

# Análise comparativa entre algoritmos de ordenação

Gabriel Martins Nunes da Silva

<sup>1</sup>Pontifícia Universidade Católica do Paraná (PUCPR)  
Curitiba – PR – Brasil

**Resumo.** *Este documento visa realizar a análise comparativa entre três diferentes métodos de ordenação: "Bubblesort", "Quicksort" e "Mergesort". Para realização da análise, será levado em consideração os fatores: quantidade de iterações, quantidade de trocas e tempo de execução. Considera-se também que todos os testes foram realizados no mesmo dispositivo.*

## 1. Repositório

Link de acesso para o repositório no GitHub:

<https://github.com/gabriel-mns/estrutura-dados-metodos-ordenacao>

## 2. Informações gerais

Para a realização de todos os testes de performance dos algoritmos de ordenação, foi utilizado um computador com as seguintes especificações:

- **Processador:** Intel Core I7-8565U 1.8GHz,
- **RAM:** 20Gb,
- **SSD:** 500Gb

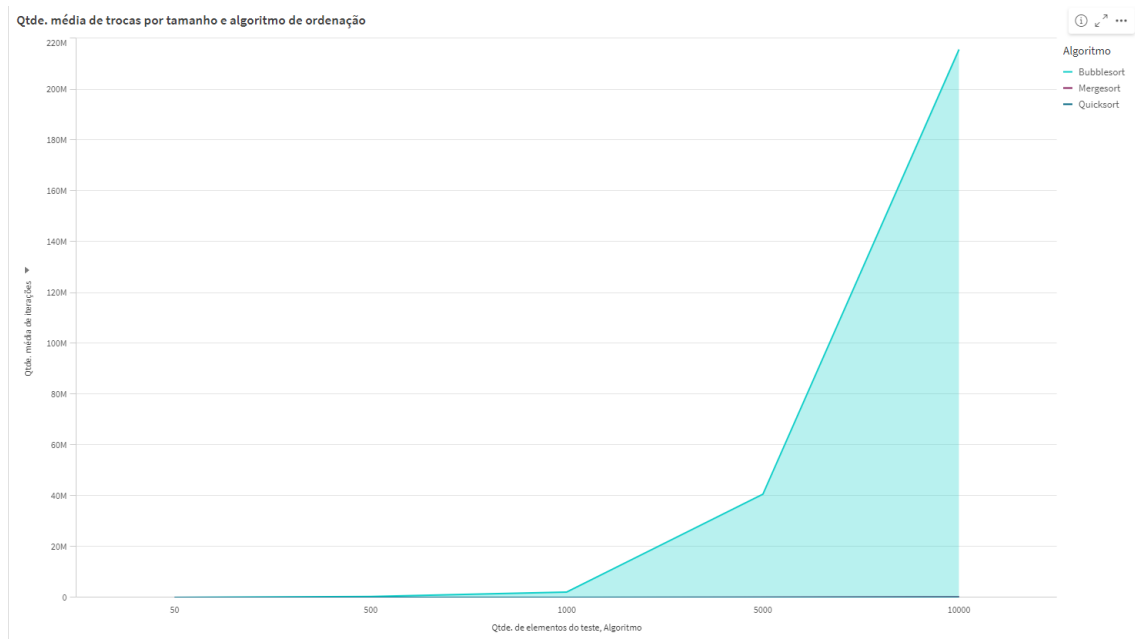
O cálculo da média dos resultados das operações foram feitos com base em 5 execuções sequenciais dos respectivos algoritmos.

## 3. Resultados obtidos

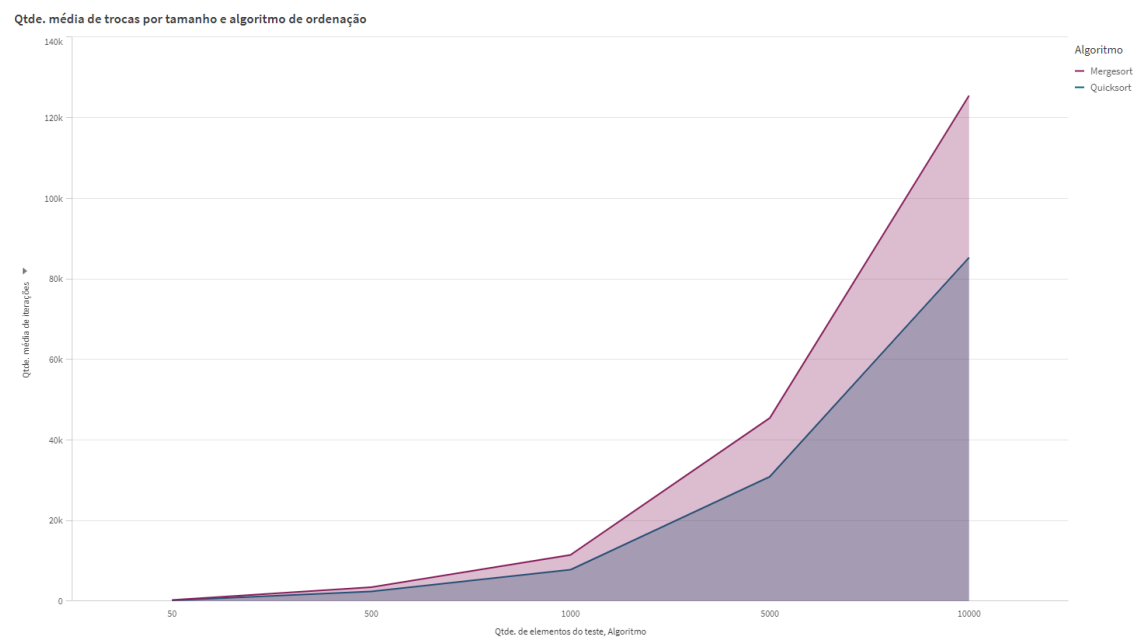
Após a realização dos testes, é possível analisar que o algoritmo do Bubblesort se mostra muito mais ineficiente quando comparado com os outros algoritmos, como é possível verificar na Figura 1. Já entre os algoritmos Mergesort e Quicksort o quicksort se mostra substancialmente mais performático do que o Mergesort, como é possível verificar na Figura 2. O mesmo padrão de resultados é encontrado ao analisar os resultados de tempo de execução de cada um dos algoritmos de ordenação.

## 4. Análise dos resultados

Após analisar os resultados, pode-se observar que o quicksort se mostra como o método mais eficiente dentre os 3 apresentados. Um dos fatos que pode ser responsável por isso, é que o Mergesort realiza a criação de 2 novos arrays a cada iteração, enquanto o Quicksort apenas utiliza um único array para ser organizado, sem ser necessário a criação de um novo array. Isso faz com que menos memória seja consumida durante a sua execução. Além disso, o Bubblesort realiza uma quantidade muito maior de iterações, comparações e trocas quando comparadas ao Quicksort. Todos esses fatores tornam o Quicksort o mais eficiente dentre os três.



**Figure 1. Qtde. Média de trocas por tamanho (todos os algoritmos)**



**Figure 2. Qtde. Média de trocas por tamanho (Quicksort e Mergesort)**