Comparación de algoritmos de ordenamiento

Javier Cruz. Carné: C02517

En este trabajo se comparan diferentes algoritmos de ordenamiento con el objetivo de determinar cuál es el más efectivo, para determinarlo se tomó en cuenta el factor de la velocidad de ejecución. El resultado de dicha comparación fue la marcada diferencia que existe entre el tiempo de ejecución cuadrática con uno casi lineal. Gracias a esta comparación se puede concluir que la efectividad y rapidez de algoritmos como el de residuos es bastante superior a la de otros algoritmos como el de selección.

I. INTRODUCCIÓN

En el presente trabajo se encontrarán las diferencias en tiempo de ejecución de diferentes algoritmos de ordenamiento como *selección, inserción, mezcla, montículos, rápido y residuos.* Se desarrollaron diferentes ejecuciones de los algoritmos, tomado nota de sus tiempos para posteriormente compararlos. En el documento se encontrarán las siguientes partes: metodología, resultados y conclusiones.

II. METODOLOGÍA

Para llevar a cabo este trabajo se utilizó el lenguaje de programación C++ con el IDE Visual Studio Code. En una computadora de escritorio con las siguientes especificaciones: Procesador AMD Ryzen 5 2600, 16 GB de memoria RAM y tarjeta de video NVIDIA GeForce GTX 1650 Super.

En el código se crearon 4 arreglos a ser ejecutados por cada algoritmo de ordenamiento. El tamaño de los arreglos fue de 50.000, 100.000, 150.000 y 200.000 elementos. Los arreglos contenían números enteros tanto positivos como negativos generados aleatoriamente.

Una vez finalizadas las ejecuciones de los arreglos para cada algoritmo, se tomó nota de los tiempos para posteriormente crear tablas comparativas y gráficos en Microsoft Excel. El resultado de dicho cuadro es el siguiente:

Cuadro 1
Tiempo de ejecución de los algoritmos

		Tiempo (ms)			
		Corrida			
Algoritmo	Tam. (k)	1	2	3	Promedio
Selección	50	2670	2666	2669	2668.33
	100	10696	10696	10690	10694.00
	150	24048	23998	24002	24016.00
	200	42750	42701	42739	42730.00
Inserción	50	2169	1721	1728	1872.67
	100	6934	6925	6900	6919.67
	150	15555	15511	15551	15539.00
	200	27669	27624	27567	27620.00
Mezcla	50	8	8	8	8.00
	100	17	17	17	17.00
	150	27	27	27	27.00
	200	37	37	37	37.00

El cuadro compara los tiempos de ejecución en milisegundos de los algoritmos de selección, inserción y mezcla luego de usarse con arreglos de diferentes tamaños.

Cuadro 2
Tiempo de ejecución de los algoritmos

		Tiempo (ms)			
		Corrida			
Algoritmo	Tam. (k)	1	2	3	Promedio
Montículos	50	14	14	15	14.33
	100	30	30	31	30.33
	150	47	47	48	47.33
	200	66	65	66	65.67
Rápido	50	8	8	9	8.33
	100	18	18	18	18.00
	150	35	31	29	31.67
	200	41	41	45	42.33
Residuos	50	3	3	3	3.00
	100	6	7	7	6.67
	150	10	10	10	10.00
	200	15	14	14	14.33

El cuadro compara los tiempos de ejecución en milisegundos de los algoritmos de ordenamiento por montículos, rápido y residuos luego de usarse con arreglos

III. RESULTADOS

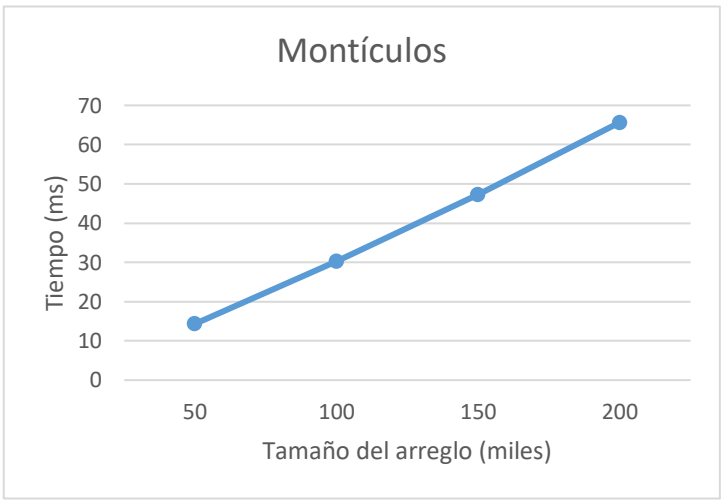
Los tiempos de ejecución de las corridas de los algoritmos se muestran en el cuadro 1. Aquí podemos apreciar cómo los tiempos de ejecución varían dependiendo del algoritmo que se utilice. Siendo el algoritmo por selección el más "lento" y siendo el de residuos el más "rápido". No hay una diferencia muy grande entre las corridas del mismo algoritmo pero si hay una diferencia notable cuando cambiamos de algoritmo.

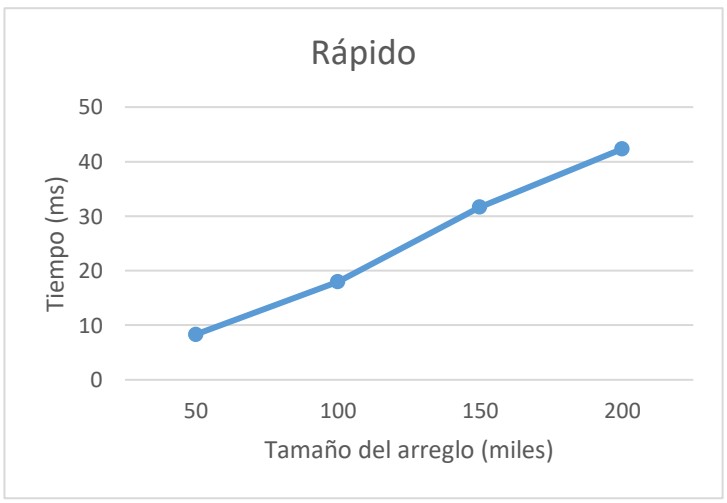
Para poder apreciar la diferencia de una forma visual se presentan los siguientes gráficos individuales de cada algoritmo, así como dos gráficos de los tres algoritmos unificados, uno con escala normal y otro con escala logarítmica.

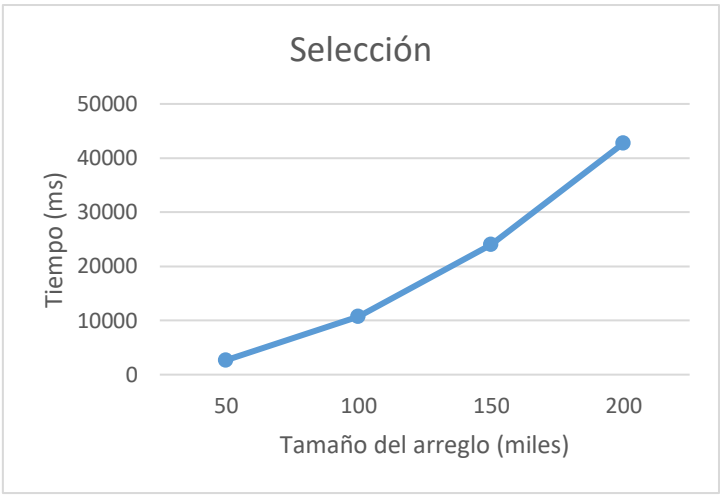
Los tiempos promedio se muestran gráficamente en la figura 1. Personalmente pensaba que las curvas de los algoritmos $\Theta(n^2)$ iban a ser ascendentes pero no esperaba que lo fueran de una forma tan "parabólica", viendo los gráficos podemos apreciar que no aumenta de una forma lineal, sino que cada vez la línea se va curvando más, contrario de lo que podemos apreciar en el gráfico del algoritmo de ordenamiento por mezcla, aquí vemos una línea recta, es decir, el tiempo aumenta de forma lineal, si esperaba que el algoritmo de mezcla fuera más rápido pero no esperaba una diferencia tan marcada.

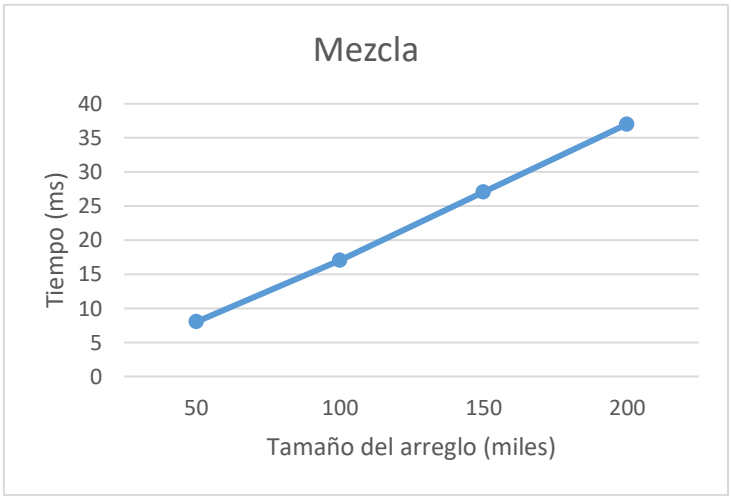
Con respecto a los algoritmos de ordenamiento por montículos, rápido y residuos resultan ser mucho más veloces o efectivos que los anteriores 3, al tener una escala de tiempo prácticamente lineal. Esto lo podemos apreciar en la gráfica y también en el cuadro de tiempos, donde el tiempo aumenta constantemente de acuerdo a como aumenta el tamaño del arreglo.

Las curvas de los algoritmos se muestran de forma conjunta en la figura 2. Aquí es donde más se puede apreciar la diferencia entre los algoritmos, en donde hace falta utilizar una escala logarítmica para poder reconocer el tiempo del algoritmo de ordenamiento por mezcla.









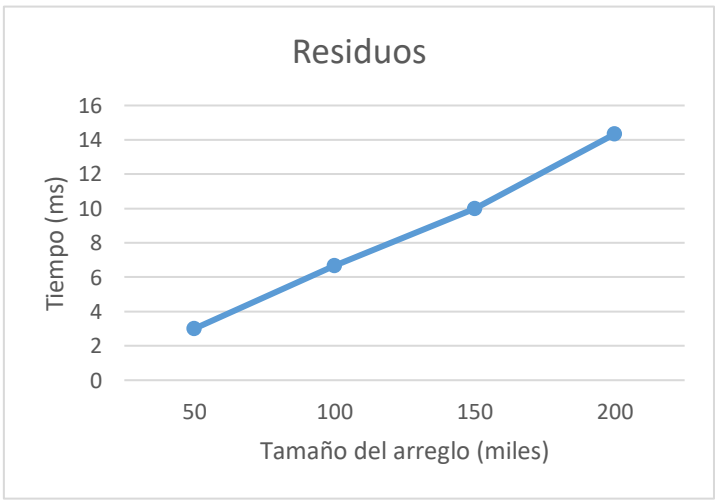
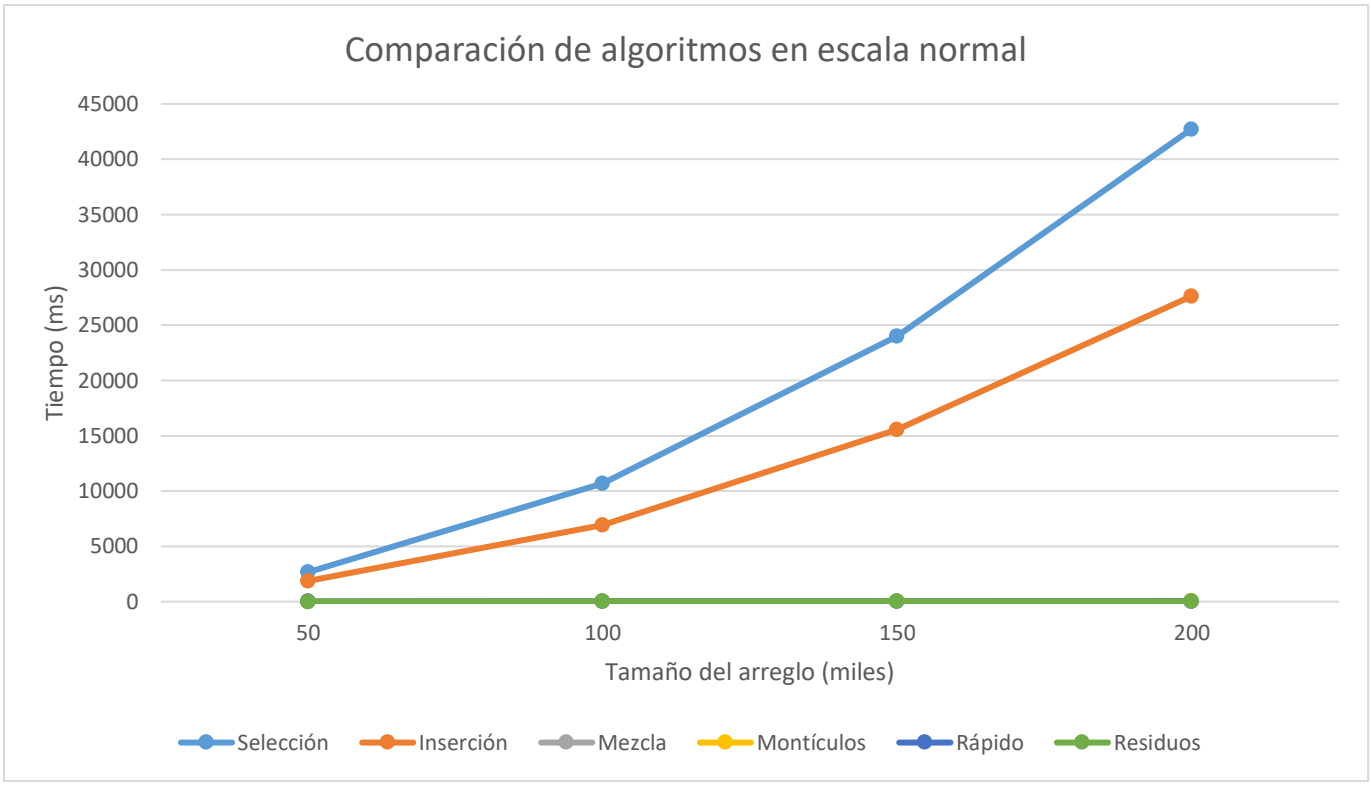


Figura 1. Tiempos promedio de ejecución de algoritmos de ordenamiento por selección, inserción, mezcla, montículos, rápido y residuos.



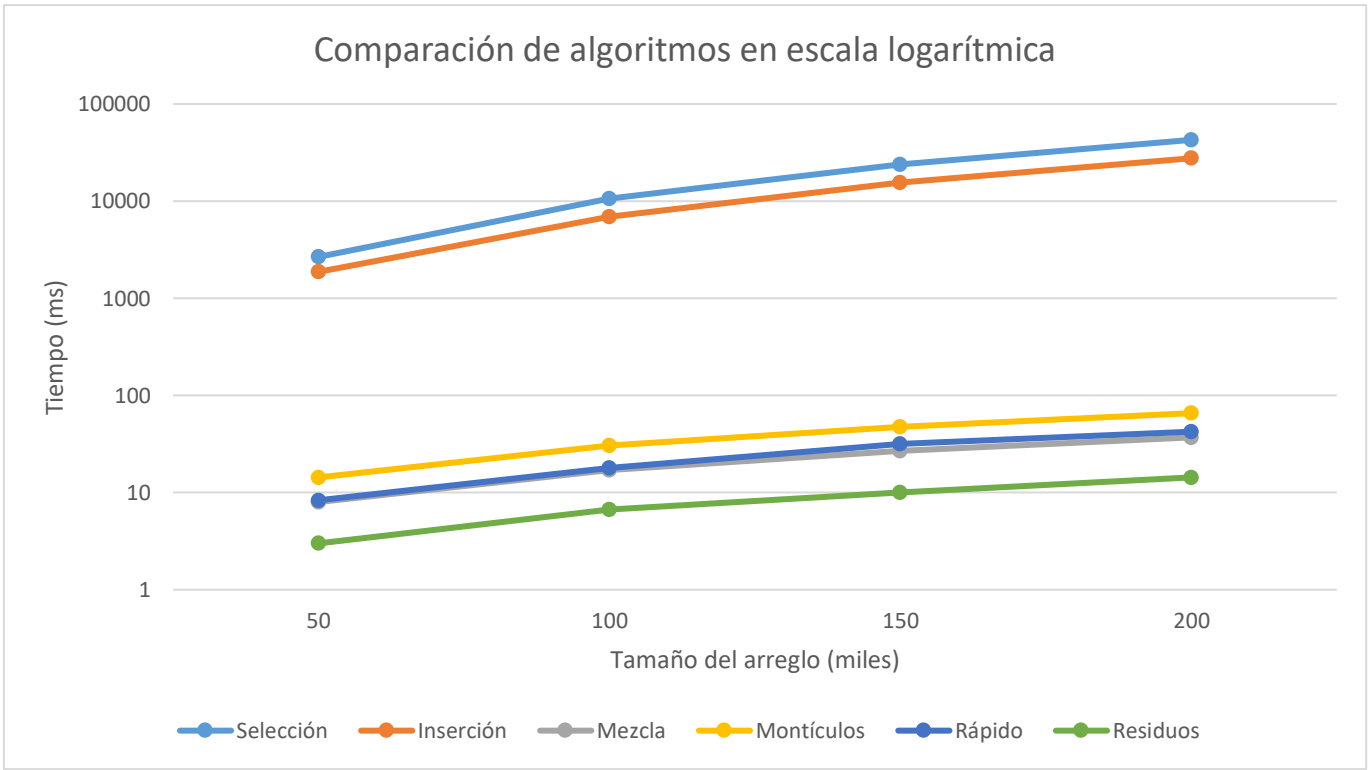


Figura 2. Grafico comparativo de los tiempos promedio de ejecución de los algoritmos.

IV. Conclusiones

A partir de los resultados obtenidos se concluye que existe una diferencia muy marcada entre los tiempos de ejecución de los algoritmos de ordenamiento por selección, inserción, mezcla, montículos, rápido y por residuos.

Siendo el algoritmo de ordenamiento por residuos el más rápido de los presentados en este trabajo.

Cabe destacar que el algoritmo que se utilice siempre depende de cada caso específico con el que se esté trabajando y las restricciones que se tengan a la hora de programar. Sin embargo, al tratarse de analizar el tiempo de ejecución y eficiencia de los algoritmos, podemos decir que el algoritmo de ordenamiento por residuos es efectivamente más rápido.