OBSERVACIONES DE LA PRACTICA

Mario Ricaurte Tobón; 201922994.

Valentina Jiménez; 201924116.

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Ryzen 5 3500U | Intel Core i7 8550U |
| Memoria RAM (GB) | 12GB | 8GB |
| Sistema Operativo | Windows 10 Home | Windows 10 Pro |

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

# **Maquina 1**

## **Resultados**



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



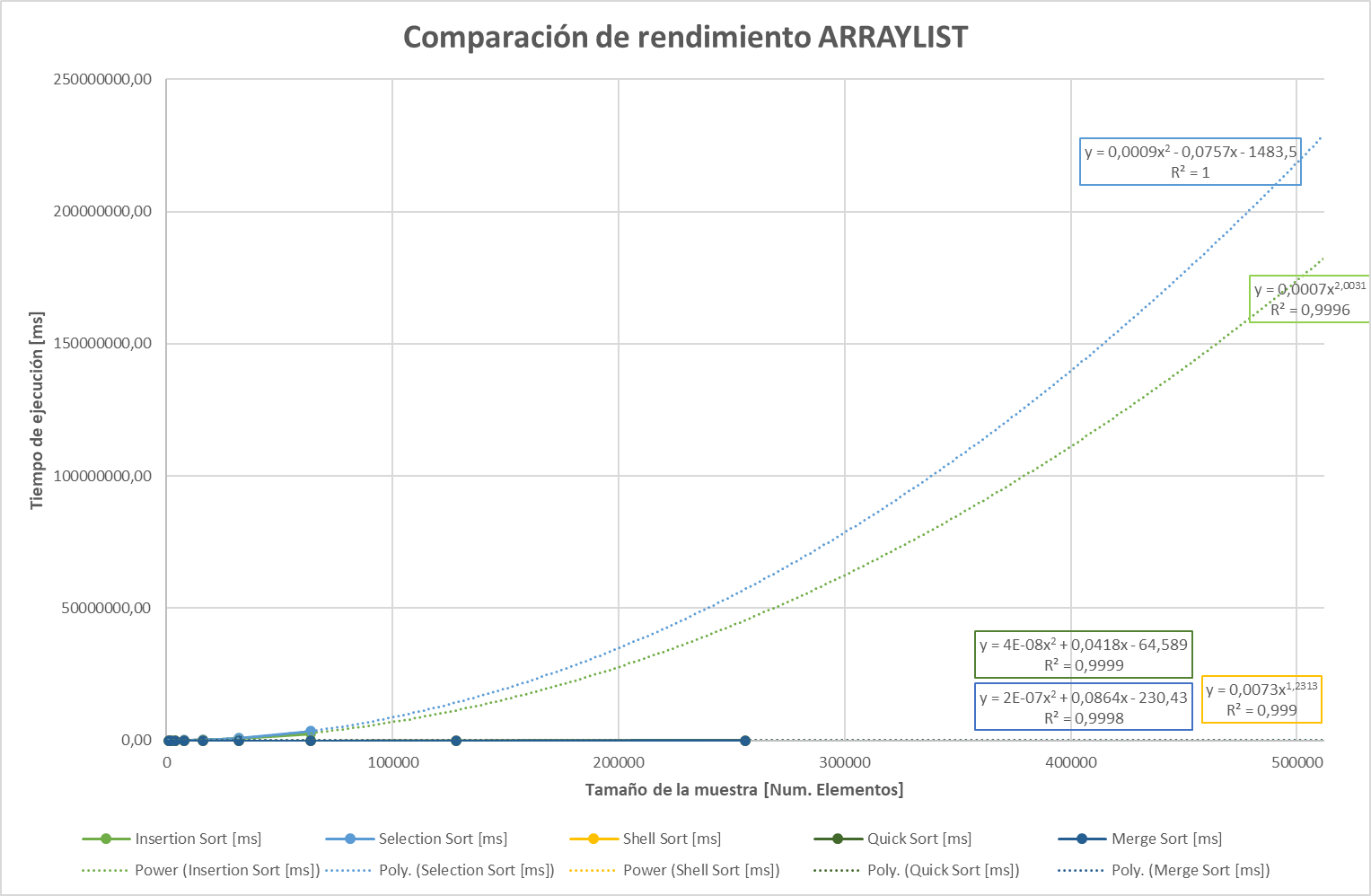
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | Menor eficiencia | Menor eficiencia |
| Quick sort | Mayor eficiencia | Mayor eficiencia |

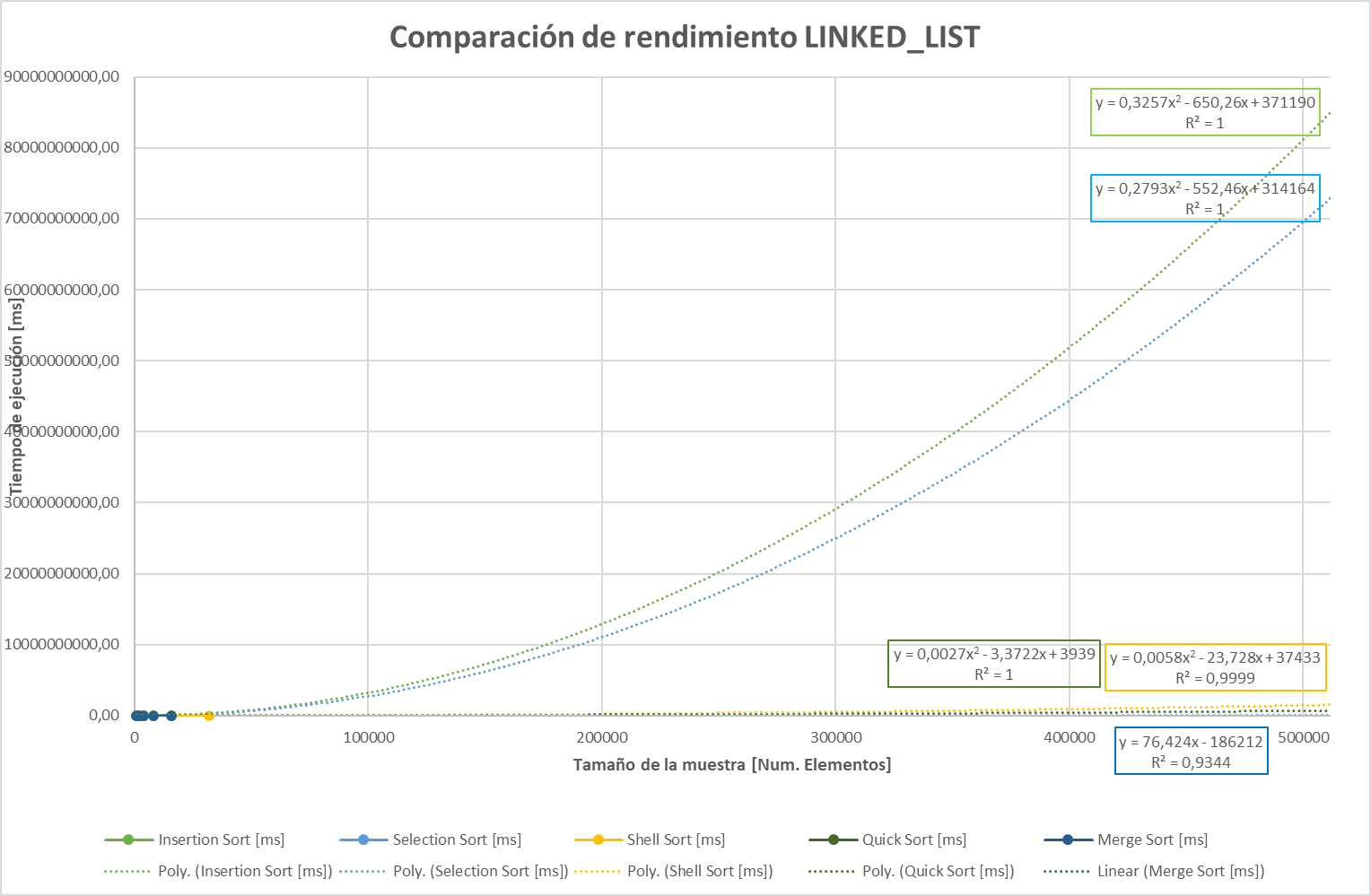
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **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**



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



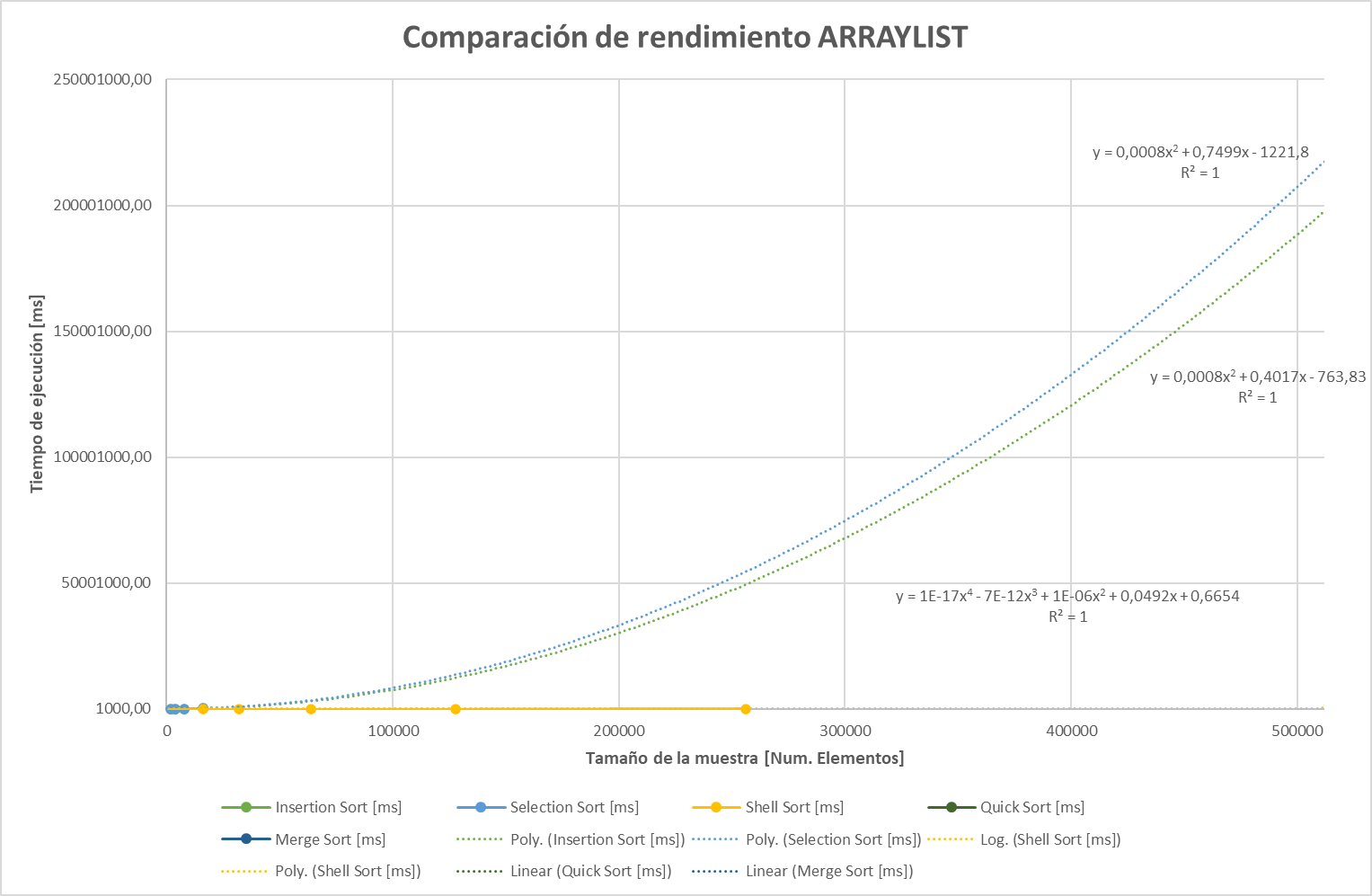
Tabla 3. Comparación de tiempos de ejecución para los ordenamientos iterativos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| Merge sort | Menor eficiencia | Menor eficiencia |
| Quick sort | Mayor eficiencia | Mayor eficiencia |

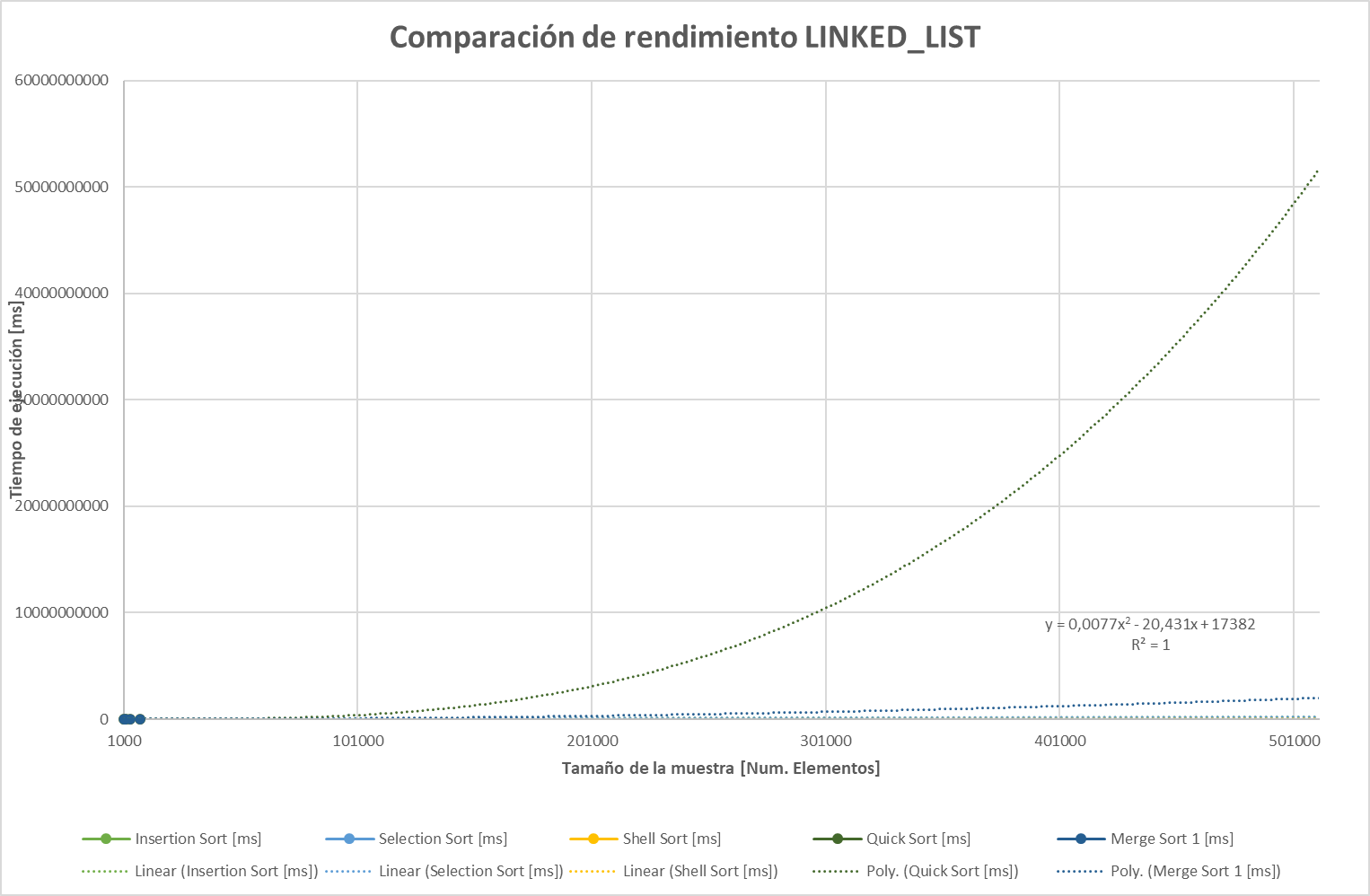
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

## **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?

Para el *MergeSort*, el comportamiento es de complejidad supeior al teórico. En ambos *Linked List* y *Array List* tiene una complejidad O(n2), pero en teoría debería ser O(nlog(n)).

Para el *QuickSort*, el comportamiento en ambos tipos de datos tiene una complejidad de O(n2), pero en teoría, el caso promedio debería ser O(nlog(n)). No obstante, el peor caso de *QuickSort* es O(n2) por lo que el algoritmo se comportó todas las veces como su peor caso, lo cual no es idea.

Cabe aclarar que ambos algoritmos se ejecutan mejor en *Array List* y tienen un tiempo de ejecución más similar al teórico, a pesar de tener un orden de complejidad distinto. Esto se debe a que ambos algoritmos están diseñados para *Array List*.

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

Ejecutar las pruebas en la máquina 2 resulta en tiempos de ejecución mucho mayores. Tanto es así que ni siquiera fue posible ejecutar pruebas con las mismas cantidad de datos antes de tener un tiempo de ejecución excesivo. Además, esto provoca que la línea de tendencia reporte una complejidad mayor, ya sea porque hay menos datos y la predicción es menos precisa, o porque con esta máquina se está más cerca al peor caso de los algoritmos de ordenamiento.

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

Las diferencias se deben a la velocidad relativa de los procesadores y la cantidad de memoria RAM. En particular, la velocidad del procesador de la máquina 1 es de 2.1 GHz y el de la máquina 2 es de 1.8 GHz. La máquina 1 tiene 12GB de memoria RAM y la máquina 2 tiene 8GM. La suma de ambas cosas resultan en tiempos de ejecución distintos y mayores en la máquina 2.

Como consecuencia de esto, la máquina 1 tiene un rendimiento superior, pues tiene un procesador con un reloj base de mayor velocidad y, además, tiene más espacio de memoria.

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

Es más rápido usa *Array List*, porque en todas las pruebas se ve que el tiempo de ejecución es sustancialmente menor.

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 más rápido es *QuickSort*, seguido en empate por *MergeSort* y *Shellsort* (porque sus tiempos de ejecución son muy similares), luego *InsertionSort* y, finalmente, *SelectionSort*.

Este orden aplica tanto para *Array List* como para *Linked List*, pero de tratarse de velocidad pura, es mejor utilizar *Array List*.