

OBSERVACIONES DE LA PRÁCTICA

Laura Sánchez 202411353

Sara Ramos Cod 202321247

Lina Diaz Cod 202321950

Ambientes de pruebas

	Máquina 1	Máquina 2	Máquina 3
Procesadores	12th Gen Intel(R) Core(TM) i7-1255U 2.60 GHz	12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz	Apple M3
Memoria RAM (GB)	8.00 GB (7.75 GB usable)	16,0 GB (15,6 GB usable)	8.00 GB
Sistema Operativo	64-bit operating system, x64-based processor	64 bits, procesador basado en x64	macOS 15.0

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

Máquina 1

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Selection Sort [ms]	Merge Sort [ms]
0.50%	50.00	1.201	1.203
5.00%	500.00	41	9.392
10.00%	1000.00	95	17.328
20.00%	2000.00	310	30.849
30.00%	3000.00	645	46.392
50.00%	5000.00	1,840	55.323
80.00%	8000.00	4,920	83.571
100.00%	10000.00	7,828	90.382

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

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Selection Sort [ms]	Merge Sort [ms]
0.50%	50.00	0.76	
5.00%	500.00	41.032	
10.00%	1000.00		101.392
20.00%	2000.00		310.896
30.00%	3000.00		590.546
50.00%	5000.00		1846.478
80.00%	8000.00		4893.674
100.00%	10000.00		7043.48

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

Algoritmo	Arreglo (ARRAY LIST)	Lista enlazada (LINKEDLIST)
<i>Selection Sort</i>	17,320,683 ms	17,379,930 ms
<i>Merge Sort</i>	98,290 ms	106,830 ms

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

Máquina 2

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAY_LIST)	Selection Sort [ms]	Merge Sort [ms]
0,50%	50,00	679	251
5,00%	500,00	38,171	3,963
10,00%	1000,00	144,265	12,877
20,00%	2000,00	541,334	27,303
30,00%	3000,00	1,349,726	33,872
50,00%	5000,00	4,014,688	38,335
80,00%	8000,00	10,651,896	51,36
100,00%	10000,00	16,244,771	96,766

Tabla 5. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKED_LIST)	Selection Sort [ms]
0,50%	50,00	662
5,00%	500,00	31,932
10,00%	1000,00	132,844
20,00%	2000,00	536,779
30,00%	3000,00	1,341,829
50,00%	5000,00	4,007,932
80,00%	8000,00	10,645,338
100,00%	10000,00	16,229,293

Tabla 6. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAY LIST)	Lista enlazada (LINKEDLIST)
-----------	----------------------	-----------------------------

<i>Selection Sort</i>	16,244,771 ms	16,229,293 ms
<i>Merge Sort</i>	96,766 ms	

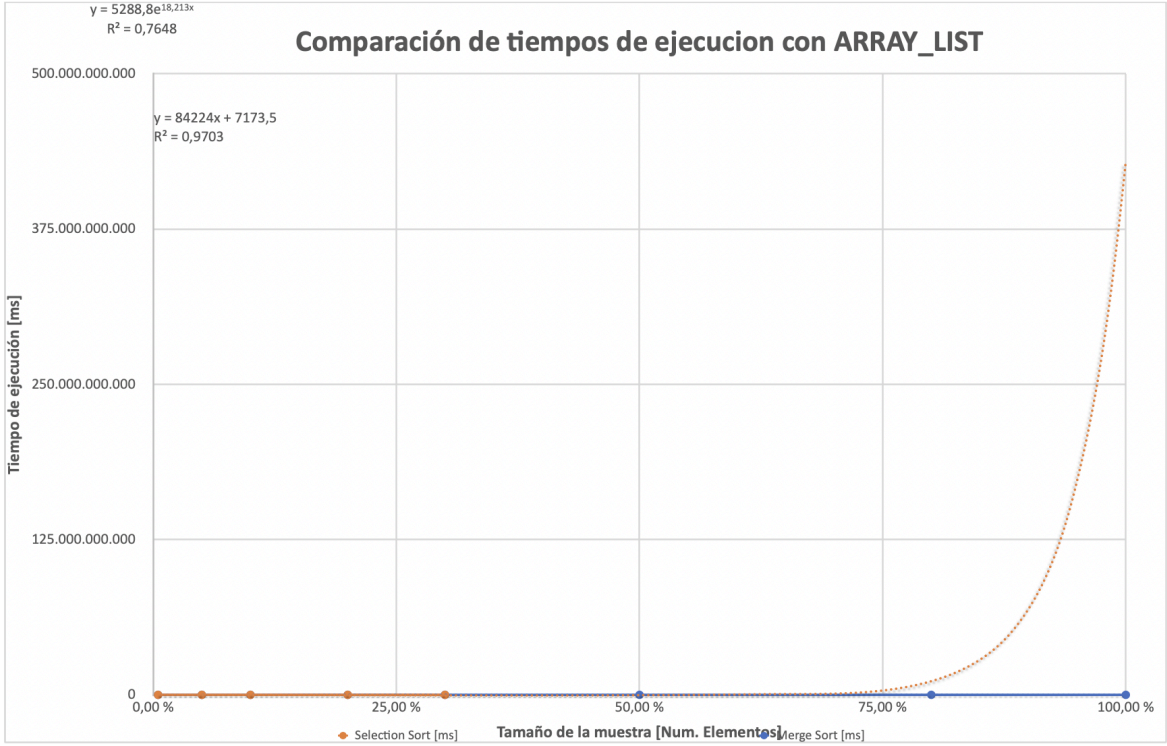
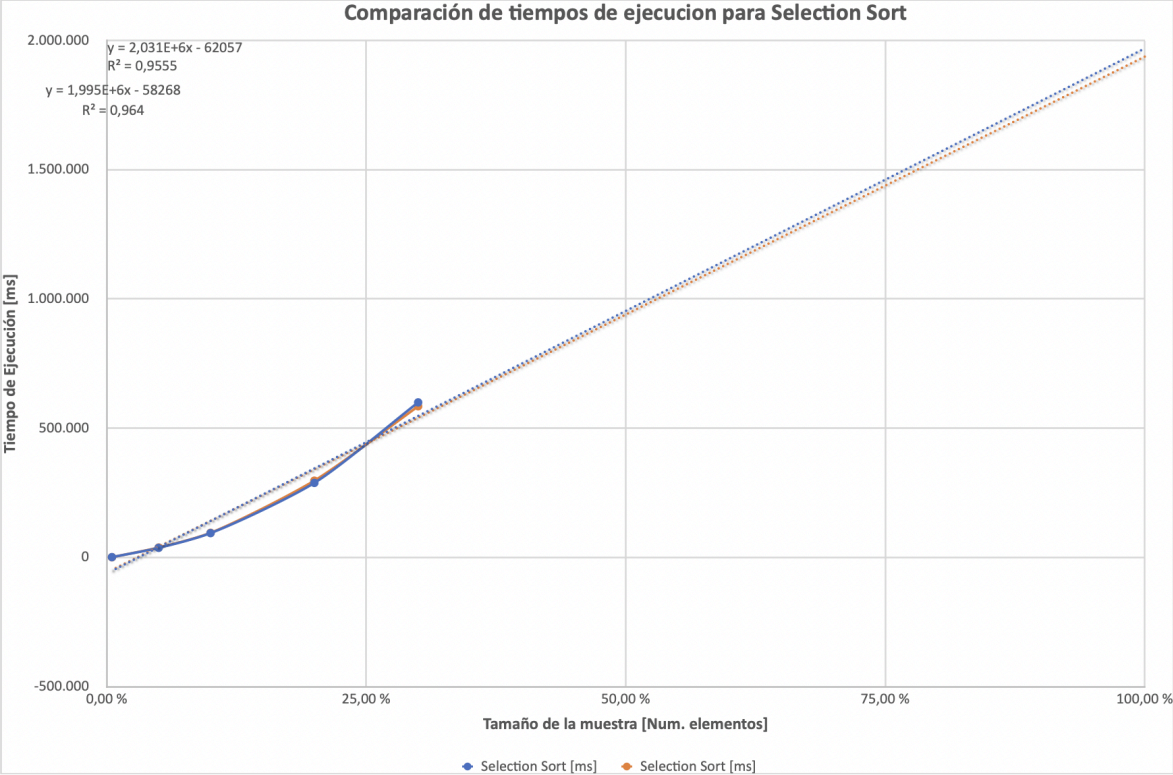
Tabla 7. Comparación de eficiencia de acuerdo con los algoritmos de ordenamiento y estructuras de datos utilizadas.

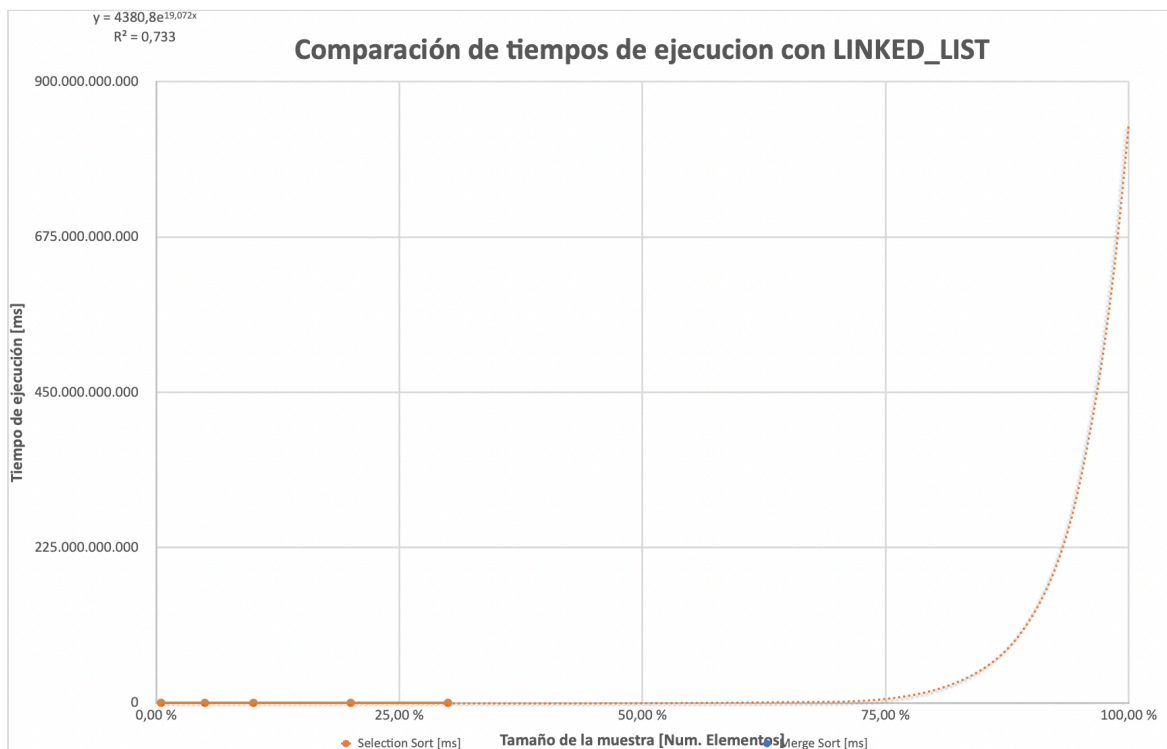
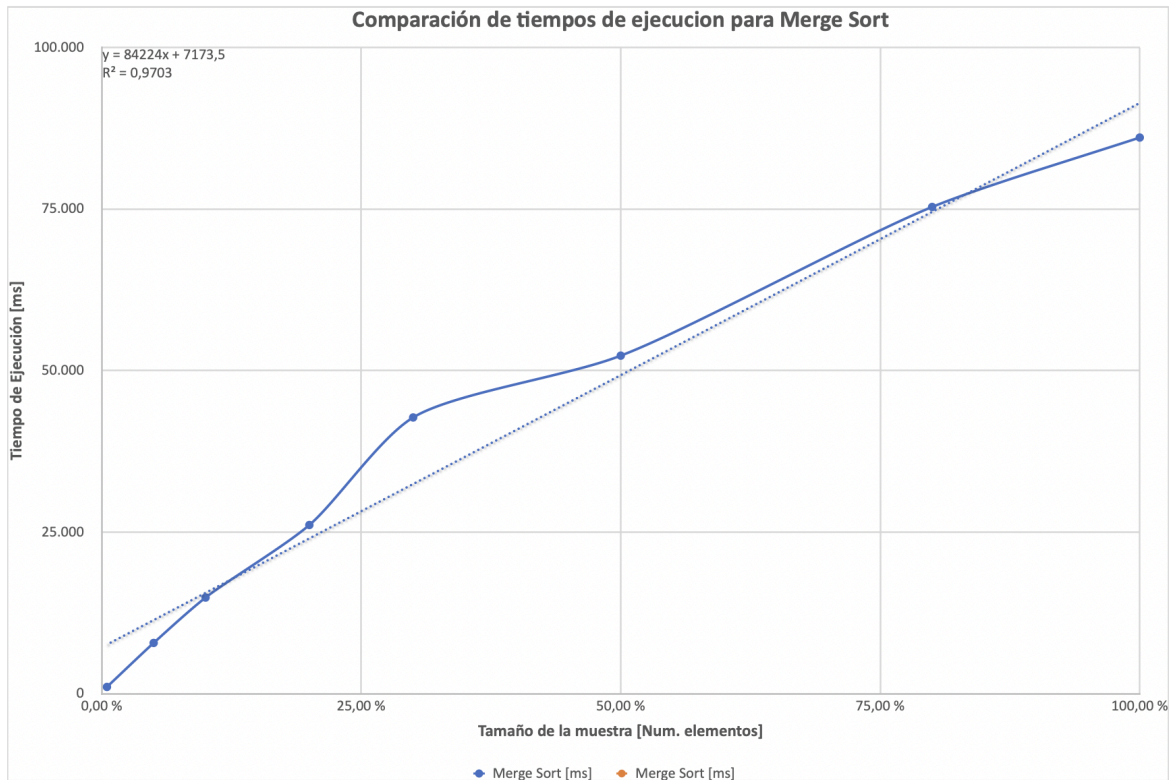
Máquina 3

Resultados

Porcentaje de la muestra [pct]	Tamaño de la muestra (ARRAYLIST)	Selection Sort [ms]	Merge Sort [ms]
0.50%	50,00	1.044	1.040
5.00%	500,00	36.998	7.848
10.00%	1000,00	94.010	14.870
20.00%	2000,00	288.229	26.098
30.00%	3000,00	599.705	42.740
50.00%	5000,00	1617.258	52.304
80.00%	8000,00	4146.016	75.314
100.00%	10000,00	6525.958	86.055

Tabla 8. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.





Porcentaje de la muestra [pct]	Tamaño de la muestra (LINKEDLIST)	Selection Sort [ms]	Merge Sort [ms]
--------------------------------	-----------------------------------	---------------------	-----------------

0.50%	50,00	681	
5.00%	500,00	38.172	
10.00%	1000,00	95.167	
20.00%	2000,00	296.922	
30.00%	3000,00	584.754	
50.00%	5000,00	1646.128	
80.00%	8000,00	4109.807	
100.00%	10000,00	6571.580	

Tabla 9. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

Algoritmo	Arreglo (ARRAY LIST)	Lista enlazada (LINKEDLIST)
<i>Selection Sort</i>		
<i>Merge Sort</i>		

Tabla 10. Comparación de eficiencia de acuerdo con los algoritmos de ordenamiento y estructuras de datos utilizadas.

Preguntas de análisis

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

Para el caso de Array List, la complejidad temporal de Selection Sort es $O(n^2)$, ya que el algoritmo realiza n búsquedas para encontrar el elemento mínimo en cada iteración. En este caso se cumple debido a que los tiempos crecen rápidamente con el tamaño de la muestra. Esto se puede evidenciar cuando pasa de 679 ms para 50 elementos a 16,244 ms para 10000 elementos. Merge Sort tiene una complejidad temporal de $O(n \log n)$, y es mucho más eficiente para Array porque los elementos pueden ser accedidos directamente por su índice en tiempo constante $O(1)$. Esto se refleja en los tiempos más bajos en la tabla para el ARRAY LIST, donde el tiempo fue de 96 ms para 10000 elementos.

Para el caso de single linked list, los tiempos de ejecución de Selection Sort también crecen exponencialmente a medida que el tamaño de la muestra aumenta. Asimismo,

Merge Sort tiene una complejidad temporal de $O(n \log n)$, pero en este caso es menos eficiente en comparación con el ARRAY LIST debido al acceso secuencial.

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

Sí, existen diferencias en los resultados obtenidos al ejecutar las pruebas en diferentes máquinas. Estas diferencias se deben a la variabilidad en las capacidades de hardware de las máquinas, como la velocidad del procesador, la cantidad de núcleos y la memoria RAM disponible. También influyen otros factores, como la carga del sistema y la cantidad de procesos que se están ejecutando simultáneamente en cada máquina, además de las optimizaciones específicas del sistema operativo y la versión de Python utilizada en cada entorno de ejecución.

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

Las diferencias que evidenciamos en los tiempos de ejecución al correr las pruebas en diferentes máquinas se deben a factores como la capacidad de procesamiento de los dispositivos, la cantidad de memoria disponible, el sistema operativo, y otros recursos hardware/software que dependen exclusivamente de cada máquina.

4) ¿Qué estructura de datos es mejor utilizar si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

Si solo se consideran los tiempos de ejecución de los algoritmos, la **Array List** es la estructura de datos preferida. Esto se debe a que permite el acceso directo a los elementos en tiempo constante $O(1)$, lo que mejora notablemente el rendimiento de algoritmos como **Merge Sort** y **Selection Sort**. En comparación, la **Single Linked List** tiene un acceso secuencial a los elementos, lo que introduce un retraso adicional en los tiempos de ejecución, especialmente en algoritmos que requieren muchas operaciones de acceso a los datos, como los ordenamientos.