

Análisis de Rendimiento: QuickSort

Cristofer Patricio

Carnet: 202500553

Universidad Da Vinci de Guatemala

Facultad de Ingeniería en Sistemas

22 de febrero de 2026

Introducción

El presente documento analiza el rendimiento del algoritmo *QuickSort*, uno de los métodos de ordenamiento más eficientes basados en la técnica de *divide y vencerás*. El algoritmo selecciona un elemento como "pivote" y partitiona el arreglo alrededor del mismo, ubicando los elementos menores a la izquierda y los mayores a la derecha, resolviendo recursivamente los subarreglos.

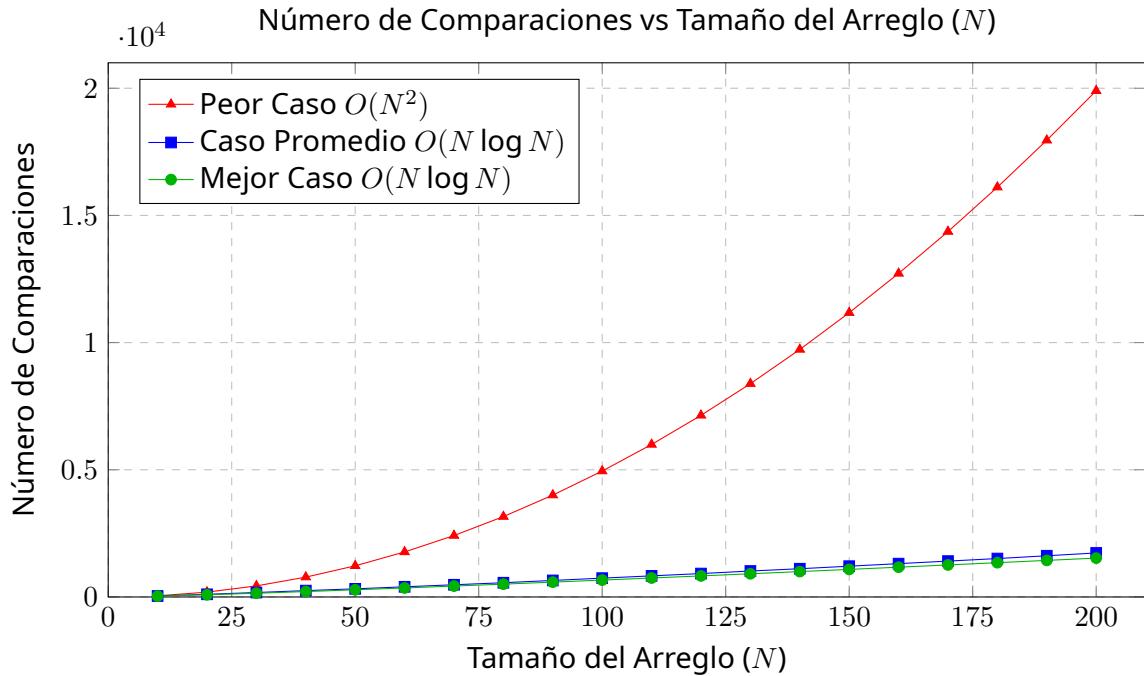
Metodología y Escenarios Evaluados

A diferencia de algoritmos menos eficientes como Bubble Sort, QuickSort tiene un comportamiento altamente dependiente de la elección del pivote. Para esta gráfica, utilizando el último elemento como pivote, se evaluaron **tres escenarios** variando el tamaño del arreglo (N) desde 10 hasta 200 elementos:

- **Peor Caso (Arreglo Ya Ordenado o Invertido):** Cuando se elige consistentemente el extremo como pivote en datos ya ordenados, el arreglo se divide de la forma más desequilibrada posible (1 elemento de un lado, $N - 1$ del otro).
- **Caso Promedio (Datos Aleatorios):** Los elementos están distribuidos al azar. Las particiones son razonablemente equilibradas.
- **Mejor Caso (Partición Perfecta):** El pivote elegido resulta ser la mediana exacta en cada paso, dividiendo el arreglo siempre en dos mitades iguales.

Gráfica de Resultados

A continuación se presenta la gráfica comparativa de comparaciones en los tres escenarios descritos para QuickSort.



Conclusión

A partir del análisis de la gráfica, podemos extraer las siguientes conclusiones sobre la naturaleza recursiva del algoritmo QuickSort:

- **La Eficiencia del Promedio ($O(N \log N)$):** Se observa claramente que, tanto en el **Mejor Caso** (verde) como en el **Caso Promedio** (azul), la curva se aplasta hacia el eje X. Esto demuestra el inmenso poder de lograr tiempos $O(N \log N)$ al particionar la carga de trabajo, volviéndolo drásticamente más rápido que los algoritmos cuadráticos en la práctica diaria.
- **La Debilidad del Peor Caso ($O(N^2)$):** La línea roja evidencia la gran vulnerabilidad del algoritmo si se elige un mal pivote (como el último elemento en un arreglo que ya se encuentra ordenado). En esta situación, QuickSort degenera y hace tantas comparaciones como un Bubble Sort clásico.
- **Impacto Real:** En entornos reales de la Ingeniería en Sistemas, se suelen implementar versiones mejoradas de QuickSort (como elegir el pivote aleatoriamente o usar la "mediana de tres") precisamente para evitar que la gráfica salte accidentalmente a la línea roja, asegurando siempre el rendimiento fenomenal de la línea azul.