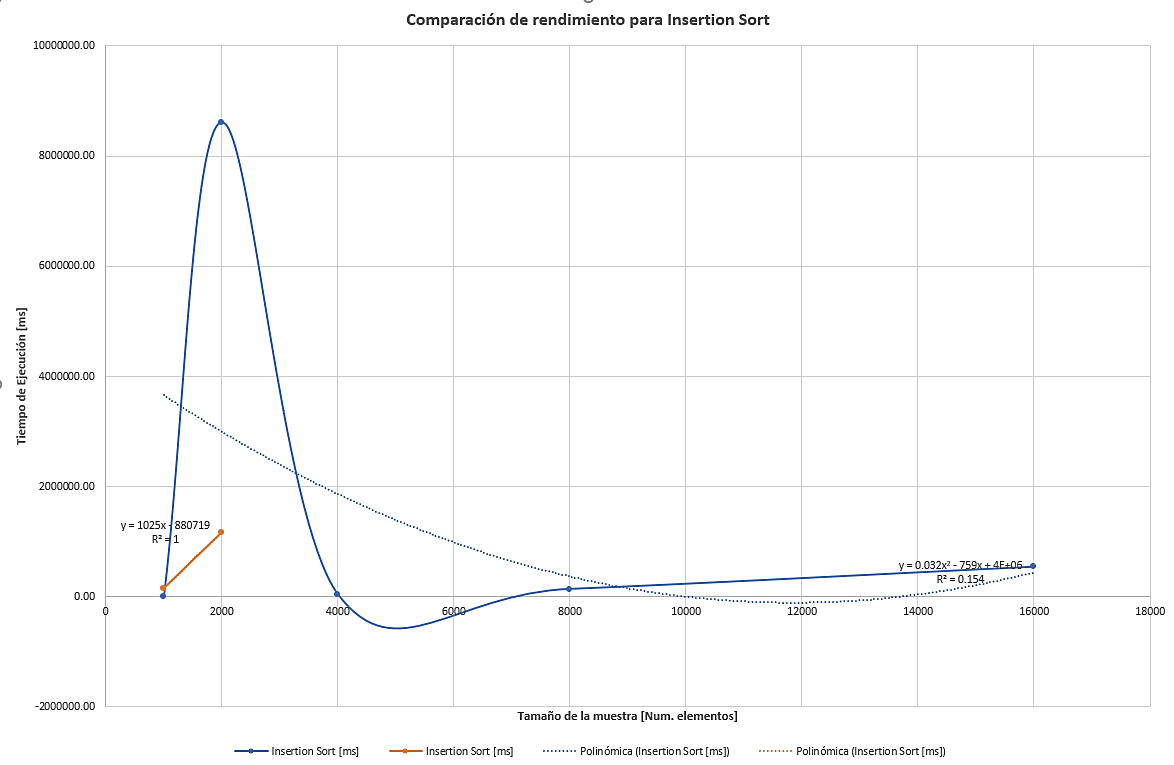
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.



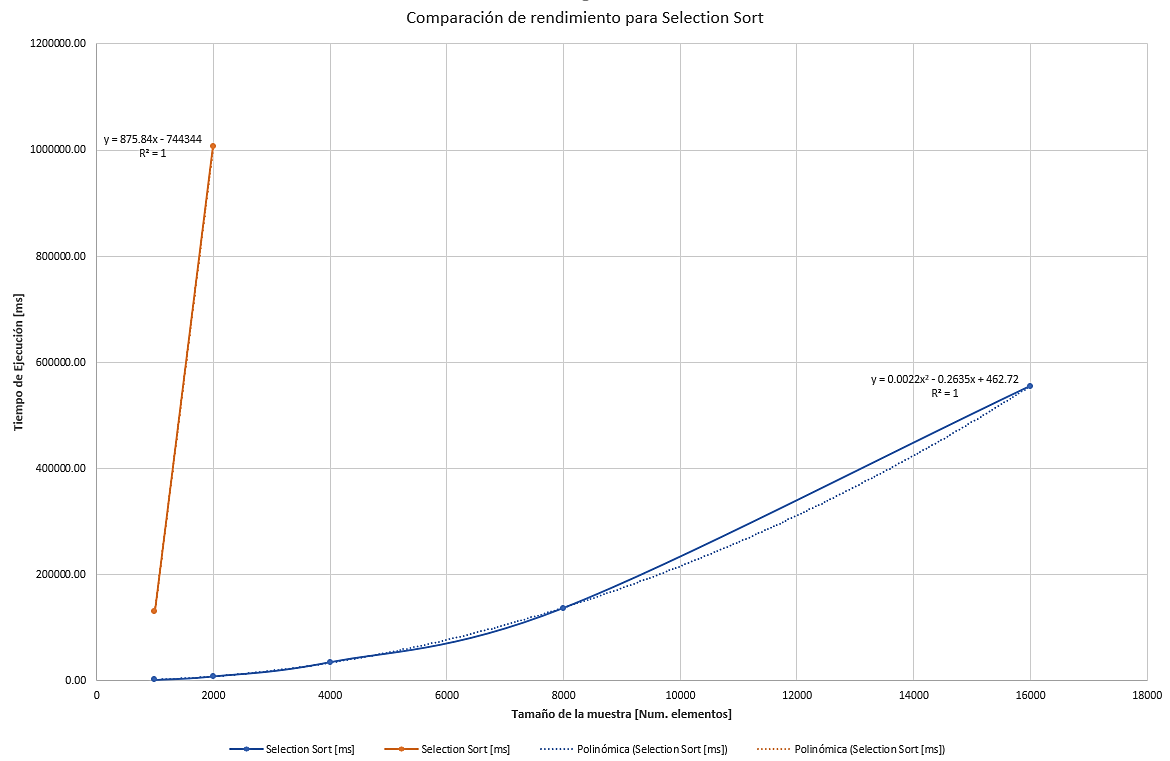
* + Comparación de rendimiento LINKED\_LIST.



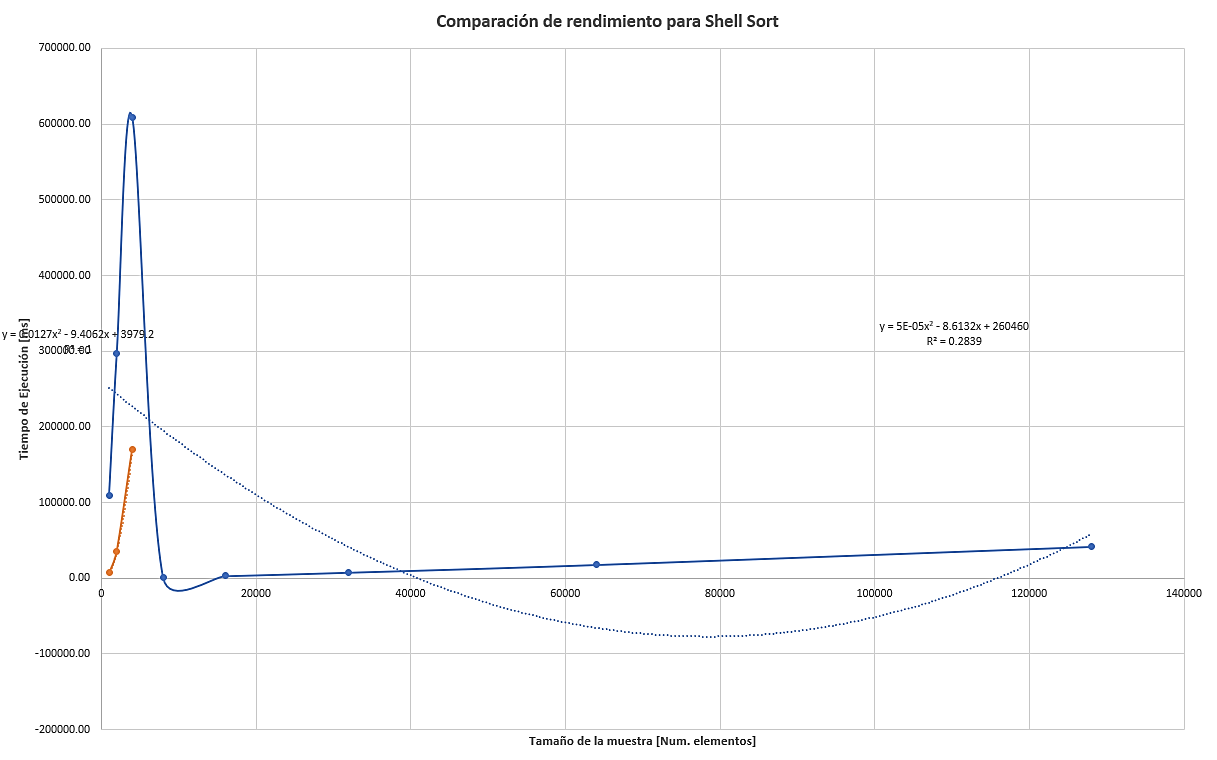
* + Comparación de rendimiento para Insertion Sort.



* + 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 | 781.25 | 739.58 | 46.88 |
| 2000 | 3156.25 | 2968.75 | 93.75 |
| 4000 | 13078.12 | 12343.75 | 250.00 |
| 8000 | 53593.75 | 55734.38 | 531.25 |
| 16000 | 222937.5 | 226546.88 | 1250.00 |
| 32000 | Tiempo excedido | Tiempo excedido | 3046.88 |
| 64000 | Tiempo excedido | Tiempo excedido | Error\* |
| 128000 | Tiempo excedido | Tiempo excedido | Error\* |
| 256000 | Tiempo excedido | Tiempo excedido | Error\* |
| 512000 | Tiempo excedido | Tiempo excedido | Error\* |

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

\*Para tamaños de muestra desde 64000 en el shell sort aparece un error que, después de haber investigado en internet, parece ser del sistema y no del código implementado (de hecho fue un problema que se presentó solo en una de las dos máquinas). Se trató de hallar una solución pero no fue posible.



|  |  |  |  |
| --- | --- | --- | --- |
| Tamaño de la muestra (LINKED\_LIST) | Insertion Sort (ms) | Selection Sort (ms) | Shell Sort (ms) |
| 1000 | 71671.875 | 65953.12 | 3359.38 |
| 2000 | 560562.50 | 498562.50 | 16984.38 |
| 4000 | Tiempo excedido | Tiempo excedido | 77875.00 |
| 8000 | Tiempo excedido | Tiempo excedido | 343109.38 |
| 16000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 32000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 64000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 128000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 256000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |
| 512000 | Tiempo excedido | Tiempo excedido | Tiempo excedido |

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

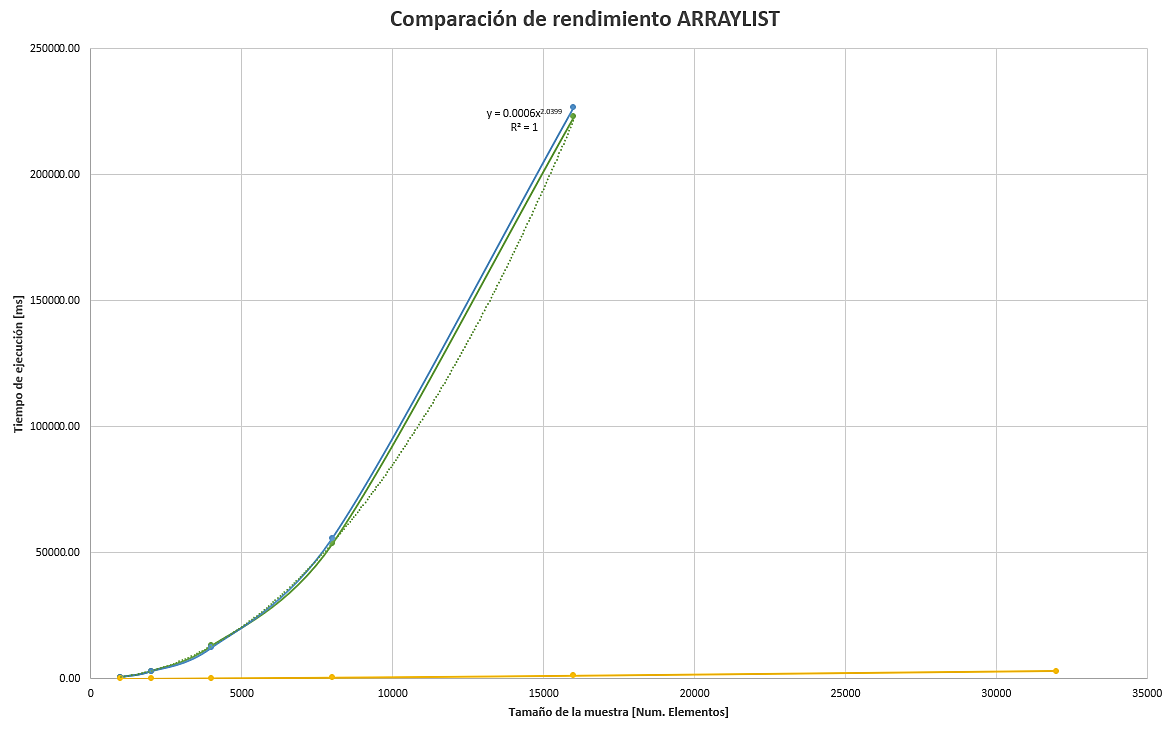
|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Insertion sort |  |  |
| Selection sort |  |  |
| Shell sort | x | x |

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

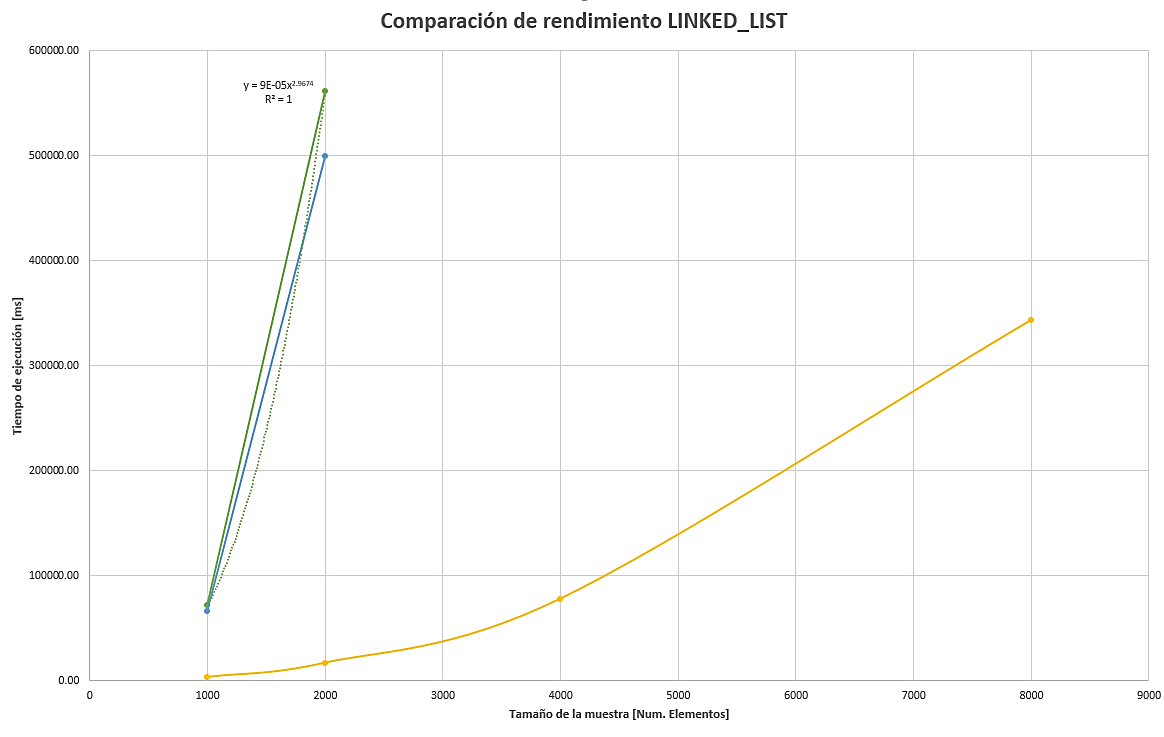
Se ha marcado en la anterior tabla, para cada tipo de estructura de datos, qué tipo de ordenamiento iterativo fue más eficiente en las pruebas. Según los datos, para ambas estructuras fue el Shell Sort el ordenamiento más eficiente, mostrando tiempos significativamente menores al de los demás ordenamientos y, por tanto, alcanzando el tiempo excedido para tamaños de muestra mucho mayores. Además, se ha determinado que el arreglo (array\_list) es el tipo de estructura de datos que exhibió tiempos de ejecución menores, resultando ser también la mejor opción entre las dos posibles.

## **Graficas**

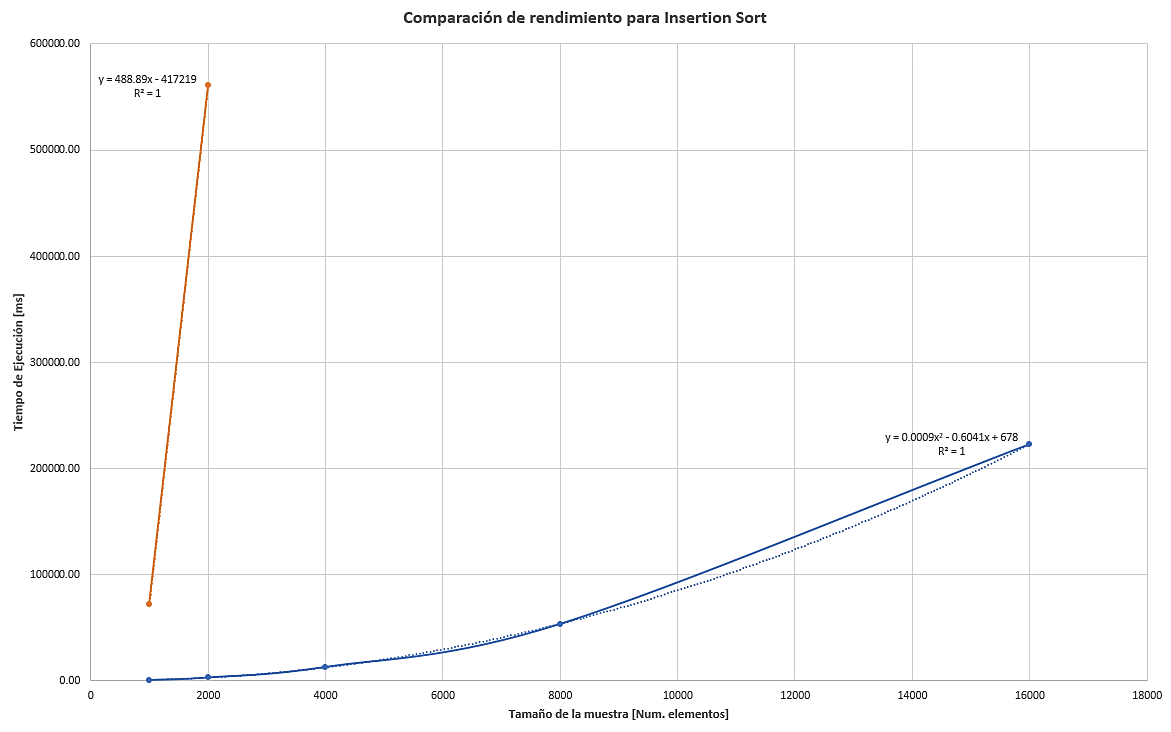
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.



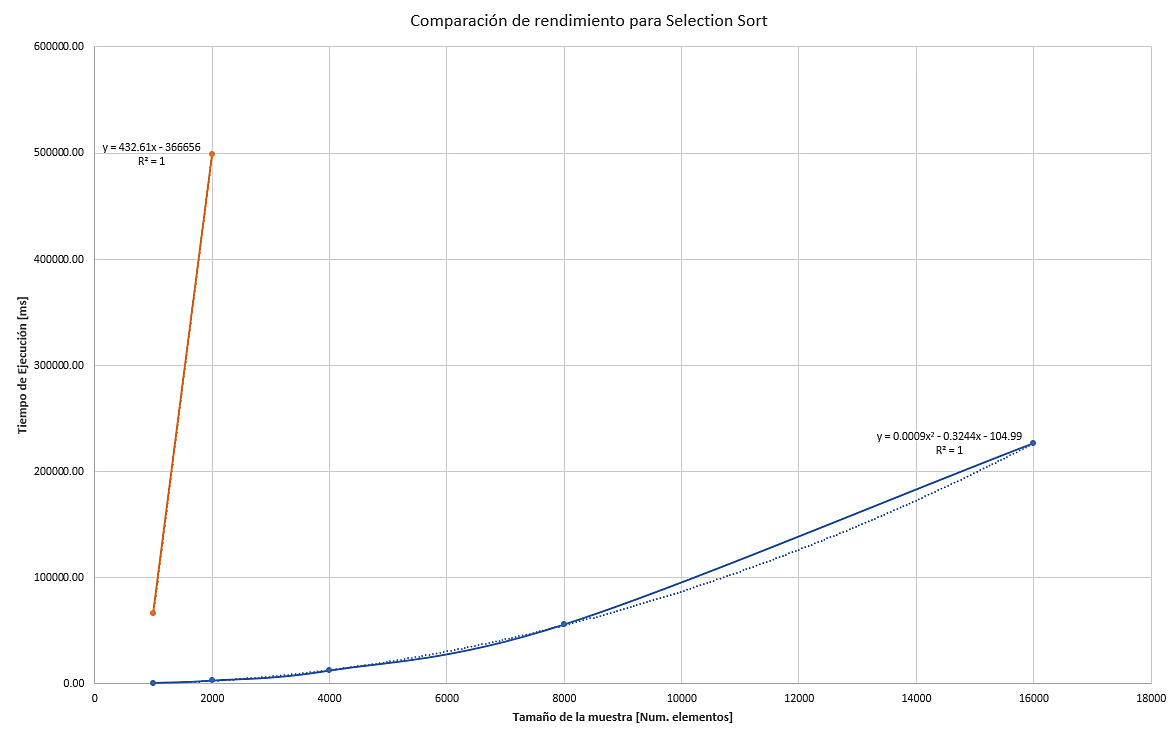
* + Comparación de rendimiento LINKED\_LIST.



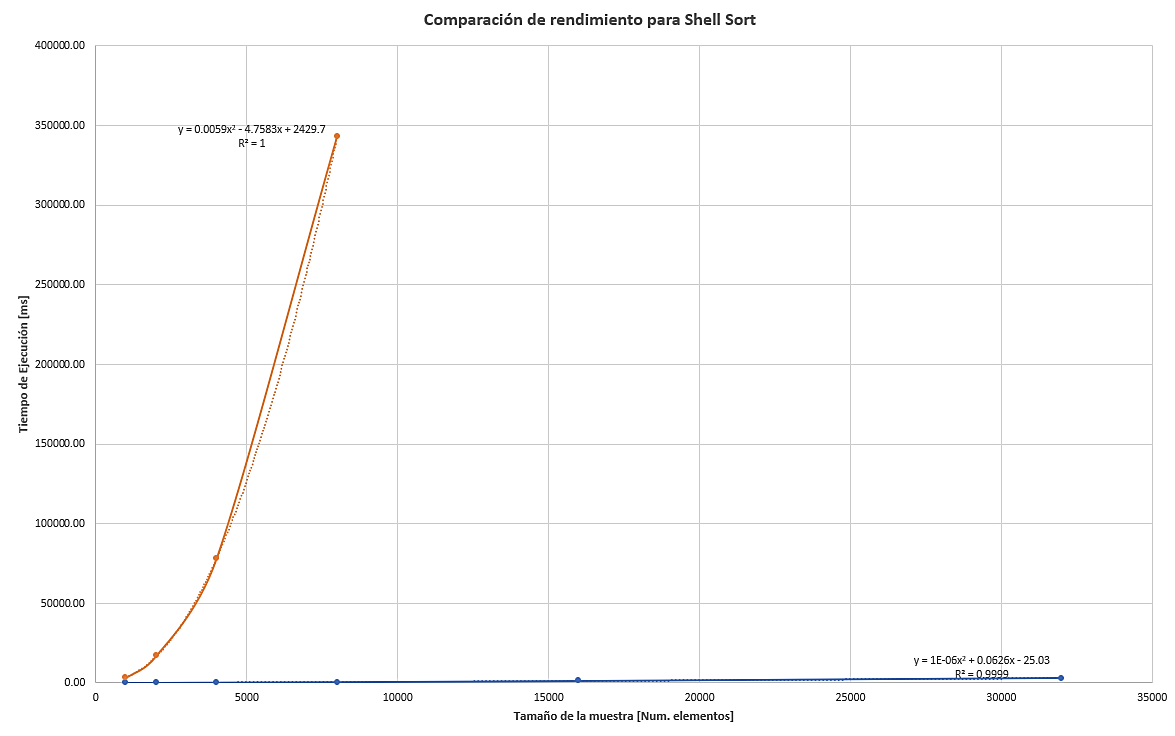
* + Comparación de rendimiento para Insertion Sort.



* + 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?

Teóricamente se ha observado que, de acuerdo a los órdenes de complejidad de cada algoritmo, debería ser significativamente más eficiente el Shell Sort, seguido del Insertion Sort, y finalmente el Selection Sort.

|  |  |  |  |
| --- | --- | --- | --- |
| **Algoritmo** | **Mejor caso** | **Promedio** | **Peor caso** |
| Shell Sort |  |  |  |
| Insertion Sort |  |  |  |
| Selection Sort |  |  |  |

Y estos fueron precisamente los comportamientos observados en las pruebas.

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

Sí. Como ya se ha mostrado en la tabla 1, las dos máquinas son bastante distintas entre sí. Esto se vio claramente evidenciado en los datos; si bien los excesos de tiempo (12 minutos) fueron alcanzados para tamaños de muestra parecidos, cada tiempo de ejecución le tomó a la máquina 1 en general el doble que a la máquina 2.

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

Estas diferencias podrían deberse principalmente al rendimiento de los procesadores de cada máquina (se ha encontrado que tiene un mejor rendimiento el procesador de la máquina 2, con un rendimiento global estimado en 217.3% mejor que el del procesador correspondiente a la máquina 1).

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

Se ha encontrado que la mejor estructura de datos ha sido el arreglo (array\_list), pues exhibió menores tiempos de ejecución para todos los ordenamientos que la lista encadenada (linked\_list), y en las gráficas en general exhibe órdenes de complejidad correspondientes a algoritmos de mayor eficiencia.

# **Referencias**

[1] Technical City. (s.f.). AMD A10-8700P vs. Ryzen 5 3500U. *Technical City*. Obtenido de https://technical.city/es/cpu/A10-8700P-vs-Ryzen-5-3500U

[2] Wikipedia. (2021). *Wikipedia*. Obtenido de https://en.wikipedia.org/wiki/Sorting\_algorithm#Classification