

# OBSERVACIONES DE LA PRACTICA

Juan David Salguero 201923136

David Molina 202125176

	Máquina 1	Máquina 2
Procesadores	AMD Ryzen 7	Intel Core i5
Memoria RAM (GB)	8GB	8GB
Sistema Operativo	Windows 11- 64 bits	macOs Catalina - 64 bits

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

## Maquina 1

### Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
0.50%	21	0000.0	0000.0	0000.0	0	0
5.00%	219	55	55	8	8	16
10.00%	439	199	202	24	13	18
20.00%	878	905	949	40	29	40
30.00%	1317	2183	2247	70	72	72
50.00%	2195	5634	5907	150	105	112
80.00%	3512	15566	13222	269	171	179
100.00%	4391	21598	25028	365	207	229

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

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
0.50%	21	1	0.000	0.000	2	8
5.00%	219	642	574	73	79	25
10.00%	439	5803	4.998	490	333	72
20.00%	878	24568	21599	1241	1821	241
30.00%	1317	117589	137353	3327	4592	557
50.00%	2195	627663	439717	10525	12650	1462
80.00%	3512	2072746	1753310	30413	32371	3713
100.00%	4391	tiempo excedido	tiempo excedido		53445	5547

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

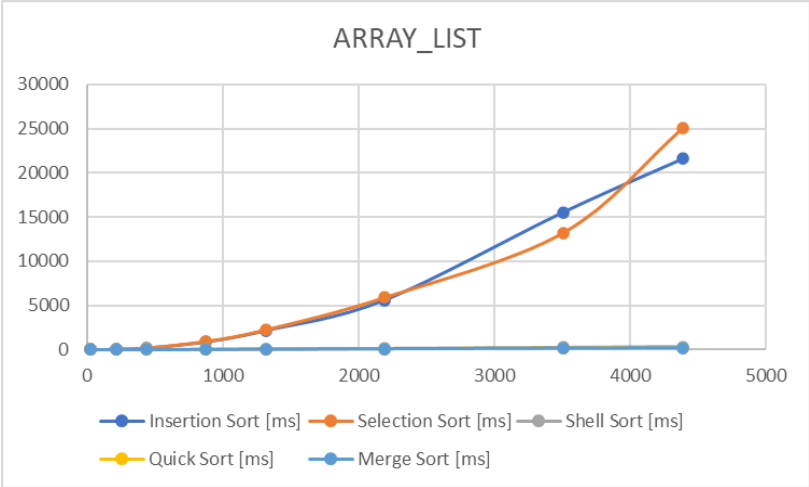
Algoritmo	Arreglo (ARRAYLIST)	Lista enlazada (LINKED_LIST)
Merge sort	Buen rendimiento	peor rendimiento

<b>Quick sort</b>	<b>Buen rendimiento</b>	<b>mal rendimiento</b>
-------------------	-------------------------	------------------------

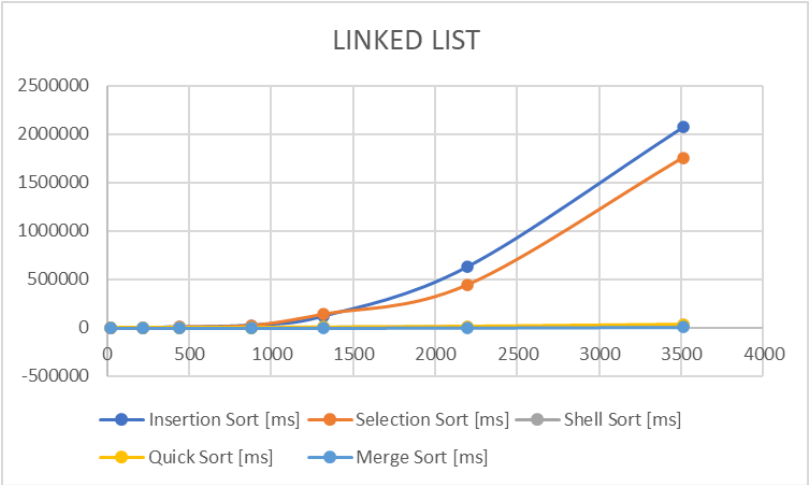
Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

Graficas

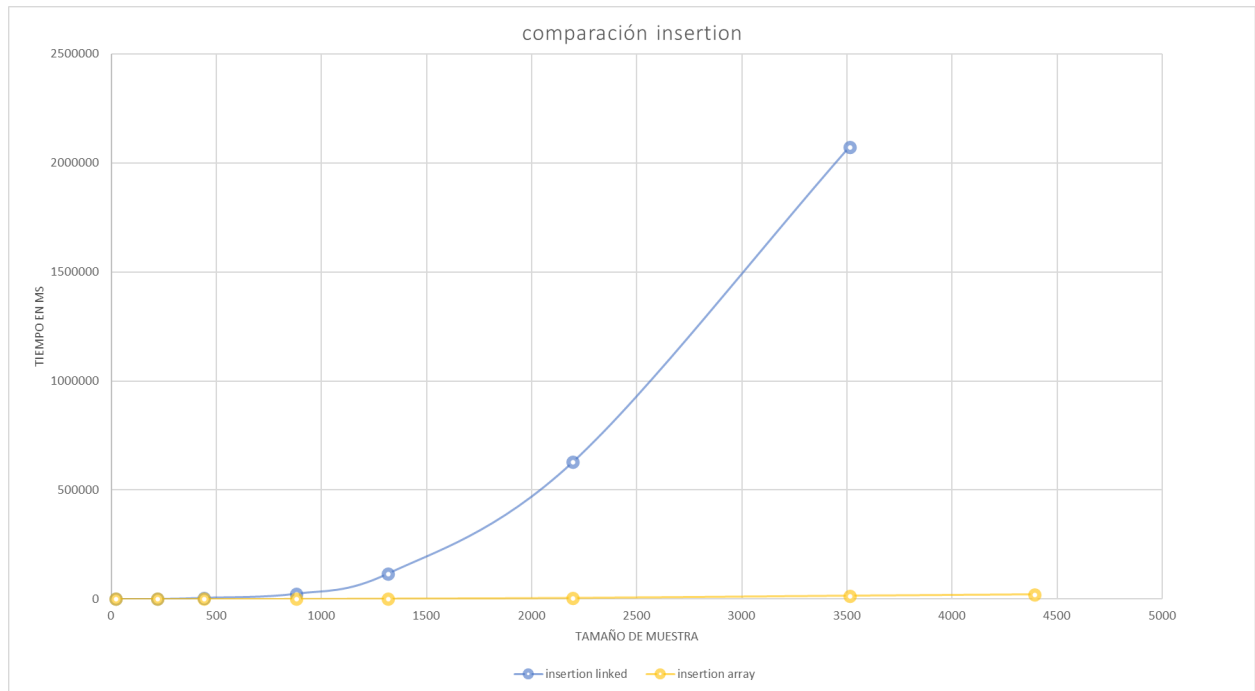
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la **Maquina 1.**
  - o Comparación de rendimiento ARRAYLIST.



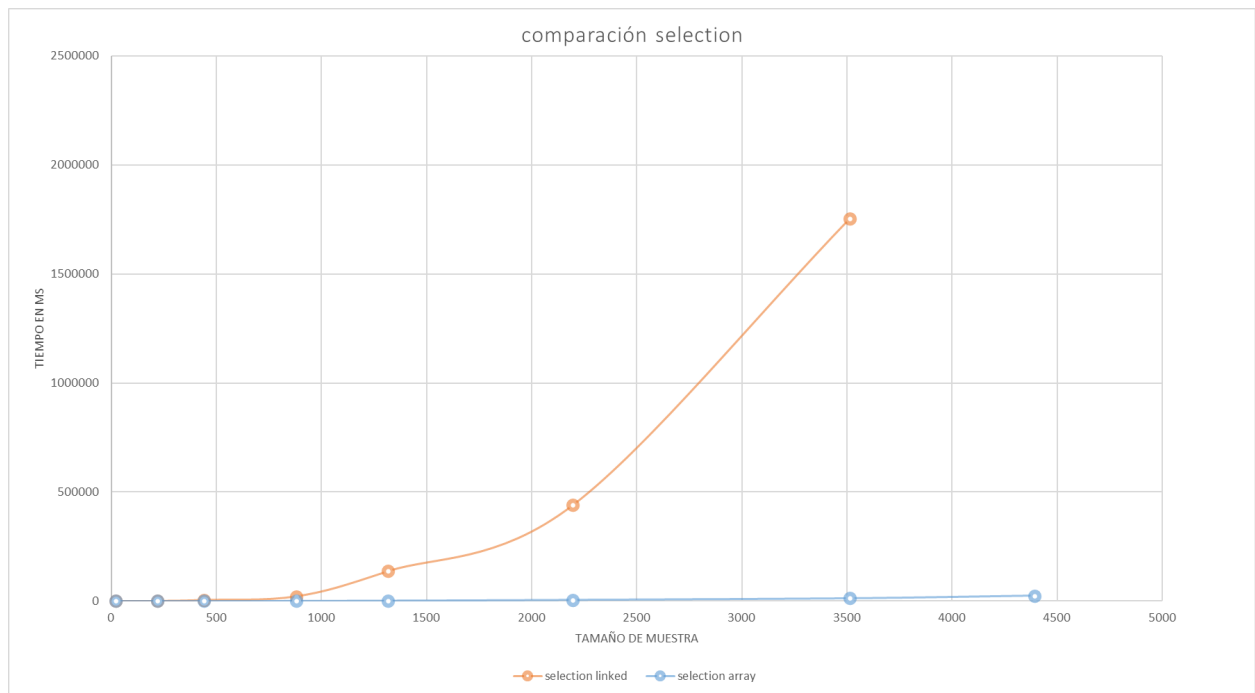
- o Comparación de rendimiento LINKED\_LIST.



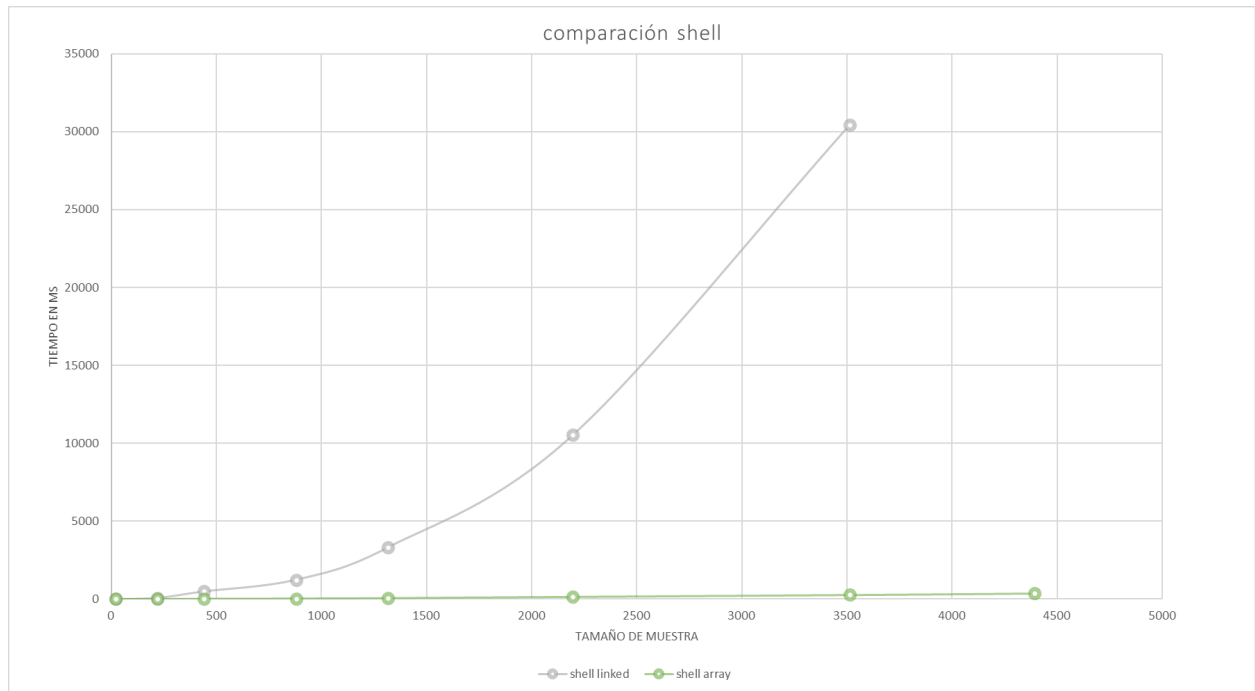
- o Comparación de rendimiento para Insertion Sort.



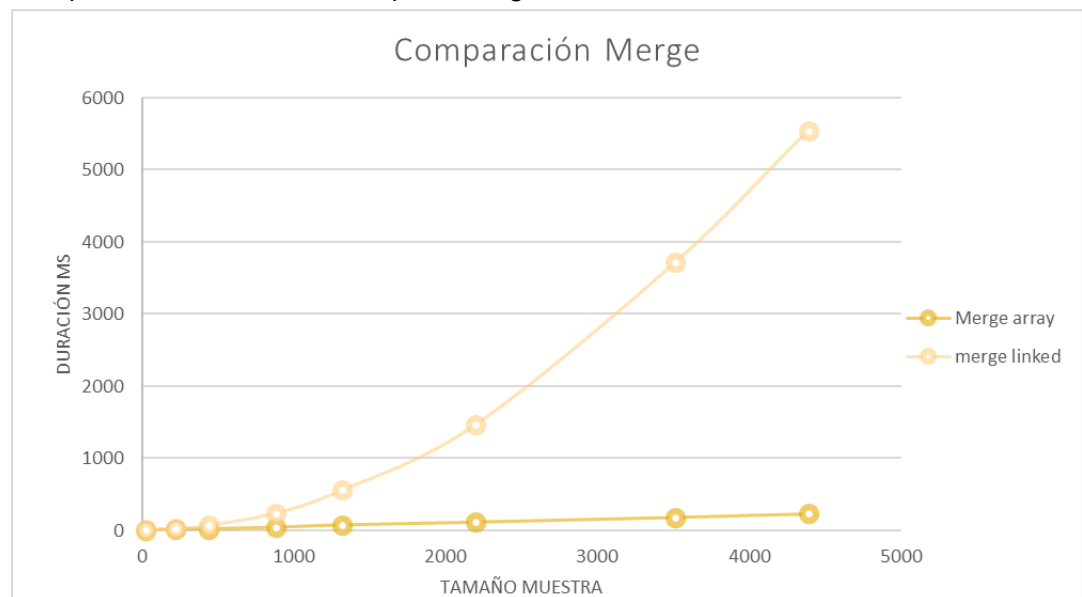
o Comparación de rendimiento para Selection Sort.



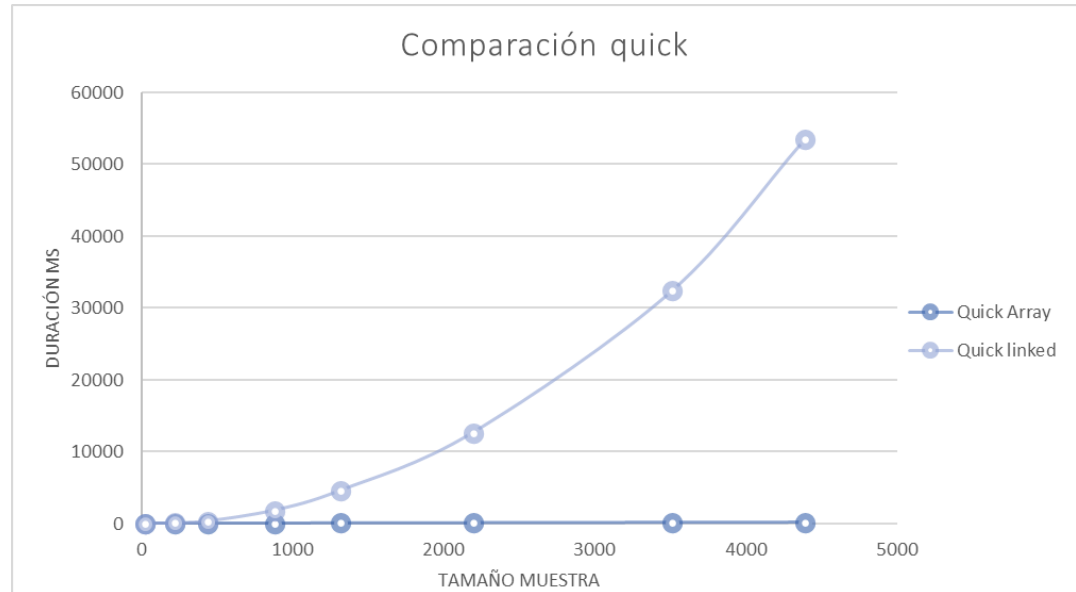
o Comparación de rendimiento para Shell Sort.



o Comparación de rendimiento para MergeSort



o Comparación de rendimiento para QuickSort.



## Maquina 2

### Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
0.50%	21	0.12	0.12	0.12	0.1	1.10
5.00%	219	46.28	53.32	7.83	7.04	11.34
10.00%	439	179.93	206.93	18.33	19.15	18.44
20.00%	878	713.72	831.12	43.46	43.45	43.54
30.00%	1317	1687.30	1893.22	70.76	76.44	70.54
50.00%	2195	4601.40	5171.57	137.78	101.56	103.11
80.00%	3512	11910.50	13540.35	246.89	170.42	167.85
100.00%	4391	19599.61	20990.44	331.53	200.01	209.36

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

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Insertion Sort [ms]	Selection Sort [ms]	Shell Sort [ms]	Quick Sort [ms]	Merge Sort [ms]
0.50%	21	0001.66	0001.31	0001.97	1.01	1.31
5.00%	219	0621.08	0581.06	0079.54	101.37	26.11
10.00%	439	4982.78	4467.77	0445.09	344.55	78.92
20.00%	878	39957.76	36945.41	2204.89	1849.39	244.56
30.00%	1317	142550.43	128764.76	5738.81	4827.96	527.88
50.00%	2195	666914.10	606240.90	18573.73	13142.97	1434.54

80.00%	3512	55005.90	33555.41	3659.82
100.00%	4391	89345.67	55737.78	6783.08

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	Mejor rendimiento por poco	Mejor rendimiento por mucho
Quick sort	Peor rendimiento por poco	Peor rendimiento por mucho

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

## Graficas

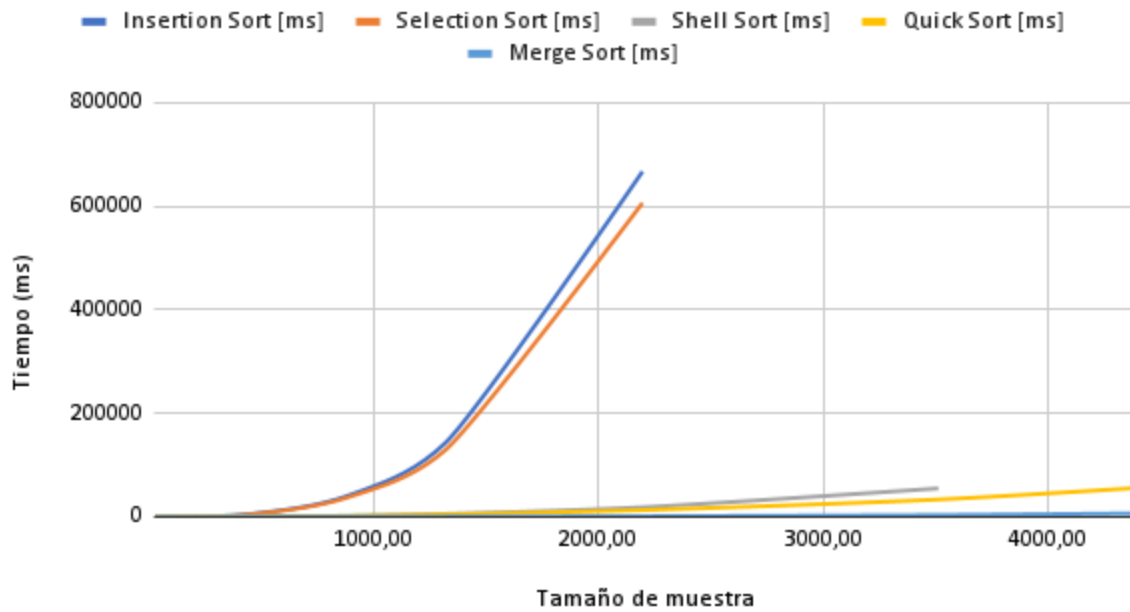
- Cinco gráficas generadas por los resultados de las pruebas de rendimiento en la
- **Maquina 2.**
  - o Comparación de rendimiento ARRAYLIST.

### Comparación ARRAY\_LIST



- o Comparación de rendimiento LINKED\_LIST.

## Comparación LINKED\_LIST



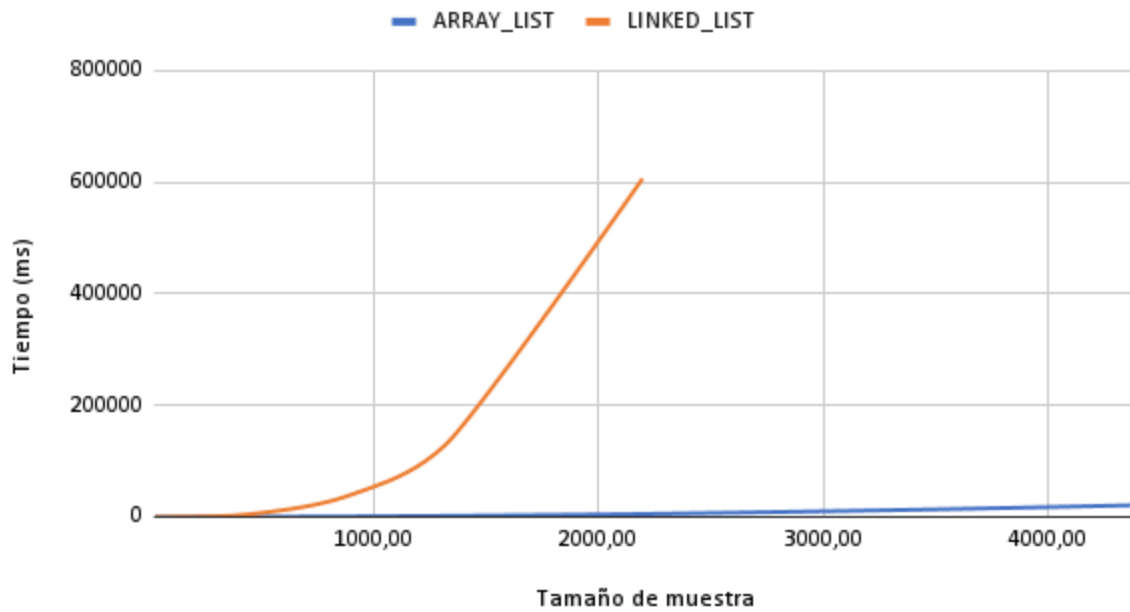
- o Comparación de rendimiento para Insertion Sort.

## Comparación Insertion Sort



- o Comparación de rendimiento para Selection Sort.

## Comparación Selection Sort



- o Comparación de rendimiento para Shell Sort.

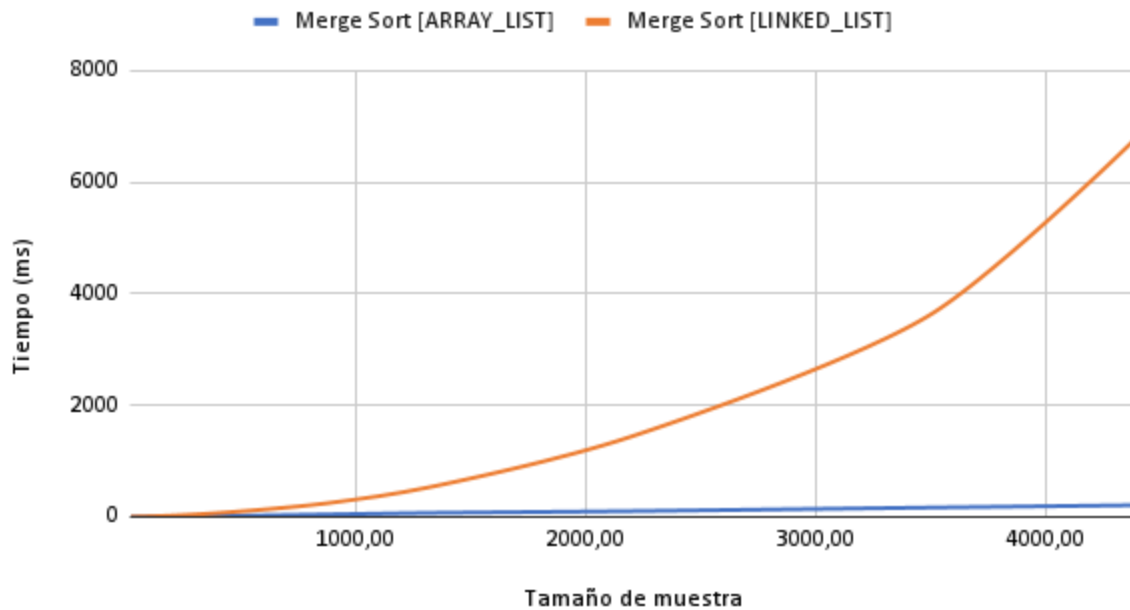
## Comparación Shell Sort



- o Comparación de rendimiento para MergeSort.

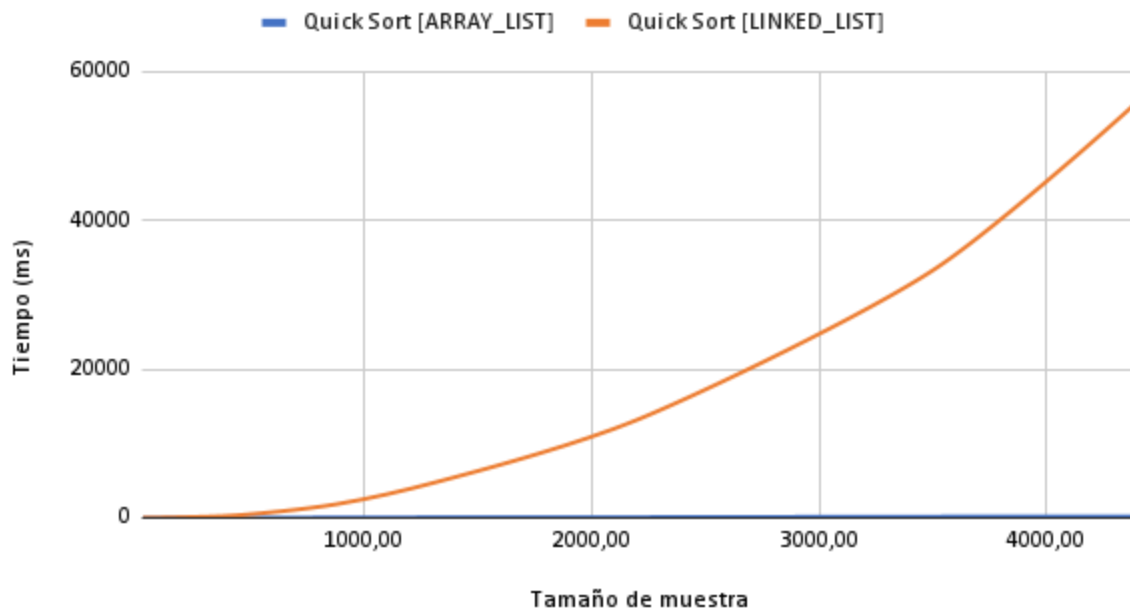


## Comparación Merge Sort



- o Comparación de rendimiento para QuickSort.

## Comparación Quick Sort



## Preguntas de análisis

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

Lo hizo. La eficiencia de los algoritmos recursivos parece ser mayormente logarítmica, y exponencialmente más rápida que los algoritmos iterativos de ordenamiento. Esto es notable especialmente en el caso de organizar una LINKED\_LIST con Merge Sort.

**2) ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?**

Los pocos cambios que existen son demasiado pequeños para ser remarcables, en general cada algoritmo de ordenamiento mantiene una eficiencia constante independientemente de la máquina.

**3) De existir diferencias, ¿A qué creen ustedes que se deben dichas diferencias?**

Estas minúsculas diferencias son probablemente causa de la diferencia del hardware entre las máquinas, y la capacidad de procesamiento disponible en cada una a la hora de realizar las pruebas de ordenamiento.

**4) ¿Cuál Estructura de Datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?**

Merge Sort es en retrospectiva el algoritmo de ordenamiento más eficaz. Aunque su velocidad similar a la de Quick Sort al trabajar con un ARRAY\_LIST, Merge Sort se destaca notablemente al ordenar una LINKED\_LIST.

**5) Para el caso analizado de ordenamiento de los artistas, teniendo en cuenta los resultados de tiempo reportados por todos los algoritmos de ordenamiento (iterativos y recursivos), proponga un ranking de los algoritmos de ordenamiento (de mayor eficiencia a menor (en tiempos de ejecución) para ordenar la mayor cantidad de artistas.**

Sin especificar la diferencia que el tipo de lista puede ejercer en el tiempo descrito, en términos generales según su promedio, el ranking es el siguiente:

- Merge Sort
- Quick Sort
- Shell Sort
- Insertion Sort
- Selection Sort