Análise dos algoritmos

-Análise dos QuickSorts

-Tabelas

As tabelas a seguir foram criadas usando os dados retornados por cada tipo de QuickSort com N variando entre 1000, 5000, 10000, 50000, 100000, 500000 e 1000000.











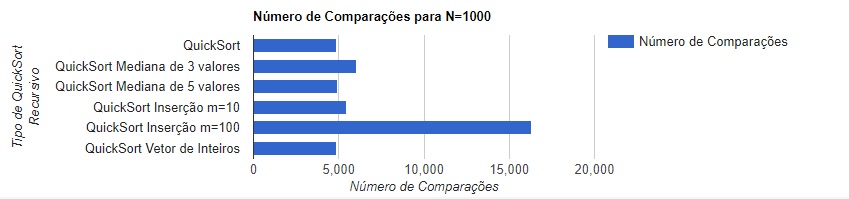


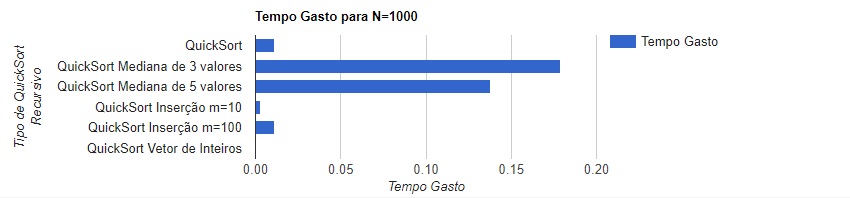


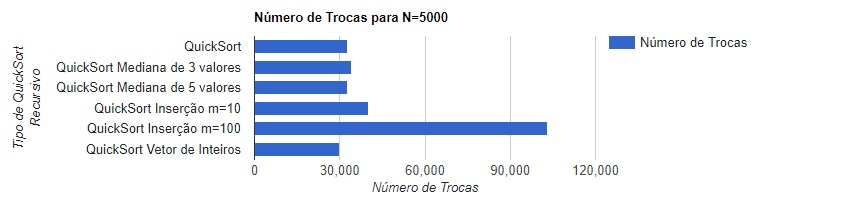
-Gráficos

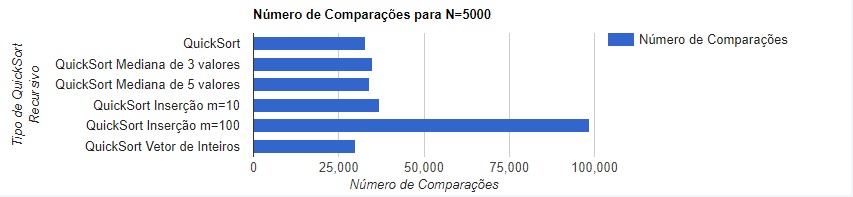
Os gráficos a seguir foram criados usando os dados retornados por cada tipo de QuickSort com N variando entre 1000, 5000, 10000, 50000, 100000, 500000 e 1000000.

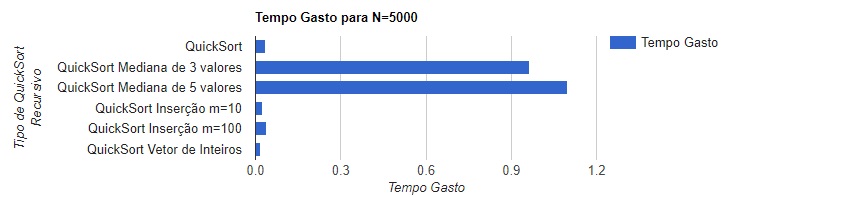


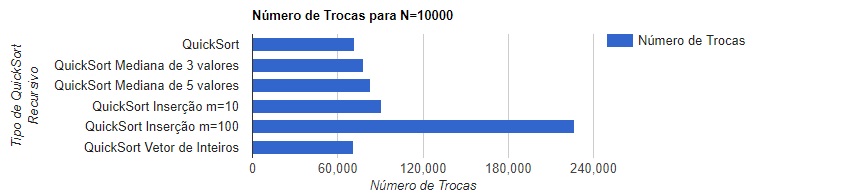




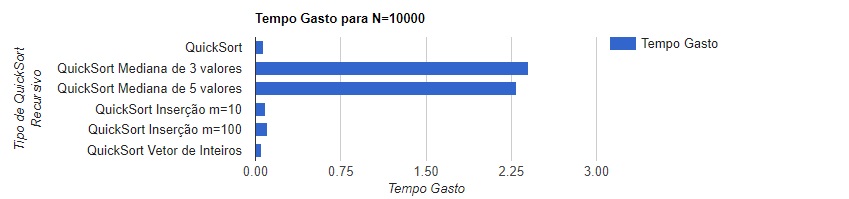




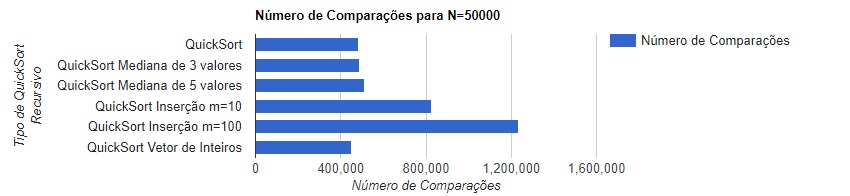




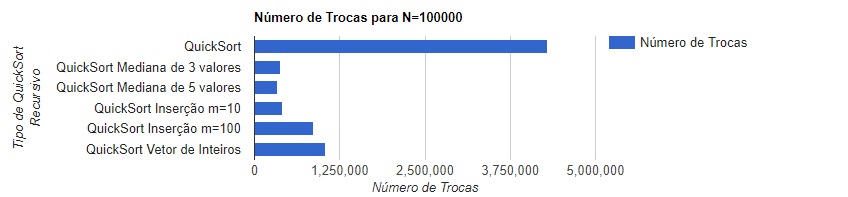


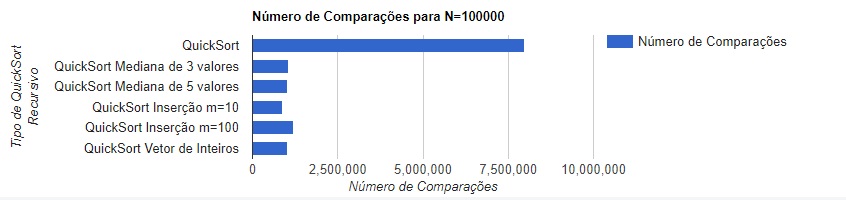






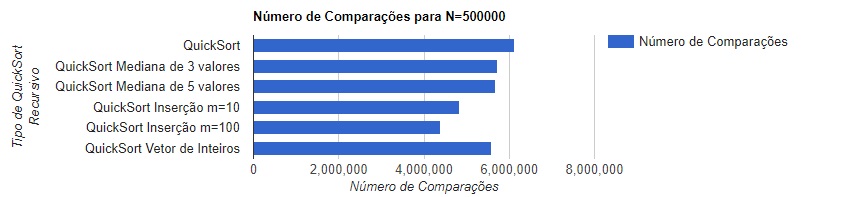


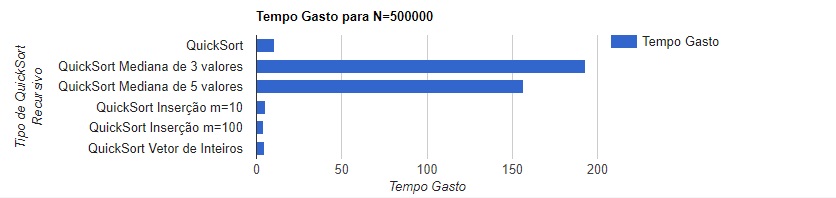


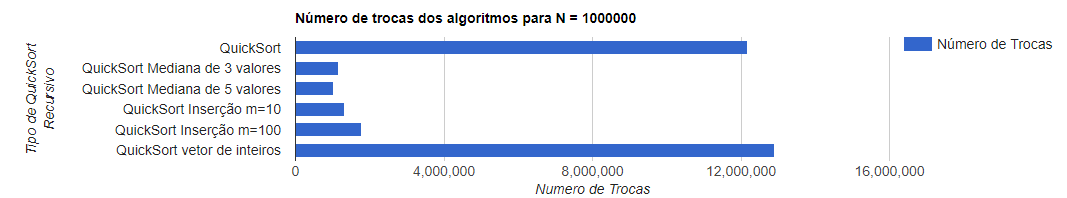


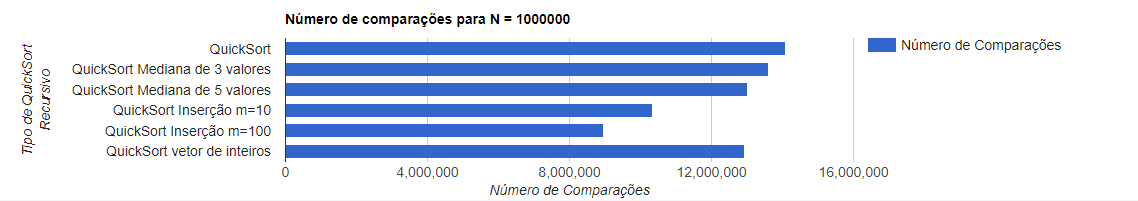














-Análise dos Algoritmos de Ordenação

-Tabelas

As tabelas a seguir foram criadas usando os dados retornados por cada algoritmo de ordenação com N variando entre 1000, 5000, 10000, 50000, 100000, 500000 e 1000000.

De acordo com o desempenho o QuickSort escolhido para comparação foi o Algoritmo QuickSort Recursivo com Insercao com m=10.

Os algoritmos InsertionSort e BubbleSort apresentam alguns dados iguais a 0 a partir de N=100000, isso ocorreu devido ao tamanho do inteiro retornado durante os testes, eles apresentavam valores maiores do que o aceitável pelo tipo de variável int.















-Gráficos

Os gráficos a seguir foram criados usando os dados retornados por cada algoritmo de ordenação com N variando entre 1000, 5000, 10000, 50000, 100000, 500000 e 1000000.

De acordo com o desempenho o QuickSort escolhido para comparação foi o Algoritmo QuickSort Recursivo com Insercao com m=10.

Os algoritmos InsertionSort e BubbleSort apresentam alguns dados iguais a 0 a partir de N=100000, isso ocorreu devido ao tamanho do inteiro retornado durante os testes, eles apresentavam valores maiores do que o aceitável pelo tipo de variável int.

