

# INTRODUÇÃO

Este documento foi desenvolvido pelo aluno Dalton D'Angelis para a disciplina de AEDS 1-P do curso Ciência da Computação da UNIFAL-MG. Este projeto é um relatório baseado em um código criado na linguagem C ++, onde o mesmo tem como objetivo realizar uma comparação entre três algoritmos de ordenação, que ordenam vetores dispostos de três formas: crescente, aleatória e decrescente. Este código contém contadores que visam contabilizar, toda vez que o vetor for utilizado, as operações de leitura e escrita. Dessa maneira, no final da compilação, os algoritmos de ordenação citados acima podem ser comparados com base na quantidade de operações realizadas em cada vetor para ordená-los. Com isso, tem-se uma base de qual dos três algoritmos utilizar em situações de ordenação específicas. A comparação foi feita entre os métodos implementados variando o tamanho do vetor em intervalos regulares, de 100 em 100 unidades até 1.000, 1.000 em 1.000 unidades até 10.000, e de 10.000 em 10.000 unidades até 100.000. Os valores obtidos foram utilizados para ilustrar graficamente os resultados.

## REFERENCIAL TEÓRICO

O referencial teórico deste projeto foram os algoritmos de ordenação disponibilizados na plataforma Google Classroom pelo professor Bressan, e as aulas práticas sobre criação de biblioteca local e implementação de funções realizadas pelo mesmo.

## MATERIAL UTILIZADO

Os materiais utilizados no trabalho foram basicamente todos softwares virtuais. Dentre os recursos usados no projeto, estão:

NetBeans -> Para construção, edição, teste e compilação do código com os algoritmos de ordenação.

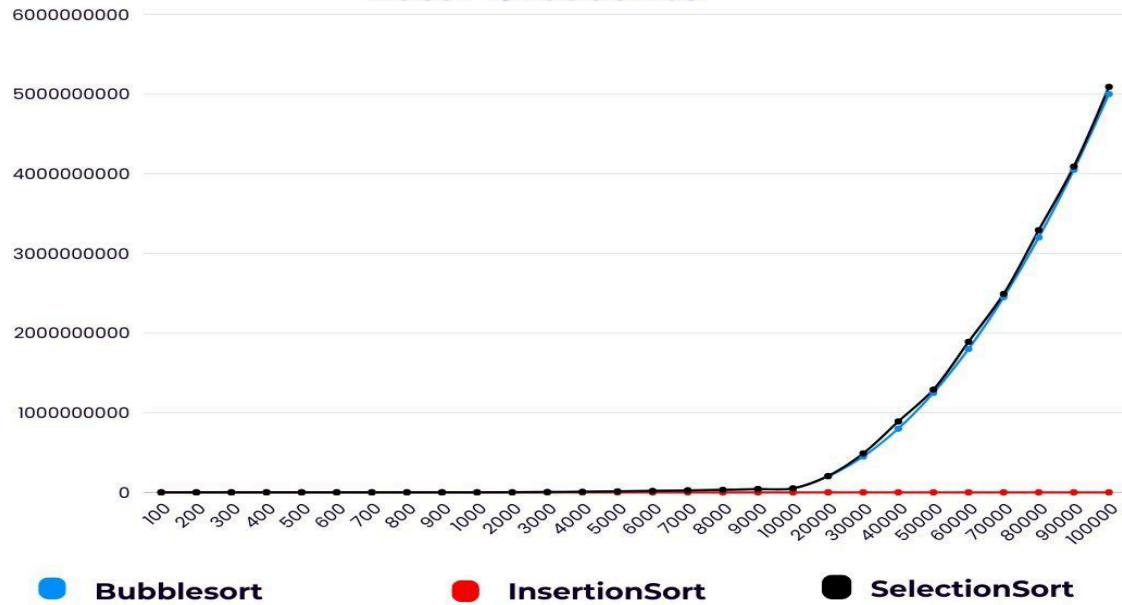
Canva -> Para a construção e implementação dos gráficos de cada algoritmo.

## MÉTODOS IMPLEMENTADOS

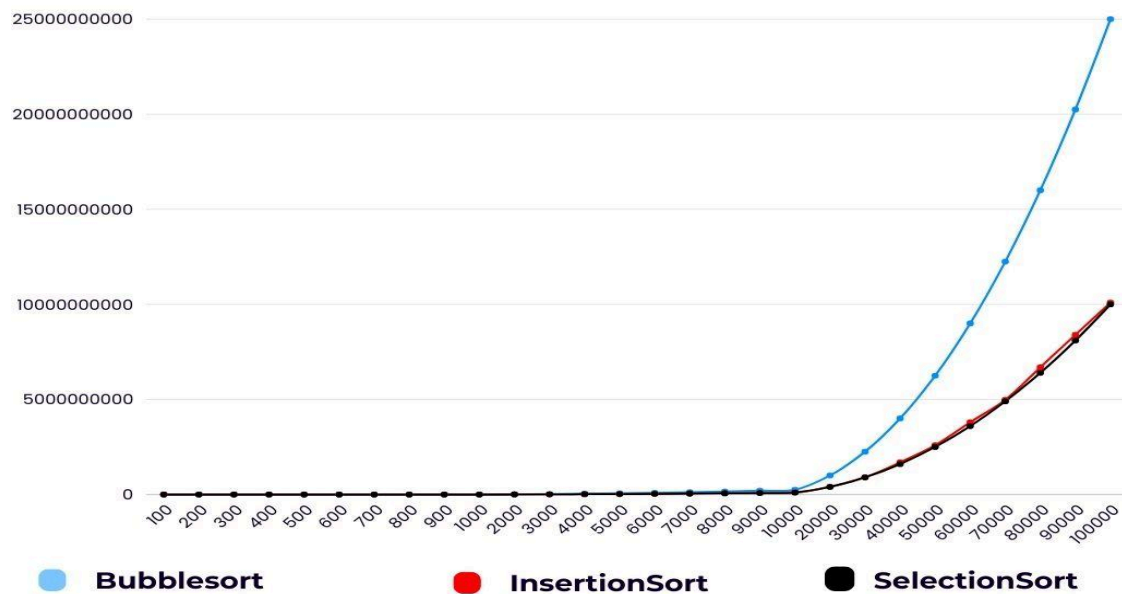
Os métodos implementados neste projeto foram os algoritmos de ordenação fornecidos pelo professor Bressan, as estratégias de criação de uma biblioteca local com algumas funções específicas, o uso de um menu de escolha que se mantém em loop até o usuário não desejar mais utilizá-lo, entre outras estratégias que melhoram a usabilidade geral do código.

# Resultados Obtidos

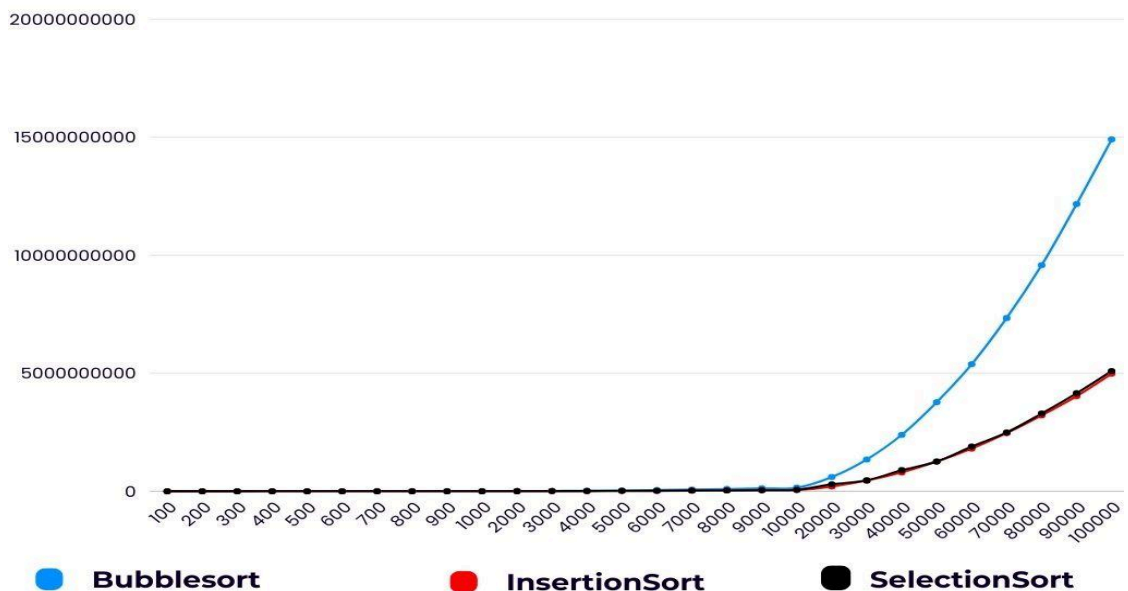
## Vetor Crescente



## Vetor Decrescente



## Vetor Aleatório



# Conclusão

Diante os resultados obtidos com a execução do código utilizando diferentes tamanhos de vetores. Tem-se as seguintes conclusões.

Nos vetores crescentes, o algoritmo de ordenação com maior destaque foi o “InsertionSort” que realizou pouquíssimas operações de leitura para concluir que o vetor já se encontrava ordenado, no entanto, os algoritmos “Bubble Sort” e “Selection Sort” tiveram um alto número de operações para realizar a mesma função.

Nos vetores decrescentes os métodos “Insertion Sort” e “Selection Sort” se destacaram mantendo um número de operações realizadas muito parecido de acordo com as mudanças de tamanho do vetor, já o “BubbleSort” realizou uma quantidade de operações muito maior que os dois citados anteriormente.

Nos vetores aleatórios, novamente os algoritmos de maior destaque foram o “Insertion” e “Selection”, obtendo números de operações muito parecidos e com crescimento similar com relação ao aumento de tamanho do vetor principal, entretanto, o “BubbleSort” apresentou números extremamente maiores em comparação aos outros dois algoritmos.