

UNIVERSIDADE FEDERAL DA FRONTEIRA SUL

CAMPUS CHAPECÓ

Ciência da Computação

Disciplina: Pesquisa e ordenação de dados

Trabalho: Comparação de métodos de ordenação simples

Alisson Luan de Lima Peloso

Chapecó - SC

2020

# **Tabelas**:

## Bubblesort

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Ordenada** | | |
|  | **Bubble sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 0,032 | 9.999 | 0 |
| **50.000** | 0,141 | 49.999 | 0 |
| **100.000** | 0,332 | 99.999 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Inversamente Ordenada** | | |
|  | **Bubble sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 314,928 | 49.995.000 | 49.994.948 |
| **50.000** | 7.537,261 | 1.249.975.000 | 1.249.973.812 |
| **100.000** | 30.019,961 | 4.999.950.000 | 4.999.945.135 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aleatório** | | |
|  | **Bubble sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 335,516 | 49.979.069 | 25.138.420 |
| **50.000** | 9.852,922 | 1.249.943.875 | 623.460.127 |
| **100.000** | 39.567,065 | 4.999.758.110 | 2.505.344.671 |

## Selectionsort

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Ordenada** | | |
|  | **Selection sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 167,741 | 50.004.999 | 9.999 |
| **50.000** | 3.678,600 | 1.250.024.999 | 49.999 |
| **100.000** | 14.741,067 | 5.000.049.999 | 99.999 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Inversamente Ordenada** | | |
|  | **Selection sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 164,634 | 50.004.999 | 9.999 |
| **50.000** | 3.783,593 | 1.250.024.999 | 49.999 |
| **100.000** | 15.581,065 | 5.000.049.999 | 99.999 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aleatório** | | |
|  | **Selection sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 151,341 | 50.004.999 | 9.999 |
| **50.000** | 3.682,866 | 1.250.024.999 | 49.999 |
| **100.000** | 14.723,393 | 5.000.049.999 | 99.999 |

## Insertionsort

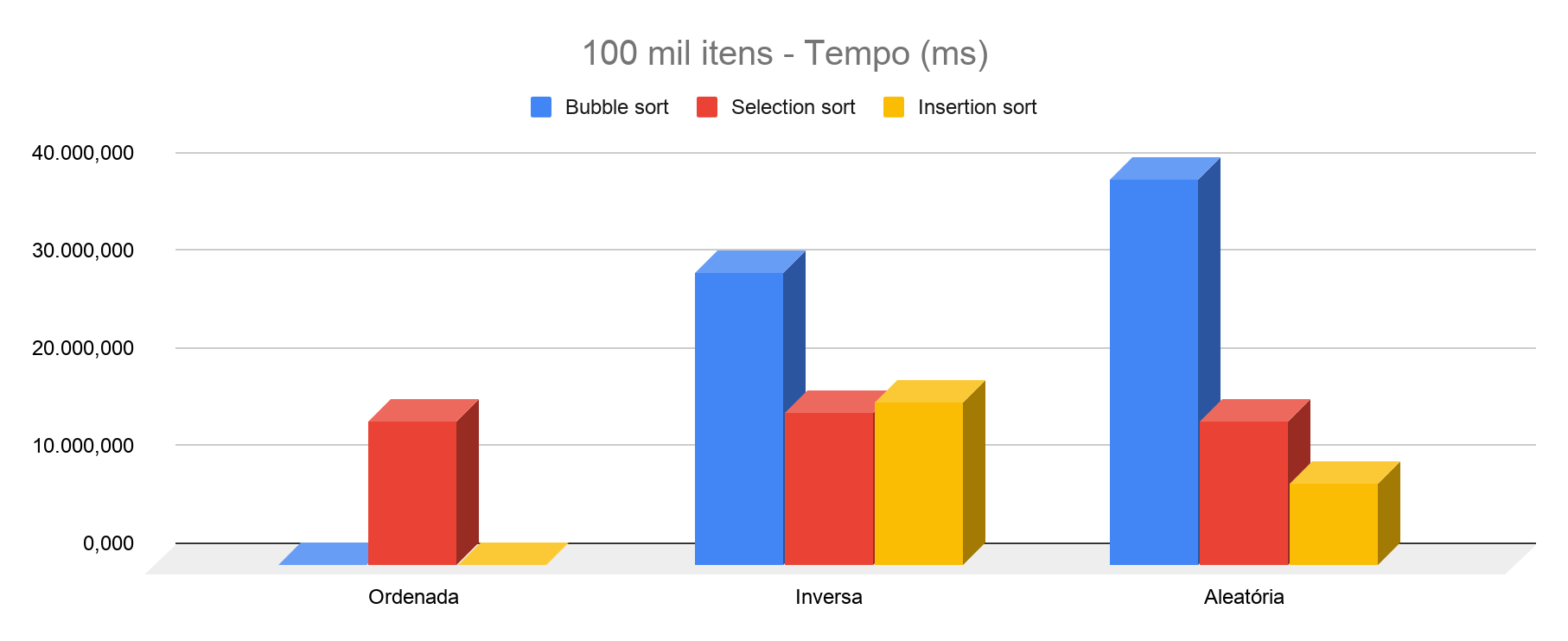
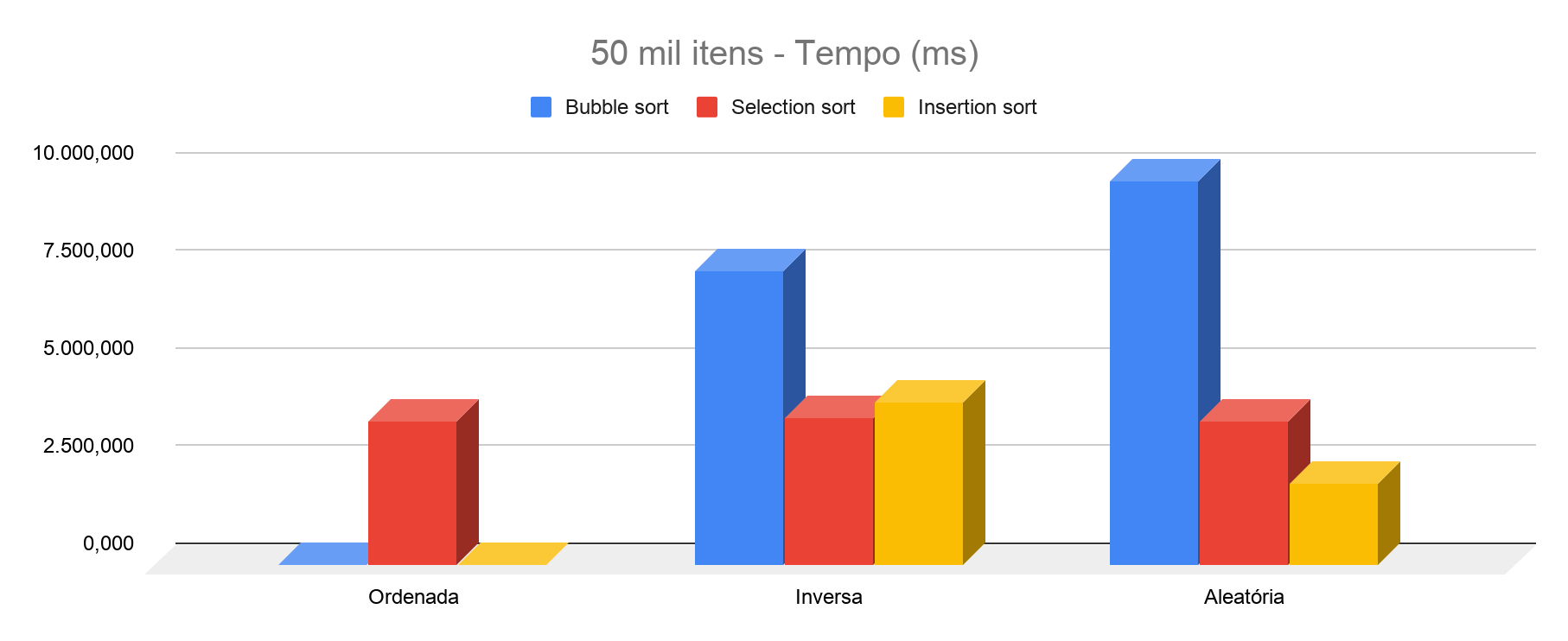
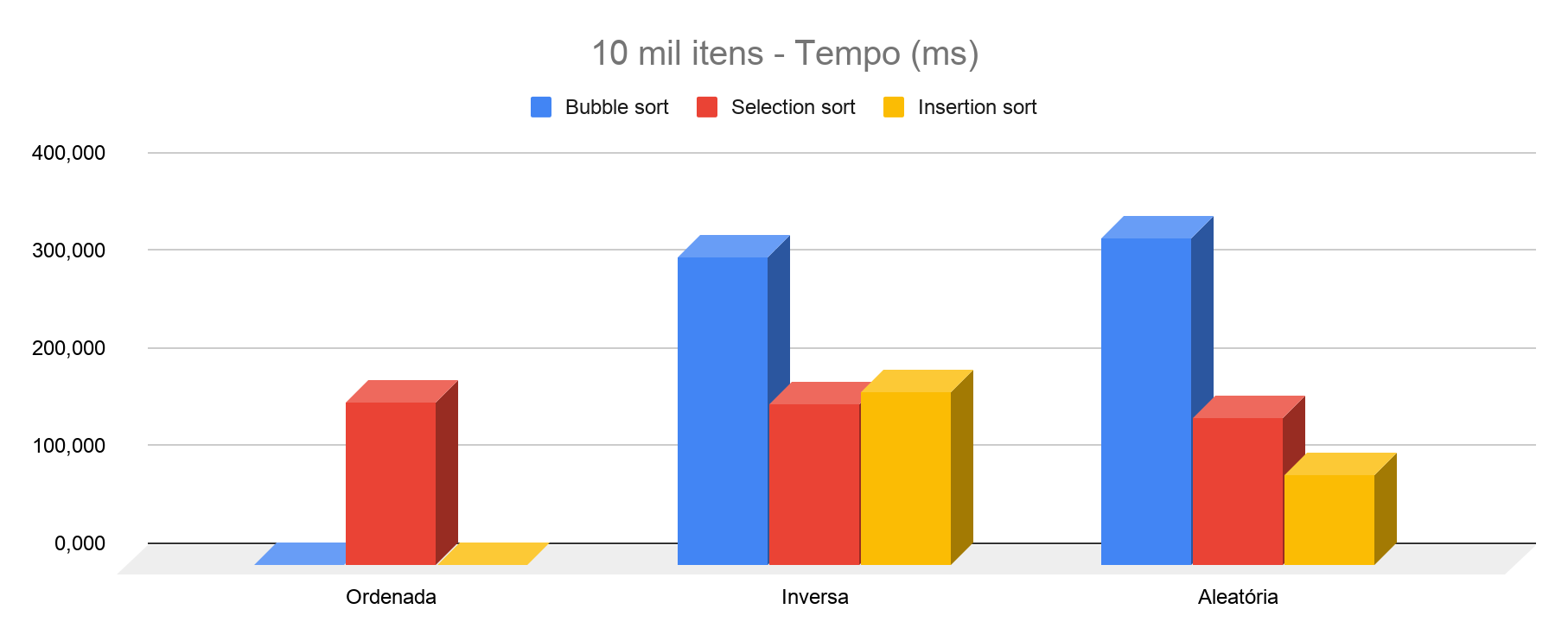
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Ordenada** | | |
|  | **Insertion sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 0,134 | 0 | 9.999 |
| **50.000** | 0,209 | 0 | 49.999 |
| **100.000** | 0,484 | 0 | 99.999 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Inversamente Ordenada** | | |
|  | **Insertion sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 177,586 | 49.994.947 | 9.999 |
| **50.000** | 4.196,920 | 1.249.973.743 | 49.999 |
| **100.000** | 16.676,151 | 4.999.944.897 | 99.999 |

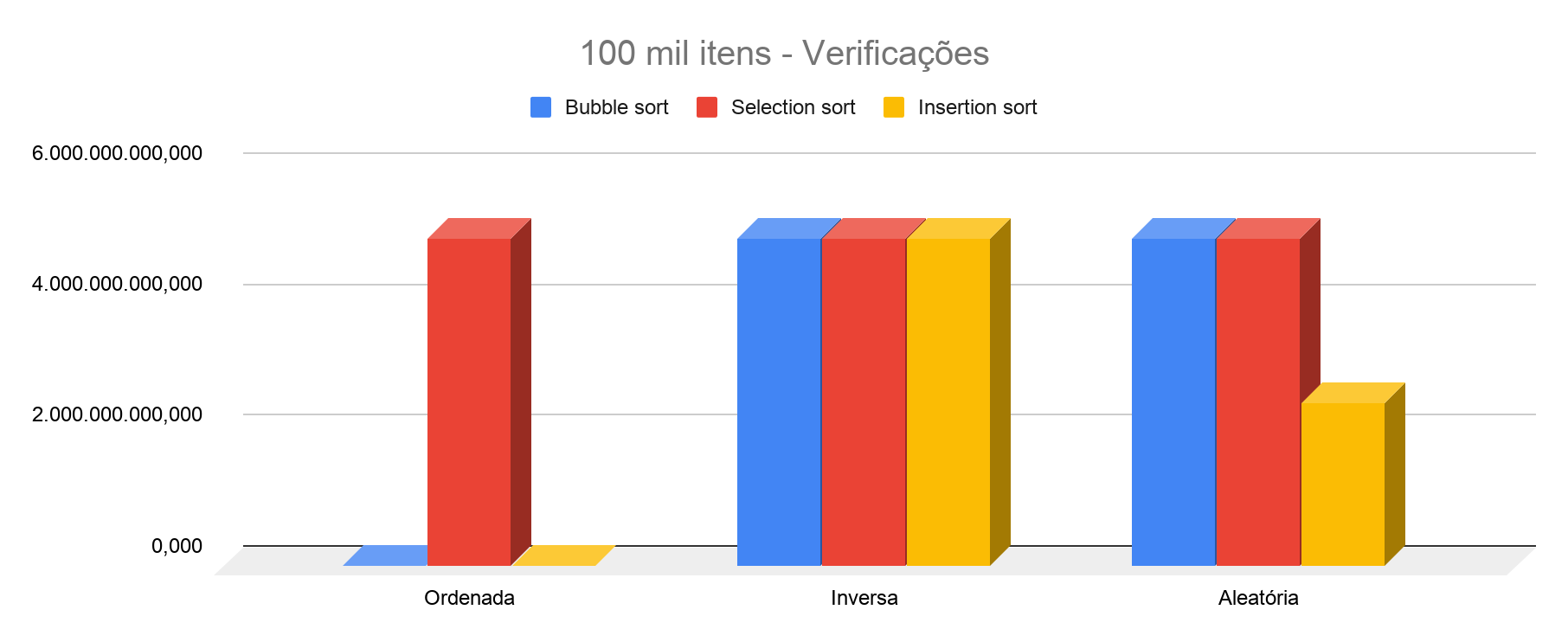
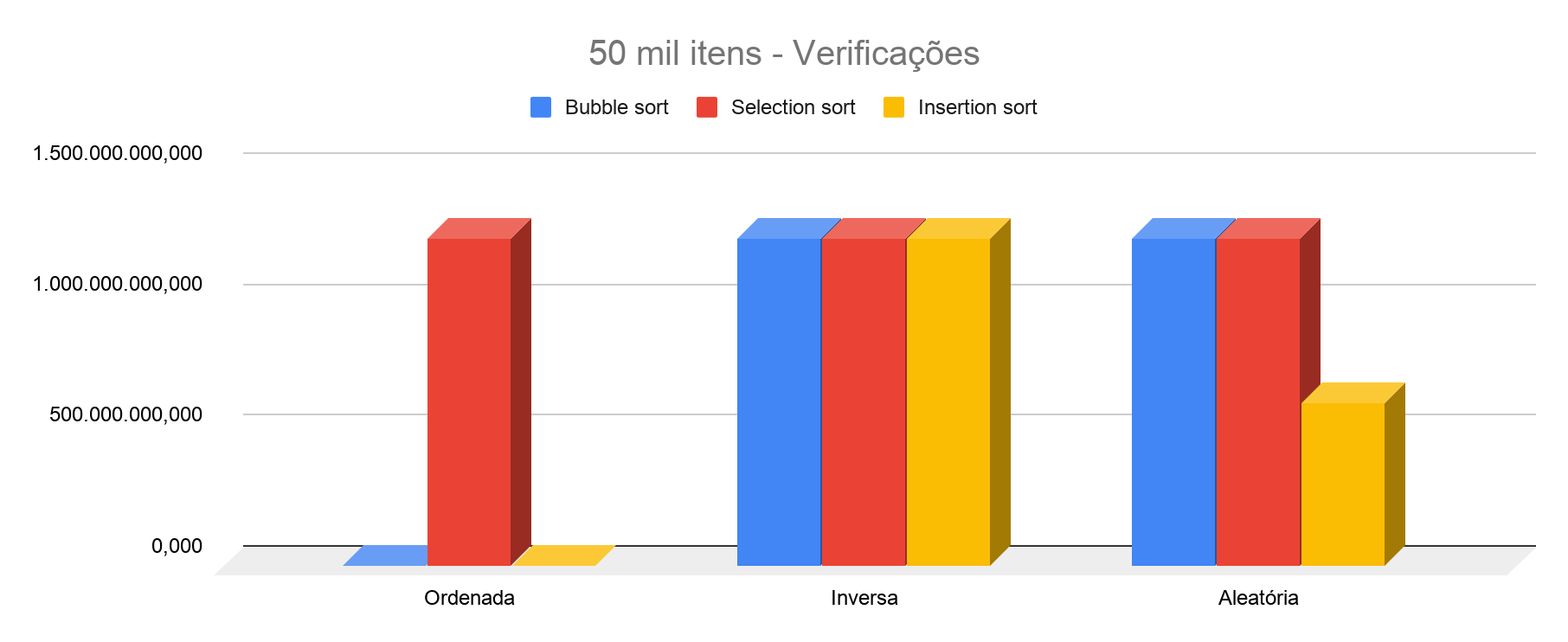
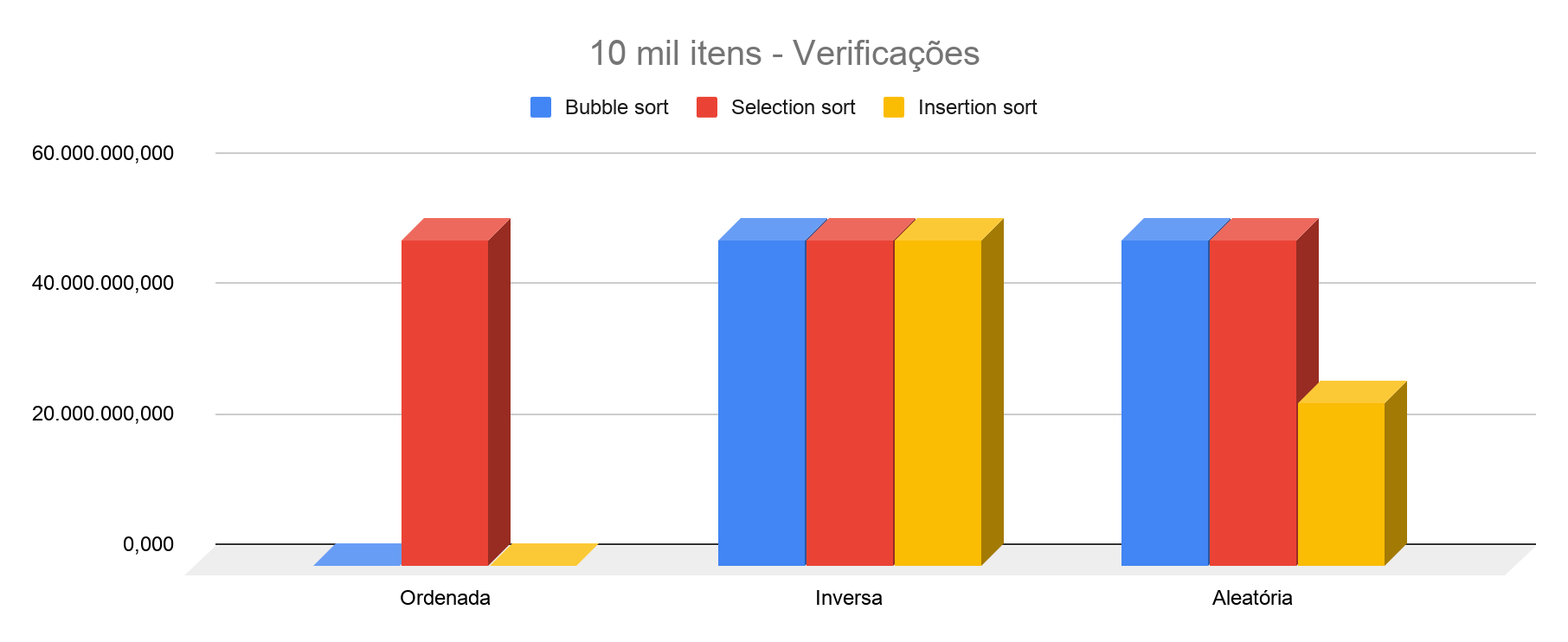
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Aleatório** | | |
|  | **Insertion sort** | | |
| **Tempo (ms)** | **Verificações** | **Interações** |
| **10.000** | 92,036 | 25.124.645 | 9.999 |
| **50.000** | 2.084,277 | 624.850.565 | 49.999 |
| **100.000** | 8.356,032 | 2.507.144.862 | 99.999 |

# Gráficos Comparativos:

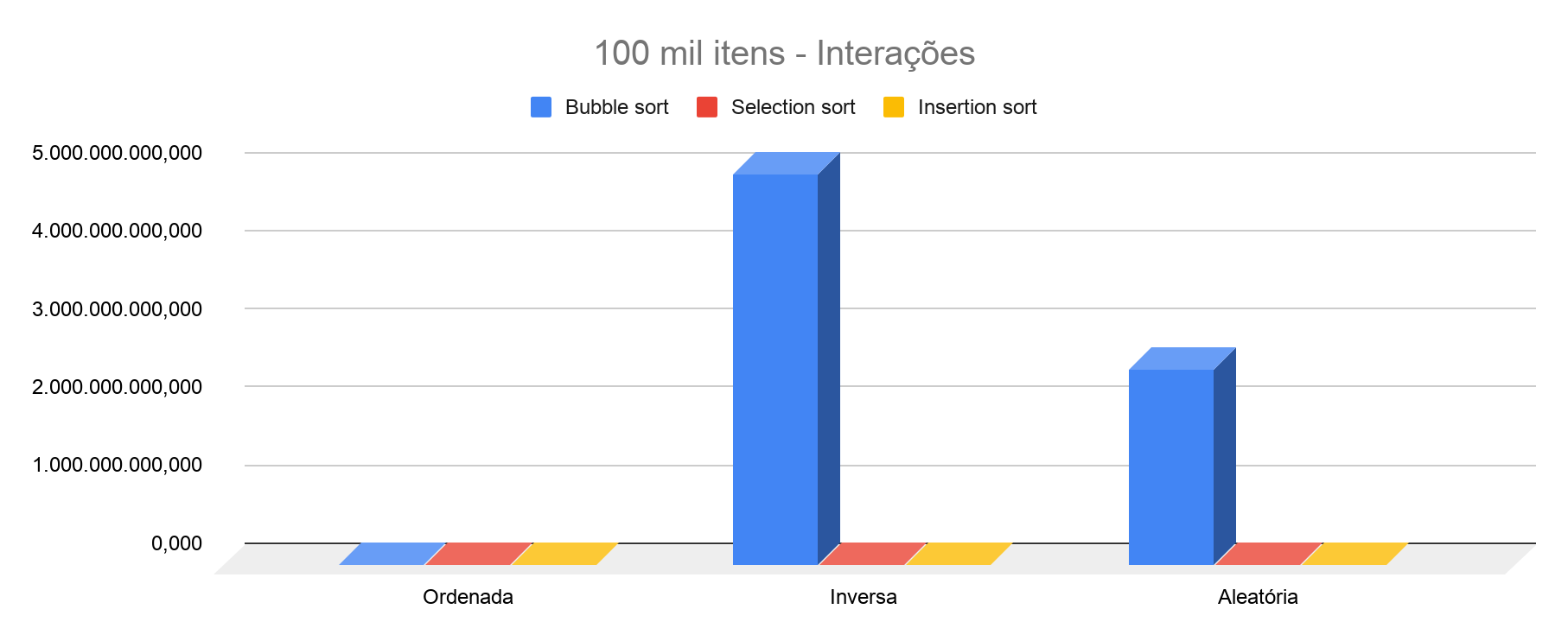
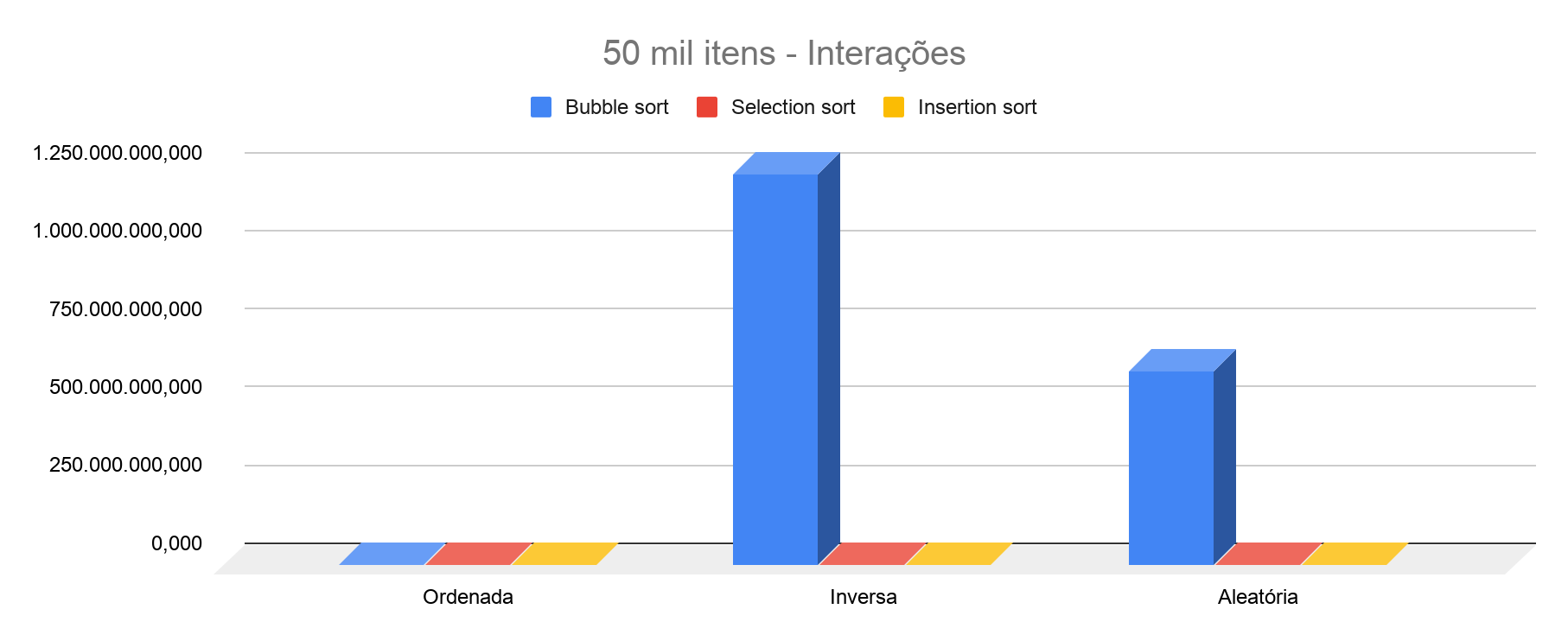
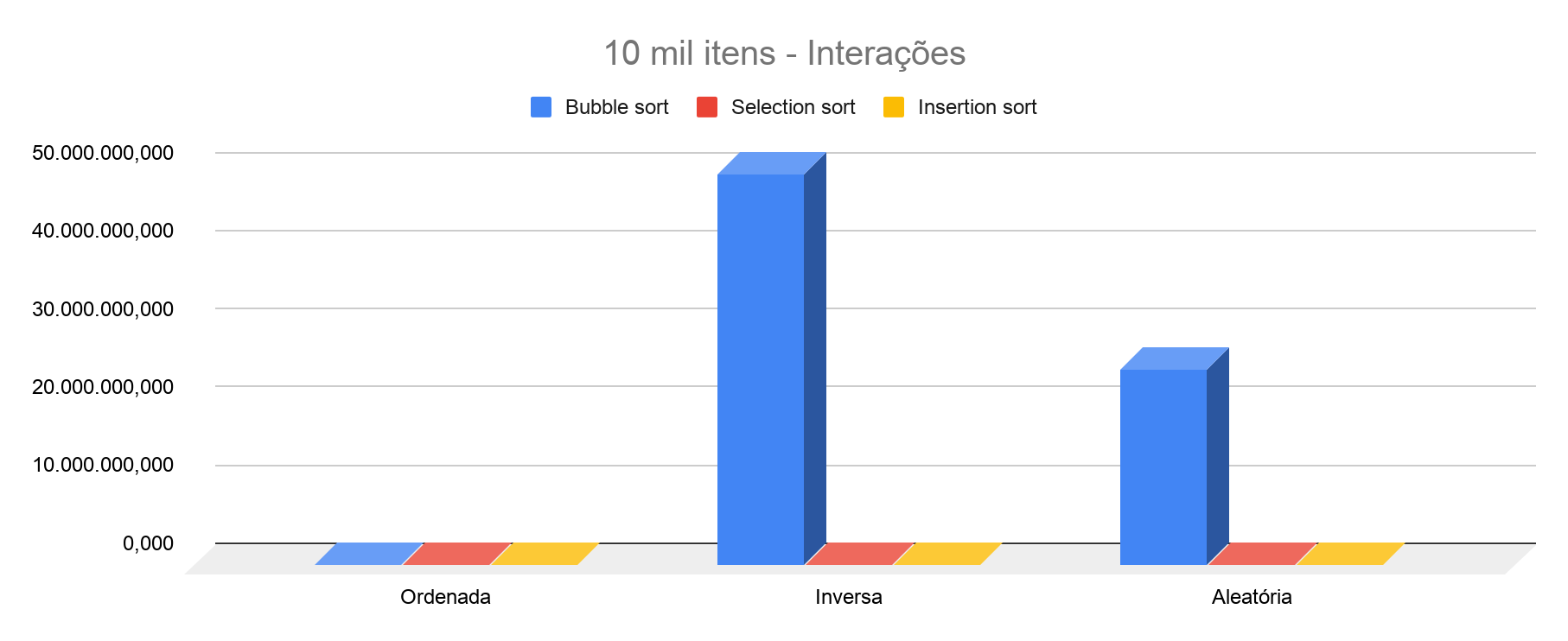
## Tempo de Execução



## Verificações Realizadas



## Interações Realizadas



# Conclusão:

Ao realizar a comparação com os gráficos é possível notar que os tempos de execução, número de interações e verificações se mantém na mesma proporção ao alterarmos a quantidade de itens a serem ordenados.

Outra coisa que podemos identificar é que o número de verificações permanecem iguais para o Selectionsort nas listas ordenadas, inversa e aleatória. No Bubblesort isso acontece apenas na lista inversa e na Aleatória. Já no Insertionsort o número de verificações se altera em todos os casos.

Analisando os gráficos de interação, percebe-se que o Selectionsort e Insertionsort não se alteram com a ordenação da lista. Isso pois o Selection realiza as verificações e faz apenas uma troca por ciclo. O Insertionsort realiza as comparações e vai deslocando para a direita até encontrar a posição correta do número, então realiza a troca. Já o Bubblesort, possui o número de interações dependentes da ordenação da lista, sempre realizando trocas entre os vizinhos e buscando levar o mais “grande” dos vizinhos para o lado esquerdo.

Por fim, entendemos que o desempenho dos algoritmos de ordenação dependem da forma inicial da lista, e para cada caso há um algoritmo que terá melhor desempenho. Como vimos, o Bubblesort possui melhor desempenho nas listas já ordenadas. Nas listas Inversas, o Selectionsort se sobressai. E em listas de ordenação aleatória, o Insertionsort se saiu melhor. É claro, levamos em consideração o tempo como métrica principal de desempenho e não a quantia de processos por trás de sua realização.

## Hardware Utilizado

**Processador**: Intel© Core™ i5-7200U CPU @ 2.50GHz × 2 with Turbo Boost up to 3.1Ghz

**Memória RAM**: 8 GB DDR4 Memory

# 