

# Comparativos para Ordenação

Felipe Rivetti Mizher<sup>1</sup>

<sup>1</sup>Instituto de Ciências Exatas e Informática –  
Pontifícia Universidade Católica de Minas Gerais (PUCMG)  
– Belo Horizonte – MG – Brazil

{Felipe.mizher@sga.pucminas.br}

**Abstract.** *This report presents a comparative analysis of different sorting algorithms: Selection Sort, Insertion Sort, Bubble Sort, and Quicksort. The goal is to highlight the main differences between them in terms of the number of comparisons, number of data movements, and execution time, considering various input sizes. Through practical tests, it is possible to observe the performance of each algorithm and identify which ones are more efficient in different scenarios, emphasizing their advantages and limitations.*

**Resumo.** *Este relatório apresenta uma análise comparativa entre diferentes algoritmos de ordenação: Ordenação por Seleção (Selection Sort), Ordenação por Inserção (Insertion Sort), Ordenação por Bolha (Bubble Sort) e Quicksort. O objetivo é evidenciar as principais diferenças entre eles em relação ao número de comparações, número de movimentações e tempo de execução, considerando variados tamanhos de entrada de dados. Através de testes práticos, é possível observar o desempenho de cada algoritmo e identificar quais são mais eficientes para diferentes cenários, destacando suas vantagens e limitações.*

## 1. Selection Sort:

### Teste com 100 números

O algoritmo levou um total de **0.007 segundos**, realizou **4950 comparações** e **285 movimentações**.

### Teste com 1000 números

O algoritmo levou um total de **0.046 segundos**, realizou **499500 comparações** e **2967 movimentações**.

### Teste com 10000 números

O algoritmo levou um total de **0.298 segundos**, realizou **49995000 comparações** e **29952 movimentações**.

### Teste com 100000 números

O algoritmo levou um total de **36.008 segundos**, realizou **704982704 comparações** e **299661 movimentações**.

Quantidade de dados	Tempo (s)	Comparações	Movimentações
100	0.007	4.950	285
1.000	0.046	499.500	2.967
10.000	0.298	49.995.000	29.952
100.000	36.008	704.982.704	299.661

Figure 1. Uma tabela comparando o tempo, o número de comparações e o número de movimentações do algoritmo de seleção

## 2. Insertion Sort:

### Teste com 100 números

O algoritmo levou um total de **0.007 segundos**, realizou **532 comparações** e **2620 movimentações**.

### Teste com 1000 números

O algoritmo levou um total de **0.046 segundos**, realizou **8569 comparações** e **248719 movimentações**.

### Teste com 10000 números

O algoritmo levou um total de **0.363 segundos**, realizou **119027 comparações** e **25001884 movimentações**.

### Teste com 100000 números

O algoritmo levou um total de **7.170 segundos**, realizou **1522755 comparações** e **1803375923 movimentações**.

Tamanho de Entrada	Tempo (segundos)	Comparações	Movimentações
100	0.007	532	2.620
1.000	0.046	8.569	248.719
10.000	0.363	119.027	25.001.884
100.000	7.170	1.522.755	1.803.375.923

Figure 2. Uma tabela comparando o tempo, o número de comparações e o número de movimentações do algoritmo de inserção

## 3. Bubble Sort:

### Teste com 100 números

O algoritmo levou um total de **0.000 segundos**, realizou **4950 comparações** e **8208 movimentações**.

### Teste com 1000 números

O algoritmo levou um total de **0.012 segundos**, realizou **499500 comparações** e **738234 movimentações**.

### Teste com 10000 números

O algoritmo levou um total de **0.187 segundos**, realizou **49995000 comparações** e **75424602 movimentações**.

### Teste com 100000 números

O algoritmo levou um total de **26.693 segundos**, realizou **704982704 comparações** e **1122266702 movimentações**.

Tamanho de Entrada	Tempo (segundos)	Comparações	Movimentações
100	0.000	4.950	8.208
1.000	0.012	499.500	738.234
10.000	0.187	49.995.000	75.424.602
100.000	26.693	704.982.704	1.122.266.702

Figure 3. Uma tabela comparando o tempo, o número de comparações e o número de movimentações do algoritmo de Bolha

## 4. Quick Sort:

### Teste com 100 números

O algoritmo levou um total de **0.000 segundos**, realizou **773 comparações** e **561 movimentações**.

### Teste com 1000 números

O algoritmo levou um total de **0.000 segundos**, realizou **13258 comparações** e **7677 movimentações**.

### Teste com 10000 números

O algoritmo levou um total de **0.000 segundos**, realizou **174926 comparações** e **103671 movimentações**.

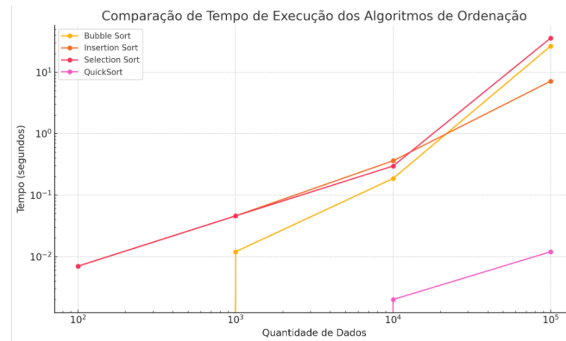
### Teste com 100000 números

O algoritmo levou um total de **0.036 segundos**, realizou **2016102 comparações** e **1398513 movimentações**.

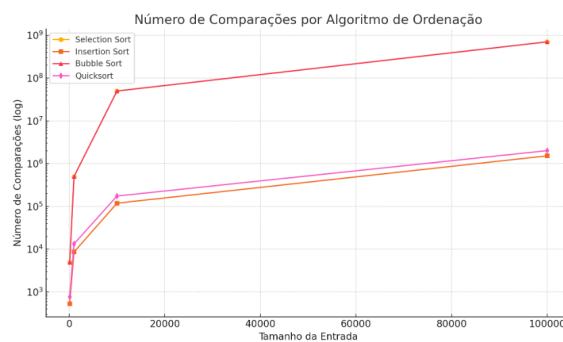
Tamanho de Entrada	Tempo (segundos)	Comparações	Movimentações
100	0.000	773	561
1.000	0.000	13.258	7.677
10.000	0.000	174.926	103.671
100.000	0.036	2.016.102	1.398.513

Figure 4. Uma tabela comparando o tempo, o número de comparações e o número de movimentações do algoritmo do quicksort

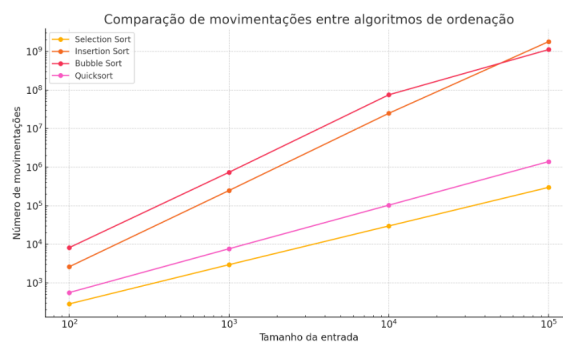
## 5. Gráficos:



**Figure 5.** O gráfico apresenta a comparação do tempo de execução entre os quatro algoritmos de ordenação em diferentes cenários de entrada, variando a quantidade de dados.



**Figure 6.** O gráfico apresenta o número de comparações feitas durante a execução dos quatro algoritmos de ordenação em diferentes cenários de entrada, variando a quantidade de dados.



**Figure 7.** O gráfico apresenta o número de movimentações feitas durante a execução dos quatro algoritmos de ordenação em diferentes cenários de entrada, variando a quantidade de dados.