OBSERVACIONES DE LA PRACTICA

Estudiante 1: Nathalia Quiroga 202013212

Estudiante 2: David Valderrama 201910987

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Core i5 8th Gen 1,8 GHz de dos núcleos | AMD Ryzen 7 4800H with Radeon Graphics, 2900 MHz, 8 Core(s) 16 Logical Processor(s) |
| Memoria RAM (GB) | 8 GB 1600 MHz DDR3 | 8GB 3200 MHz |
| Sistema Operativo | MacOS (64-bit) | Windows (64-bit) |

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 | 936,72 | 1081,90 | 64,22 | 48,01 | 47,41 |
| 2000 | 4179,46 | 4335,44 | 121,42 | 91,49 | 89,97 |
| 4000 | 15633,21 | 17379,73 | 275,45 | 186,67 | 183,70 |
| 8000 | 61177,53 | 72191,01 | 638,90 | 378,22 | 388,00 |
| 16000 | 262760,03 | 295861,36 | 1414,58 | 795,17 | 813,05 |
| 32000 | 1088380,40 | N/A | 3347,23 | 1798,57 | 1843,28 |
| 64000 | N/A | N/A | 8170,44 | 4170,56 | 4000,69 |
| 128000 | N/A | N/A | 18584,19 | 12593,45 | 7824,30 |
| 256000 | N/A | N/A | 45134,62 | 56713,28 | 16552,29 |
| 375942 | N/A | N/A | 78153,80 | 114676,93 | 25157,33 |

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 | 57043,90 | 53259,20 | 3320,09 | 2223,67 | 287,27 |
| 2000 | 483345,96 | 437846,96 | 13489,55 | 10714,94 | 1134,01 |
| 4000 | N/A | N/A | 66628,06 | 47355,34 | 4917,68 |
| 8000 | N/A | N/A | 325194,61 | 191597,12 | 18385,38 |
| 16000 | N/A | N/A | N/A | N/A | 77286,78 |
| 32000 | N/A | N/A | N/A | N/A | 306050,81 |
| 64000 | N/A | N/A | N/A | N/A | N/A |
| 128000 | N/A | N/A | N/A | N/A | N/A |
| 256000 | N/A | N/A | N/A | N/A | N/A |
| 375942 | N/A | N/A | N/A | N/A | N/A |

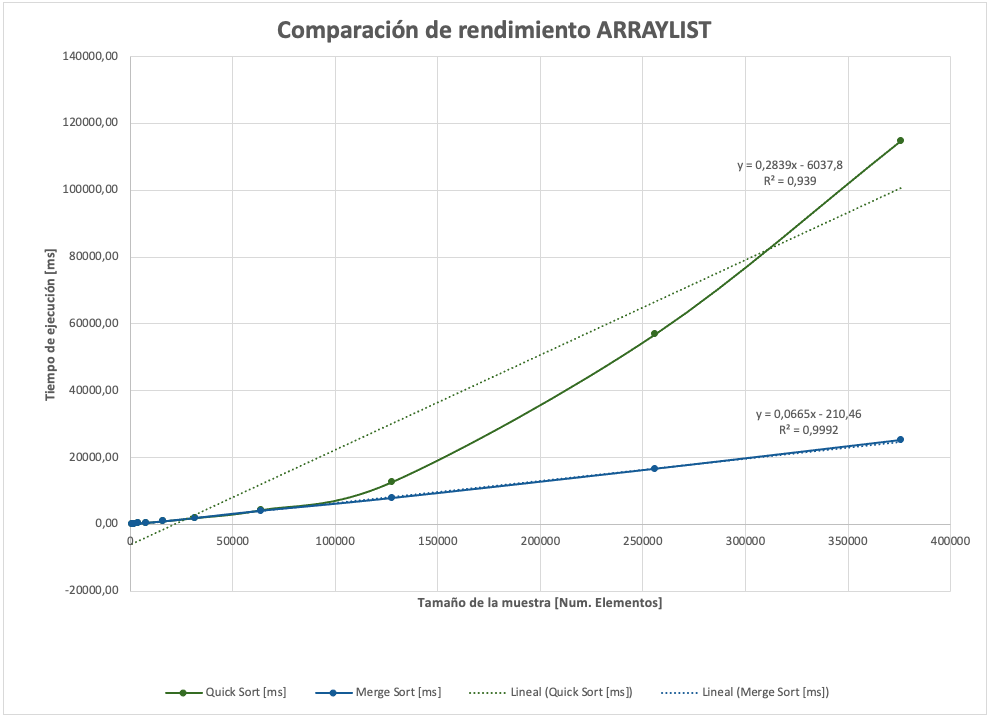
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | 1 | 1 |
| Quick sort | 2 | 2 |

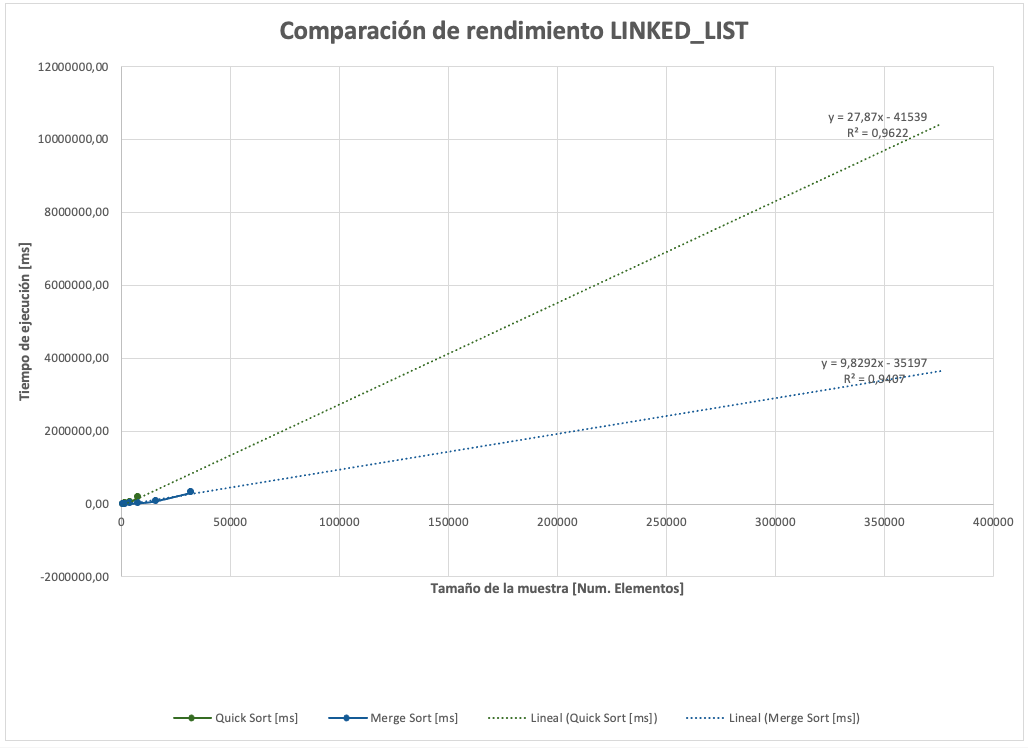
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **Graficas**

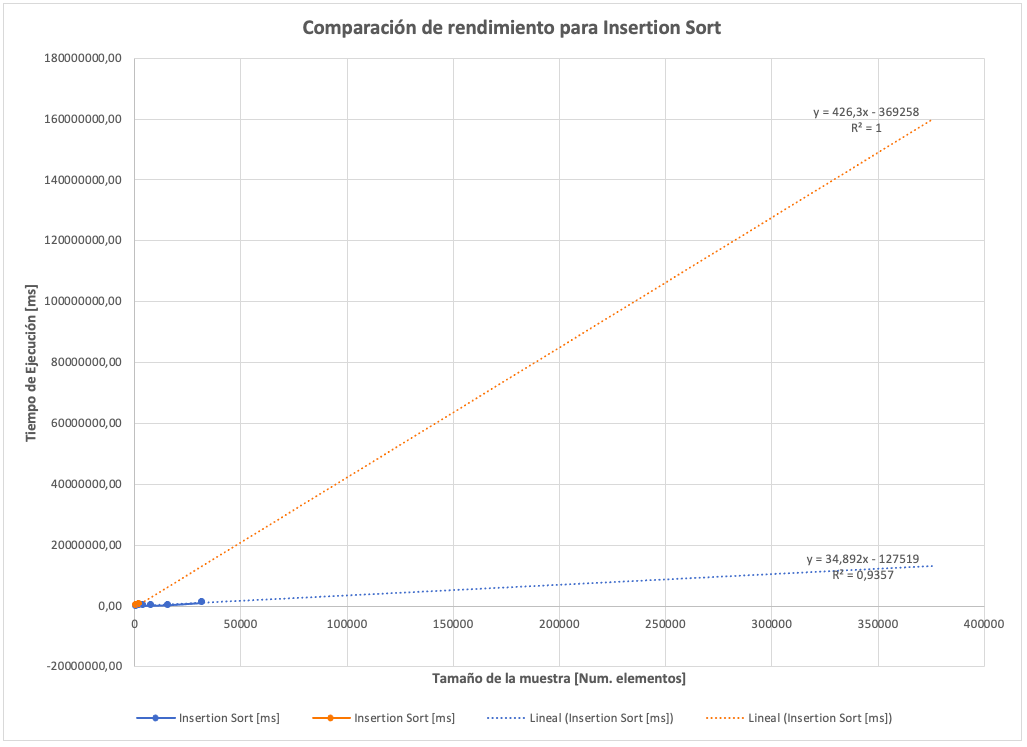
* Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  + Comparación de rendimiento ARRAYLIST.



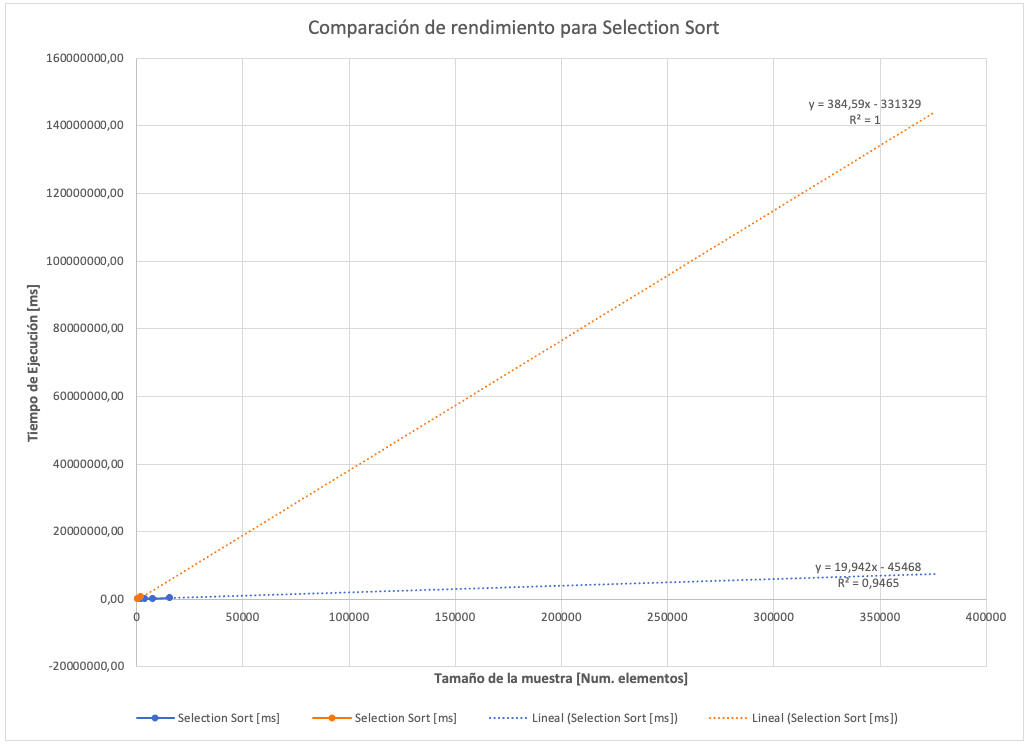
* + Comparación de rendimiento LINKED\_LIST.



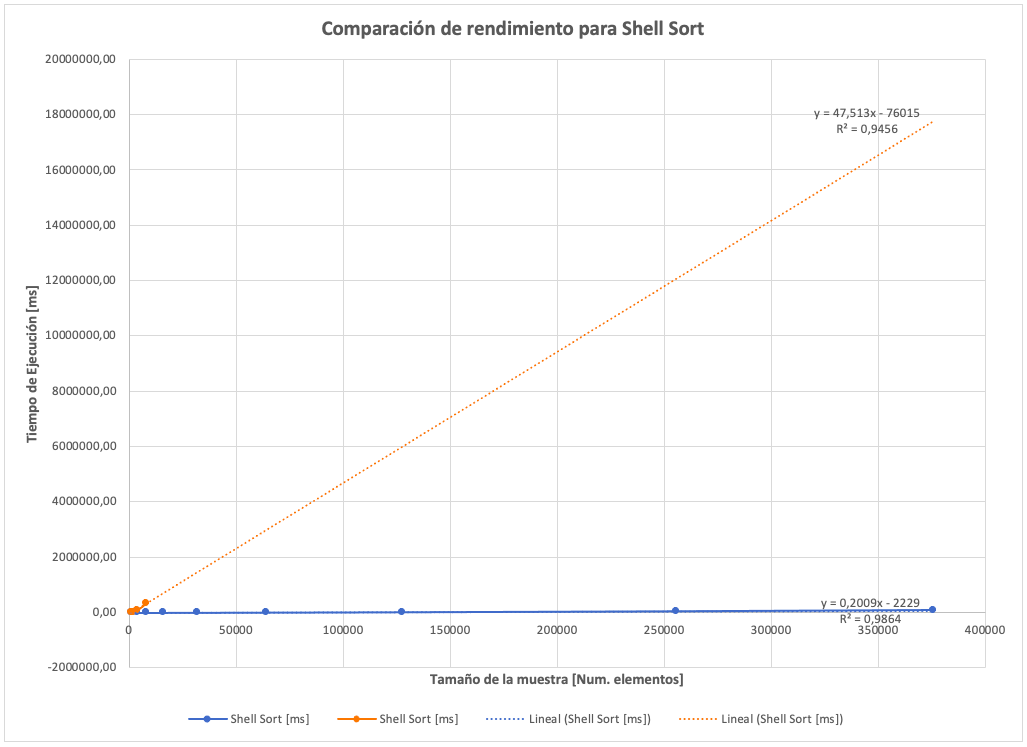
* + Comparación de rendimiento para Insertion Sort.



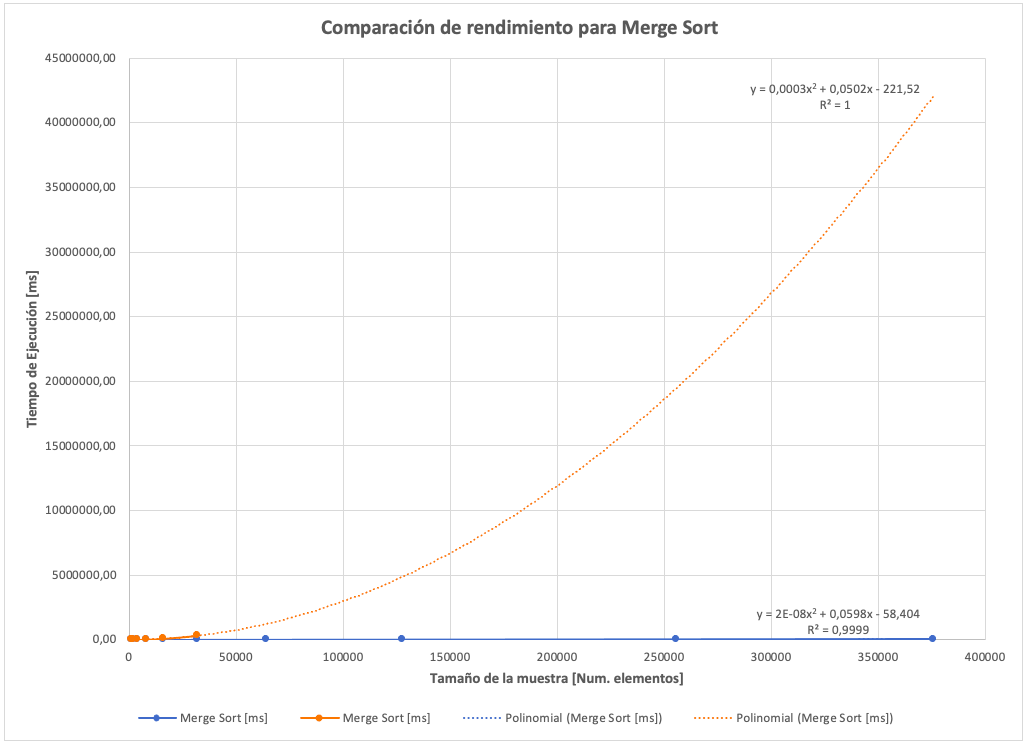
* + Comparación de rendimiento para Selection Sort.



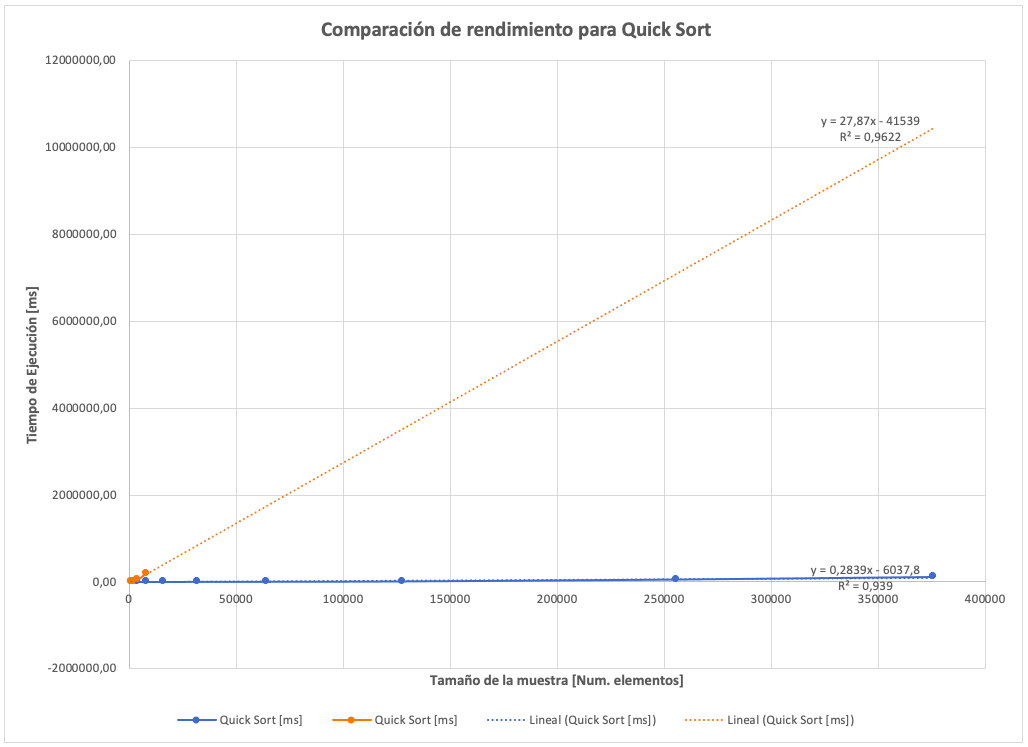
* + Comparación de rendimiento para Shell Sort.



* + 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 | 583.33 | 625.00 | | 41.67 | | | 26.04 | | | 26.04 | | |
| 2000 | 2359.38 | 2541.67 | | 67.71 | | | 46.88 | | | 46.88 | | |
| 4000 | 9604.17 | 10026.04 | | 166.67 | | | 104.17 | | | 109.38 | | |
| 8000 | 37317.71 | 42687.50 | | 375.00 | | | 223.96 | | | 234.38 | | |
| 16000 | 168953.12 | 194265.62 | | 869.79 | | | 489.59 | | | 515.62 | | |
| 32000 | 681812.50 | 764953.12 | | 2046.88 | | | 1104.17 | | | 1125.00 | | |
| 64000 | N/A | N/A | | 5020.83 | | | 2553.54 | | | 2437.50 | | |
| 128000 | N/A | N/A | | 11343.75 | | | 7947.92 | | | 5265.62 | | |
| 256000 | N/A | N/A | | 27927.08 | | | 35793.72 | | | 10937.50 | | |
| 375942 | N/A | | N/A | | 47234.38 | | | 72385.24 | | | 16687.5 | |

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 | 51817.71 | 45791.67 | 2786.46 | 1937.50 | 250.03 |
| 2000 | 427187.50 | 370281.25 | 11614.58 | 9140.62 | 948.38 |
| 4000 | N/A | N/A | 56031.25 | 41546.88 | 3781.25 |
| 8000 | N/A | N/A | 270515.62 | 161187.50 | 16390.62 |
| 16000 | N/A | N/A | N/A | 712640.62 | 63343.75 |
| 32000 | N/A | N/A | N/A | N/A | 256328.12 |
| 64000 | N/A | N/A | N/A | N/A | N/A |
| 128000 | N/A | N/A | N/A | N/A | N/A |
| 256000 | N/A | N/A | N/A | N/A | N/A |
| 375942 | N/A | N/A | 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) |
| Merge sort | 1 | 1 |
| Quick sort | 2 | 2 |

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.**
  + Chart, line chart

    Description automatically generatedComparación de rendimiento ARRAYLIST.
  + Chart

    Description automatically generatedComparación de rendimiento LINKED\_LIST.
  + Chart, line chart

    Description automatically generatedComparación de rendimiento para Insertion Sort.
  + Chart, line chart

    Description automatically generatedComparación de rendimiento para Selection Sort.
  + Chart, line chart

    Description automatically generatedComparación de rendimiento para Shell Sort.
  + Chart, line chart

    Description automatically generatedComparación de rendimiento para MergeSort.
  + Chart, line chart

    Description automatically generatedComparación de rendimiento para QuickSort.

# **Preguntas de análisis**

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

Tomando en cuenta los datos recolectados de las pruebas, podemos concluir que la teoría se cumple casi en su totalidad ya que la tendencia tanto de *quick* como de *merge* es la esperada.

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

Sí, la diferencia es notable ya que en algunos casos la máquina 2 tardó en procesar los datos hasta un 45% menos que la máquina 1.

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

Creemos que esto se puede deber a la velocidad de memoria RAM, “ya que cuanto mayor sea la velocidad de la memoria, más rápido podrá trabajar los datos” (Velasco, 2021, párr. 2). Por otro lado, cuando comparamos los dos procesadores en una página web que guarda los resultados de tests de benchmark de varios usuarios de internet (cpu.userbenchmark.com), arrojó que el AMD Ryzen 7 4800H es más rápido que el Intel Core i5 5350U en 63%.

Velasco, R. (2021, 19 de Abril). *¿Qué influye más en la memoria RAM para jugar, la frecuencia o la latencia?*. Hard Zone. https://hardzone.es/reportajes/comparativas/latencia-velocidad-ram-jugar/

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

En el caso de la implementación de diferentes tipos de algoritmos de ordenamiento se puede decir que *ARRAY\_LIST* es mejor, pues tuvo tiempos menores en los resultados de laboratorio.

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 / Insertion sort (depende de la estructura sobre la cual se implemente)