

Conclusiones Generales:

1. Los métodos de ordenamiento (Quick Sort, Merge Sort) son eficaces y cumplen su objetivo, gracias a la garantía del tiempo de ejecución que **Merge Sort** ofrece, lo hace confiable, predecible y estable. Sin embargo, el tiempo de ejecución de **Quick Sort** puede verse afectado por la elección del pivote, generando más variabilidad.
2. Son útiles para el manejo de grandes volúmenes de datos. Estos métodos son capaces de manejar de manera eficiente miles de datos; sin embargo, en el caso de **Quick Sort**, esta eficiencia puede disminuir en sus casos desfavorables, como el pivote en medio.
3. El mejoramiento y optimización de los métodos “clásicos” de ordenamiento permiten el manejo de volúmenes aún más grandes de datos.
4. A medida que el número de entradas de datos aumenta, el tiempo promedio de los métodos de ordenamiento también aumenta.
5. A mayor rango de distribución de números aleatorios se evitan ambigüedades en la ejecución de los métodos de ordenamiento y se reduce su tiempo de ejecución.
6. El tiempo medido en nanosegundos presenta una variabilidad baja.
7. Las pruebas con 1×10^9 y 1×10^{10} no pudieron ser realizadas manteniendo la estructura clásica de los métodos debido a las interrupciones forzadas por parte del sistema operativo, debido a la limitación de recursos (memoria). El uso de mucha memoria y el procesamiento intensivo pueden provocar interrupciones por falta de recursos, lo que impide que los algoritmos se completen a tiempo o de manera eficiente.
8. El comportamiento del tiempo de ejecución del método de ordenamiento **Quick Sort** con pivote aleatorio y al final fue más estable que el método de ordenamiento **Quick Sort** con pivote en medio.