OBSERVACIONES DE LA PRACTICA

Gregorio Salazar 202022085

Valentina Uribe 201817485

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz | Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz |
| Memoria RAM (GB) | 8 | 8 |
| Sistema Operativo | Windows 10 | Windows 10 |

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 | 812.51 | 968.75 | 46.88 | 20.63 | 20.63 |
| 2000 | 3567.72 | 4218.51 | 109.38 | 62.5 | 46.88 |
| 4000 | 13484.38 | 15375.01 | 218.75 | 156.25 | 93.75 |
| 8000 | 56890.63 | 65062.51 | 578.13 | 234.38 | 220.75 |
| 16000 | 212187.51 | 262515.63 | 1312.51 | 578.13 | 484.38 |
| 32000 | 895343.75 | 1076437.51 | 2828.13 | 1078.13 | 1062.51 |
| 64000 | TE | TE | 7296.88 | 2578.13 | 2375.13 |
| 128000 | TE | TE | 16640.63 | 5102.75 | 4921.88 |
| 256000 | TE | TE | 42234.38 | 11062.32 | 10140.41 |
| 375942 | TE | TE | 69359.38 | 19140.25 | 16421.34 |

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 | 50781.31 | 37531.25 | 2421.88 | 1859.38 | 200.21 |
| 2000 | 356781.21 | 318828.25 | 9421.88 | 7609.38 | 734.38 |
| 4000 | TE | TE | 46906.23 | 30046.01 | 2851.75 |
| 8000 | TE | TE | 233859.38 | 137203.21 | 16281.32 |
| 16000 | TE | TE | 1085000.21 | 596850.88 | 53546.78 |
| 32000 | TE | TE | TE | TE | 228593.88 |
| 64000 | TE | TE | TE | TE | TE |
| 128000 | TE | TE | TE | TE | TE |
| 256000 | TE | TE | TE | TE | TE |
| 375942 | TE | TE | TE | TE | TE |

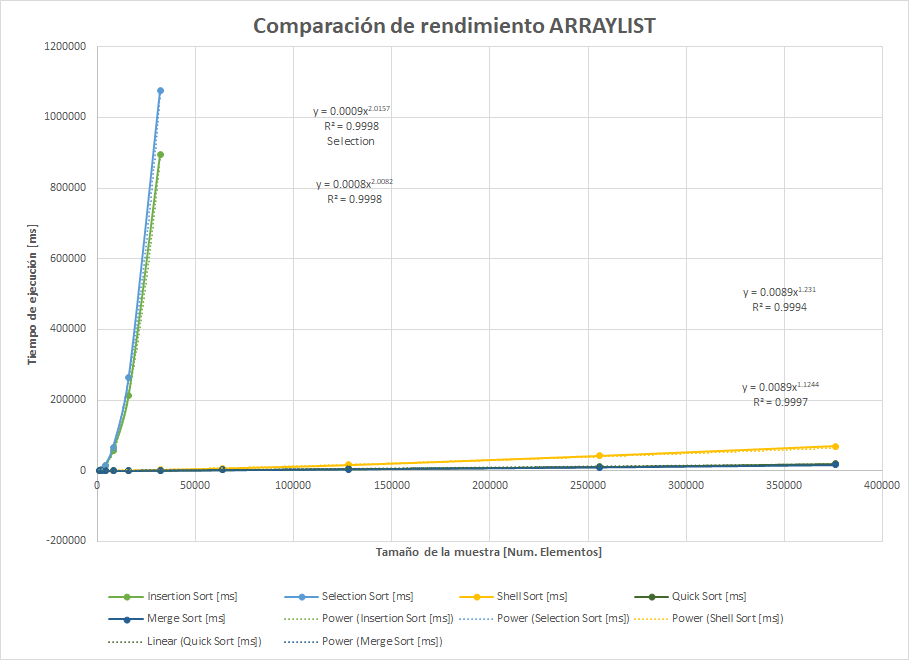
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

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

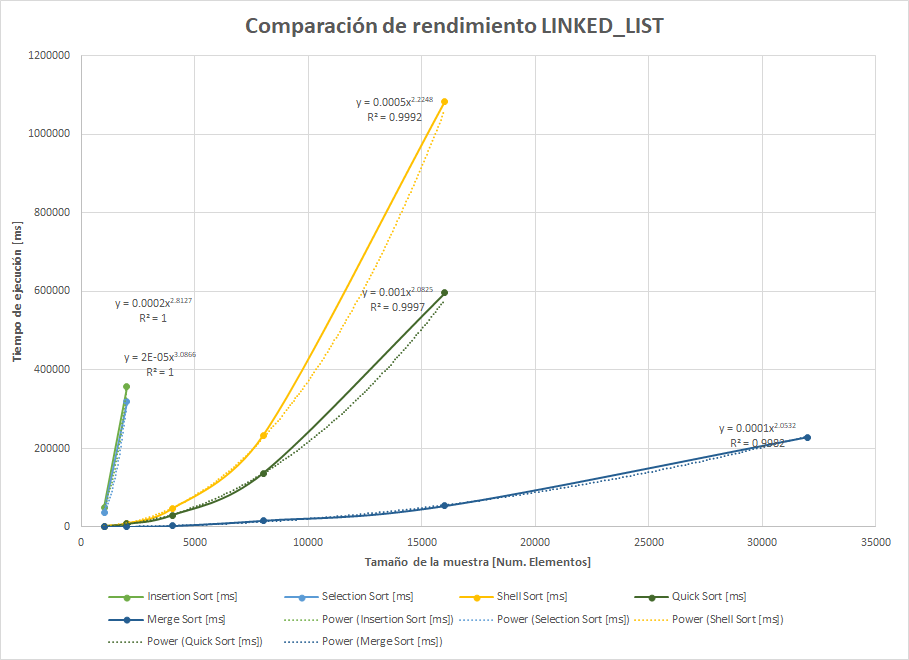
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

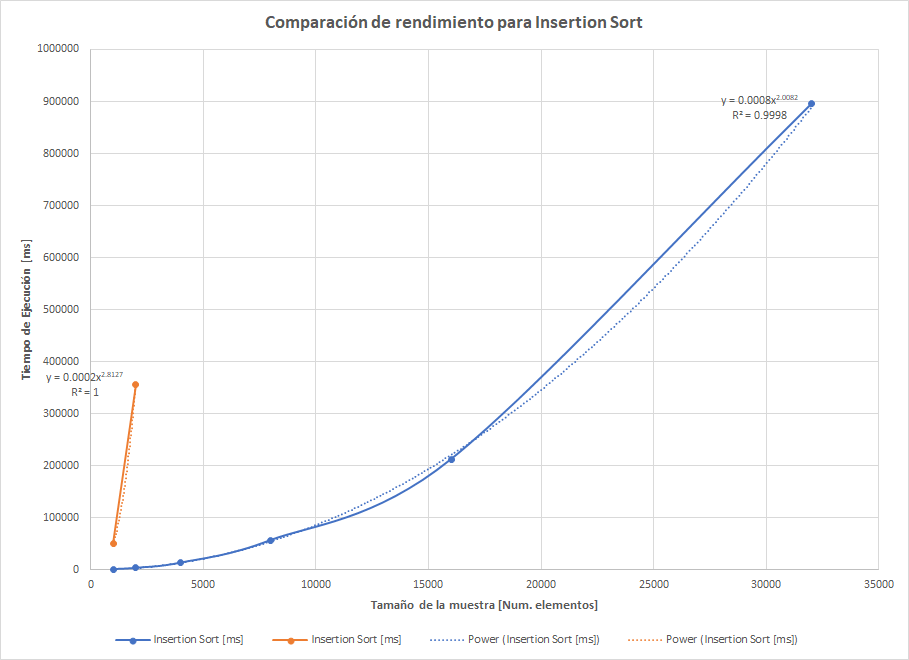
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.



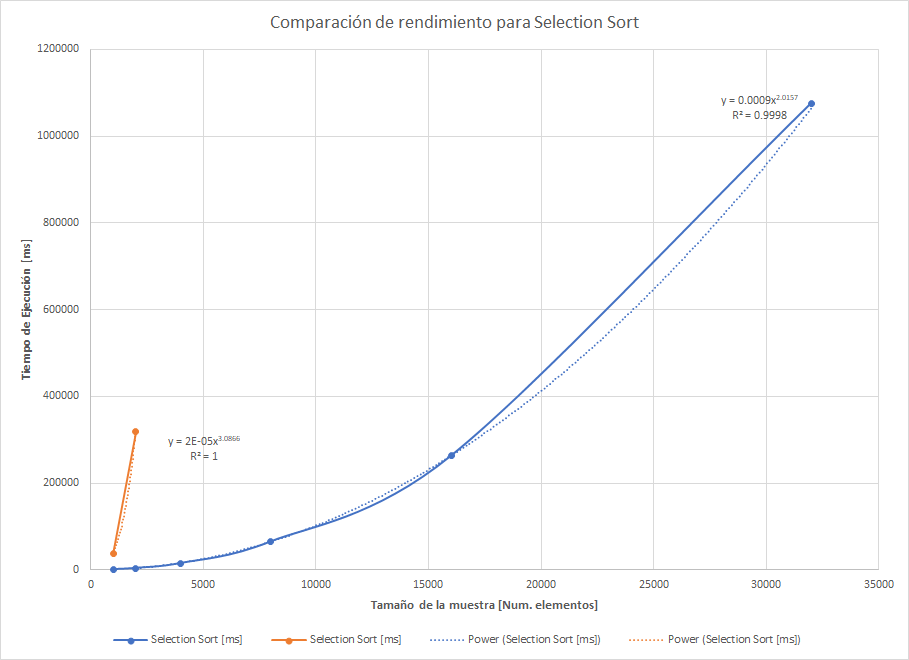
* + Comparación de rendimiento LINKED\_LIST.



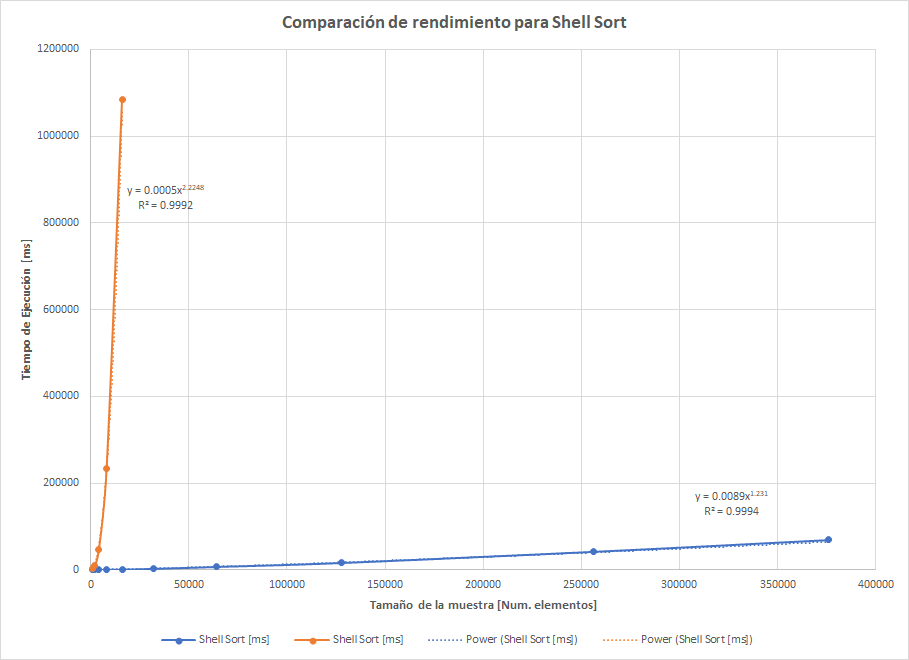
* + Comparación de rendimiento para Insertion Sort.



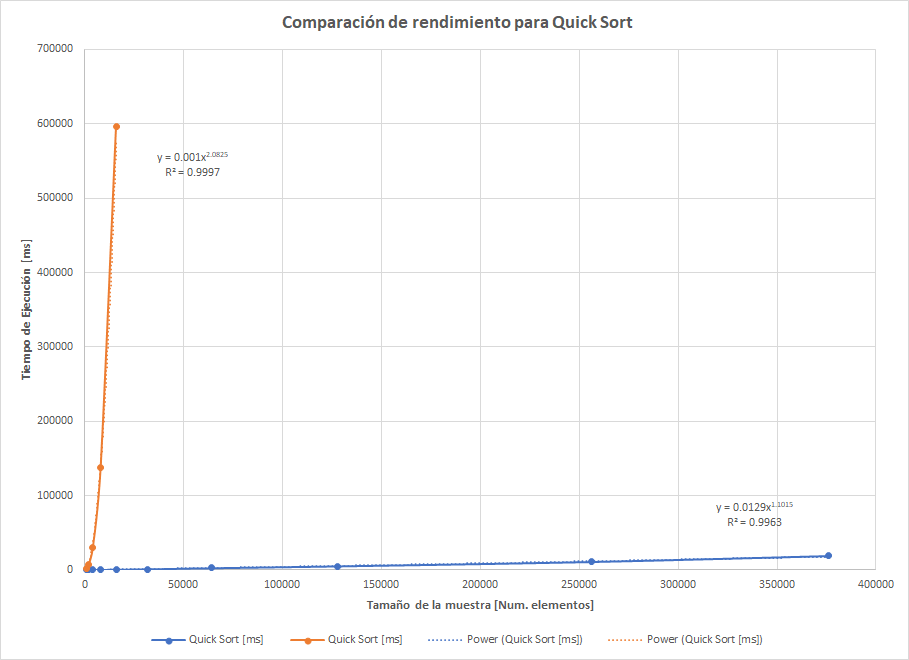
* + Comparación de rendimiento para Selection Sort.



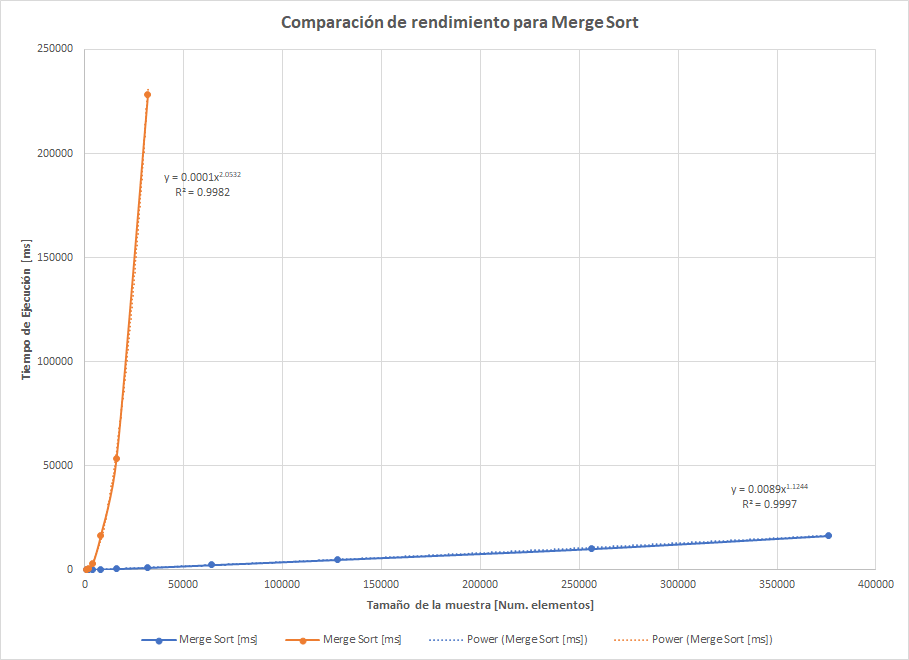
* + Comparación de rendimiento para Shell Sort.



* + 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 | 718.75 | 843.75 | 46.875 | 15.625 | 15.625 |
| 2000 | 2906.25 | 3656.25 | 93.75 | 78.125 | 46.875 |
| 4000 | 11500 | 13281.25 | 218.75 | 125.0 | 140.625 |
| 8000 | 47437.5 | 53937.5 | 515.625 | 281.25 | 281.25 |
| 16000 | 188453.125 | 22658.375 | 1140.625 | 625.0 | 609.375 |
| 32000 | 798750.0 | 94131.5 | 2593.75 | 1296.87 | 1265.625 |
| 64000 | TE | TE | 6312.5 | 2796.87 | 2687.5 |
| 128000 | TE | TE | 15375 | 6140.62 | 5750.0 |
| 256000 | TE | TE | 37843.75 | 13125 | 12187.5 |
| 375942 | TE | TE | 61484.375 | 21078.1 | 18578.25 |

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 | 42953.125 | 38375.0 | 2281.25 | 1718.75 | 203.125 |
| 2000 | 352656.25 | 312531.25 | 9296.875 | 7890.62 | 781.25 |
| 4000 | TE | TE | 45781.875 | 31390.62 | 3156.25 |
| 8000 | TE | TE | 232546.875 | 3390.625 | 1356.25 |
| 16000 | TE | TE | 1079078.125 | 137687.5 | 50437.5 |
| 32000 | TE | TE | TE | 597218.75 | 203484.375 |
| 64000 | TE | TE | TE | TE | TE |
| 128000 | TE | TE | TE | TE | TE |
| 256000 | TE | TE | TE | TE | TE |
| 375942 | TE | TE | TE | TE | TE |

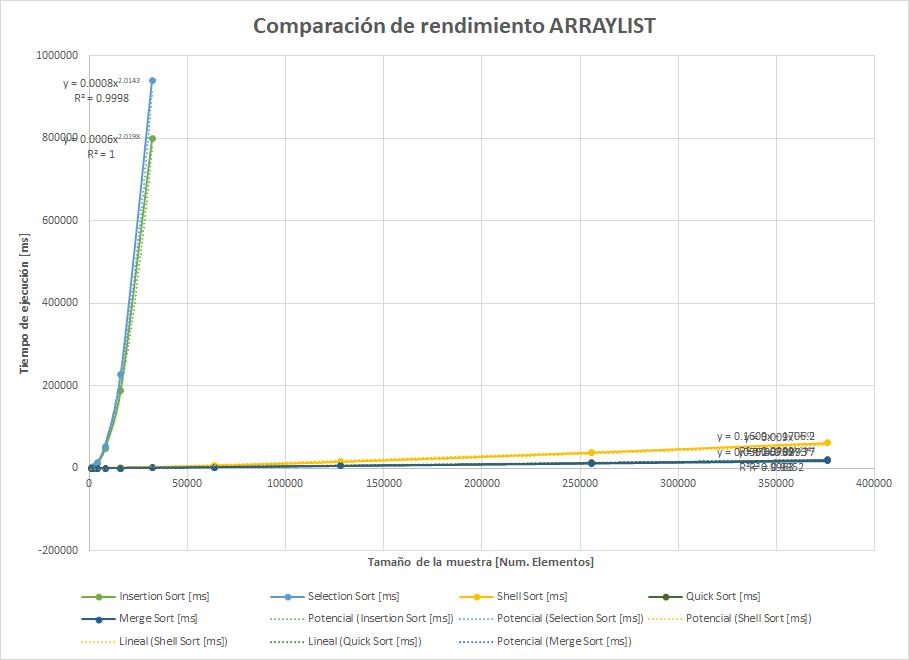
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

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

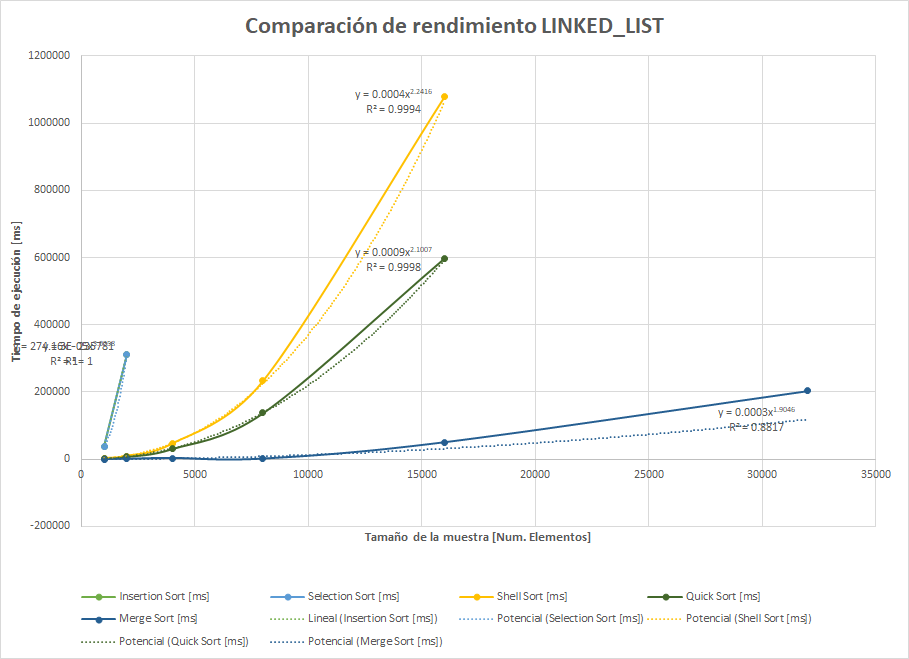
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

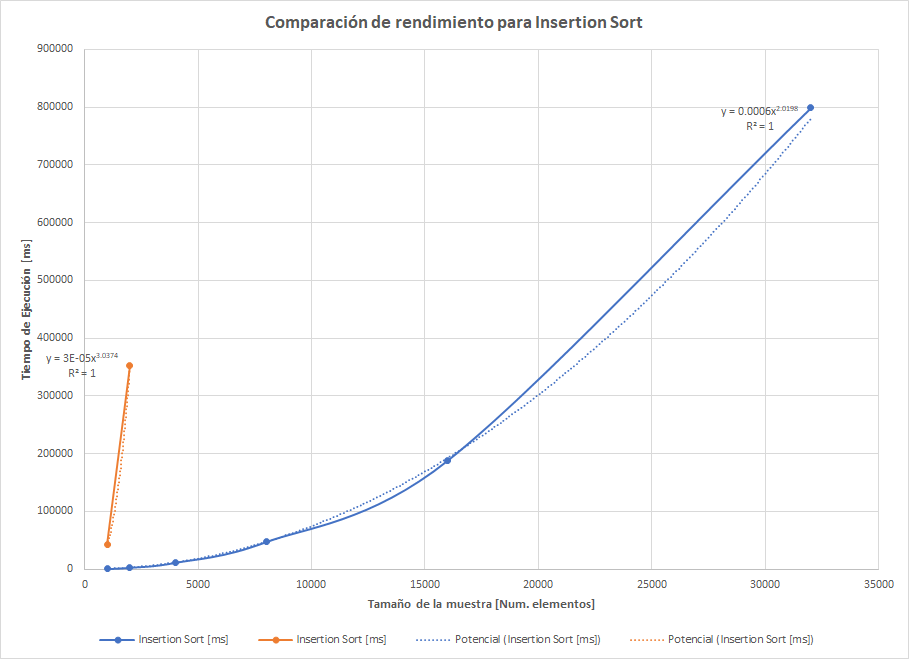
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 2.**
  + Comparación de rendimiento ARRAYLIST.



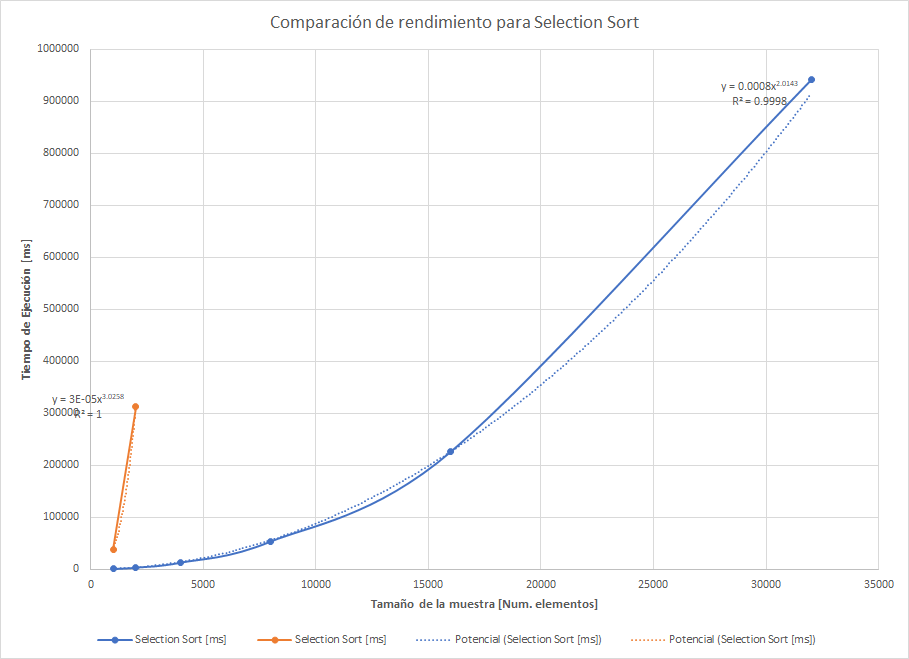
* + Comparación de rendimiento LINKED\_LIST.



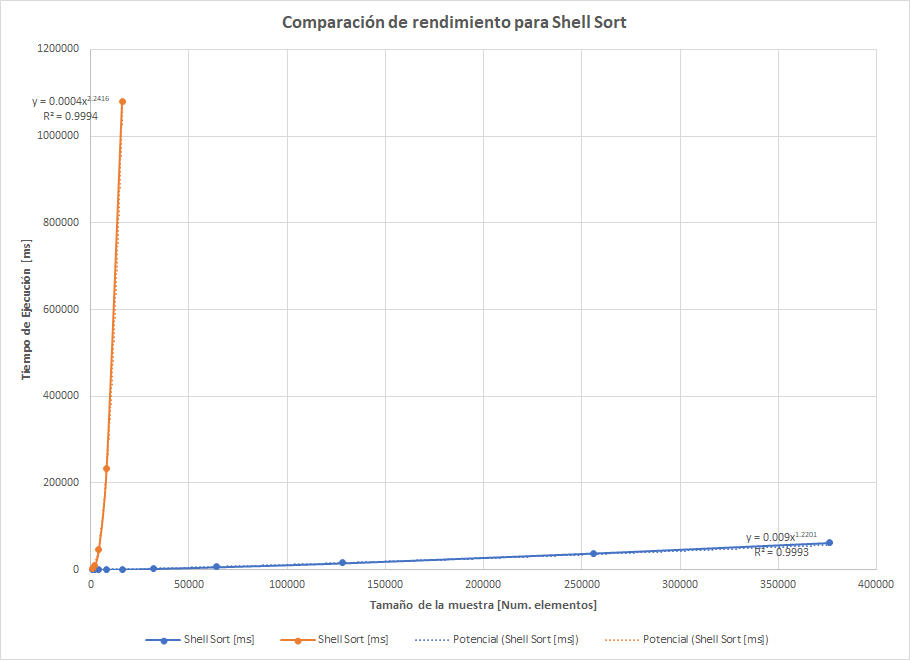
* + Comparación de rendimiento para Insertion Sort.



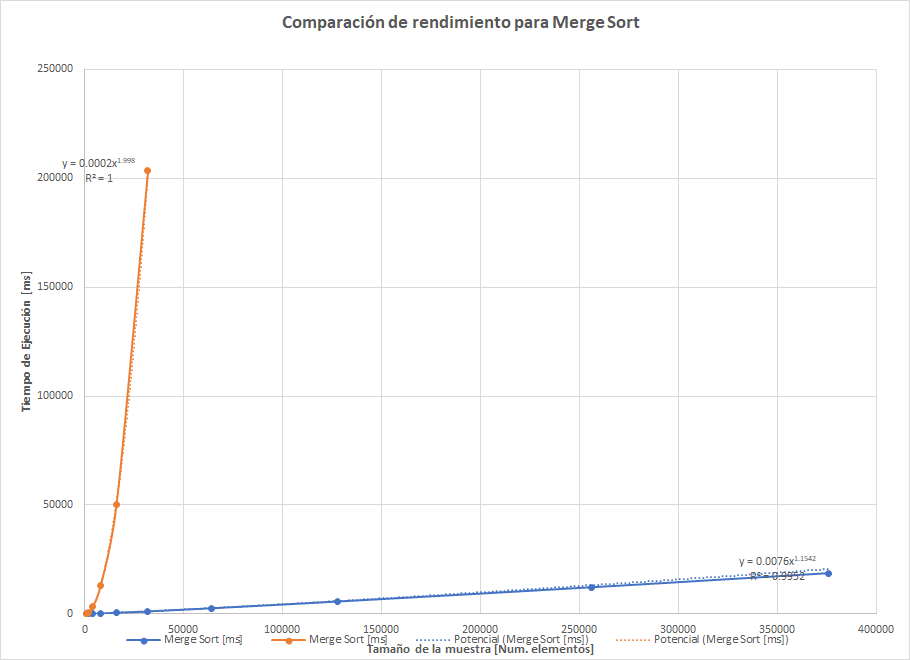
* + Comparación de rendimiento para Selection Sort.\



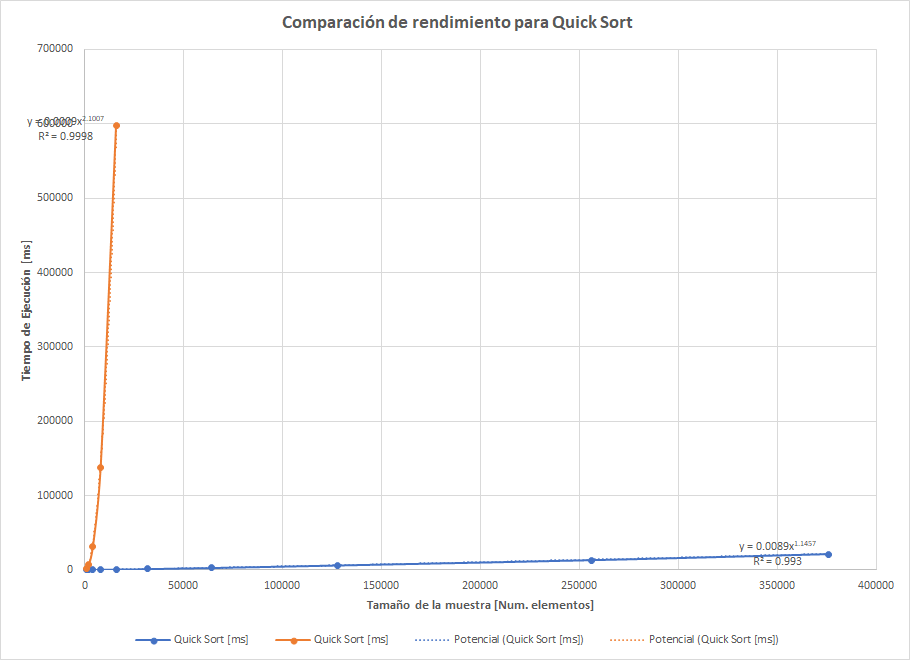
* + Comparación de rendimiento para Shell Sort.



* + 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 es acorde con lo esperado. Todas las gráficas fueron aproximadas con una función potencia del estilo ax^b, cosa que permite analizar su complejidad de crecimiento big O mucho mejor que otras aproximaciones. Lo visto en los casos Merge y Quick fueron lo esperado. En array ambos valores fueron aproximadamente iguales e iguales a O(n^1.1). Lo esperado sería O(nlogn) pero no existe esta aproximación en excel, y O(nlogn) es mayor que O(n^1) pero menor que O(n^b) para todo b mayor que 1, por lo que una aproximación cercana a 1 es muy acorde con lo esperado, y entre más datos se tomen se puede esperar que el exponente baje un poco pero no sea nunca 1. En linked list se ven valores cercanos a 2, es decir del estilo O(n^2.01). Esto es un poco mayor a 2 ya que el linked, al tener operaciones de complejidad O(n) que en el arreglo eran de complejidad O(1), se le suma un uno al exponente (1 es n^0), por lo que por la misma lógica anterior del logn, lo esperable es un valor un poco superior a 2.

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

En estas pruebas no hubo ninguna diferencia significativa. La maquina 1 fue ligeramente más rápida, pero el comportamiento cualitativo de los algoritmos fueron exactamente iguales.

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

La máquina 1 tiene un mejor procesador, no mucho mejor. Esto puede explicar un mejor rendimiento en el merge y quick. En el laboratorio anterior la máquina 2 registro levemente mejores tiempos. Esto se puede atribuir a que durante las pruebas la 2 estaba conectada a la electricidad y la 1 no, mientras que este laboratorio ambas máquinas estaban en igualdad de condiciones

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

Es evidentemente el ARRAY. Los resultados son contundentes. También se observa que En cada algoritmo, la complejidad de array contra linked difiere por 1 en el exponente. Como se dijo anteriormente esto se debe a que varias operaciones en linked tienen complejidad O(n) que en array tienen O(1). Array es bastamente superior en tiempos de carga.

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.

El ranking con los resultados obtenidos es evidente, en orden de mejor a peor son: Merge, Quick, Shell, Insertion y por útlimo Selection. Esto se concluye de los tiempos obtenidos en las tablas, y del analisis de complejidad de las gráficas.