UNIVERSIDADE PAULISTA

CIÊNCIA DA COMPUTAÇÃO

ANDRÉ CARVALHO CERQUEIRA – D27057-5

ERICK WILLIAM LIMA RODRIGUES – N13795-9

GUSTAVO GARCIA LEAL ALVES DE MOURA – N130EA-8

THIAGO OLIVEIRA SANTOS – D227FC-2

**DESENVOLVIMENTO DE SISTEMA PARA ANÁLISE DE PERFORMANCE DE ALGORITMOS DE ORDENAÇÃO DE DADOS**

SÃO PAULO – SP

2018

ANDRÉ CARVALHO CERQUEIRA – D27057-5

ERICK WILLIAM LIMA RODRIGUES – N13795-9

GUSTAVO GARCIA LEAL ALVES DE MOURA – N130EA-8

THIAGO OLIVEIRA SANTOS – D227FC-2

**DESENVOLVIMENTO DE SISTEMA PARA ANÁLISE DE PERFORMANCE DE ALGORITMOS DE ORDENAÇÃO DE DADOS**

Orientador: Alan.

SÃO PAULO – SP

2018

**Sumário**

[1. Objetivo do trabalho 4](#_Toc529734522)

[2. Introdução 5](#_Toc529734523)

[3. Algoritmo de ordenação 6](#_Toc529734524)

[3.1 Bubble sort 6](#_Toc529734525)

[3.2 Select sort 6](#_Toc529734526)

[3.3 Insertion sort 6](#_Toc529734527)

[3.4 Quick sort 6](#_Toc529734528)

[4. Desenvolvimento 7](#_Toc529734529)

[5. Resultados e Discussão 9](#_Toc529734530)

[6. Considerações Finais 10](#_Toc529734531)

[7. Bibliografia 11](#_Toc529734532)

[8. Código Fonte 12](#_Toc529734533)

# Objetivo do trabalho

O objetivo deste trabalho é trazer um programa totalmente feito com a linguagem C, onde este programa terá como objetivo fazer ordenações de vetores com o método de sorteamento escolhido pelo usuário. Ao fim ele mostrará o tempo de demora de cada sorteamento, dando assim a base para a análise de estruturas.

Junto há a analise teórica de cada método de sorteamento, falando de como funciona a analise lógica deles e como funciona matematicamente. No fim o usuário terá a conclusão de por que certos métodos são mais demorados e outros mais rápidos, tudo baseado na lógica de como cada um tem.

# Introdução

# Algoritmo de ordenação

Algoritmo de ordenação na ciência da computação tem o significado de um algoritmo (programa) que organiza os elementos em uma ordem que foi colocada no algoritmo pelo programador, as mais utilizadas são: ordens crescentes e decrescentes.

Existem diversos algoritmos de ordenação, umas mais rápidas e outras mais lentas, ambas têm vantagens e desvantagens...

## Bubble sort

Bubble sort é um algoritmo de ordenação simples, sua principal maneira de ordenar é pegando o primeiro valor e testar com o segundo para ver qual é o maior, e fazendo teste lógico em cada posição do vetor, colocando o maior número na última posição. Depois deste teste, ele vai fazer a próxima interação, onde ele vai procurar o segundo maior número e colocar na penúltima ou na segunda posição e vai fazendo mais interações até ordenar todos os valores do vetor.

A quantidade de interações é de acordo com o tamanho do vetor -1, se um vetor contém 5 posições, ele vai fazer 4 interações.

## Select sort

## Insertion sort

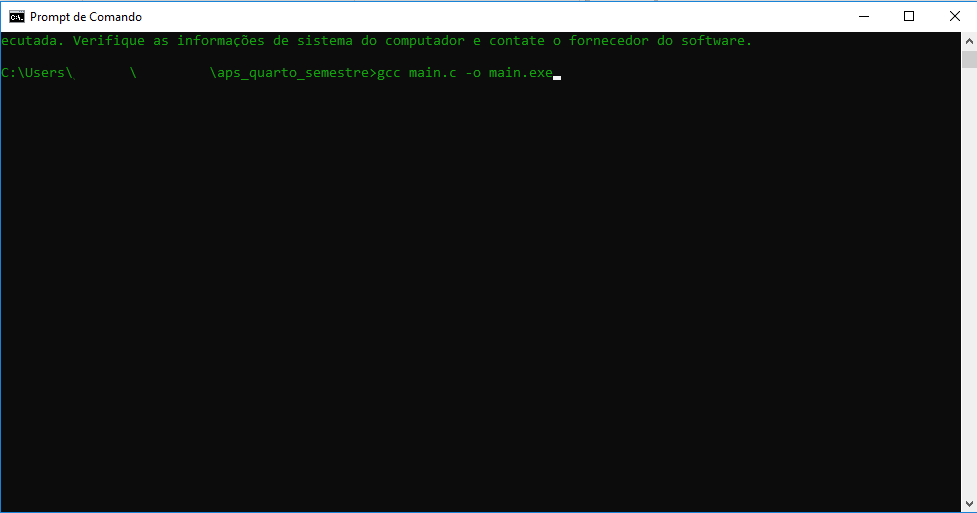
## Quick sort

# Desenvolvimento

O desenvolvimento do programa foi primeiramente baseado em desenvolver os métodos de ordenação que iriam ser aplicados. Em princípio seriam apenas três métodos, porém, com o avanço do curso, foi-se aplicado a utilização do Quicksort, pelo fato de apresentar método de recursividade. Além de aplicar essa funcionalidade extra, também a ideia do projeto é desenvolver um programa em C que possa ser executado em todos os sistemas operacionais (distribuições Linux, Windows e MacOS), assim trazendo bibliotecas que pudessem funcionar 100% em cada um.

Algumas partes do programa foram tanto compiladas e executadas pela IDE DevC++ como também foi feito testes no editor de texto Visual Studio Code, isso ficou baseado na preferência de cada um do grupo. Porém todos os executáveis foram criados através do famoso compilador GCC (GNU Compiler Collection), aplicado através de uma compilação via terminal.

Imagem x.x – Terminal CMD



Fonte: Autor do projeto – 2018

Também, para melhor rapidez de inserção de informações tanto no código do programa como também na parte escrita, este trabalho contou com a utilização do Git e Github para que o repositório sempre tivesse atualizado em qualquer alteração, assim todos do projeto ficariam cientes de quem fez o que nele. Também será mantido o projeto neste repositório para que qualquer pessoa possa ter acesso ao código e a parte escrita dele sempre sujeito aos direitos autorais.

Com o projeto organizado, com cada pasta e arquivos desenvolvidos, foi criado o arquivo main.c, onde foi aplicada todo o código. Na montagem do programa houve sempre há ideia de rebuscar ao máximo o que a linguagem C tem a proporcionar. Então para cada método de sorteamento seria feito através de uma struct (estrutura), com quatro vetores que iriam passar por cada método de sorteamento. Este struct foi alterado seu tipo primitivo apenas para ser chamado de Vetores.

Após a criação da struct Vetores, foram criados métodos auxiliares, como um para inserir valores aleatórios em um vetor, e espelhar com os outros vetores para ter maior precisão dos resultados de desempenho entre as funções de ordenação. A função “MostrarTela” para mostrar que o vetor foi corretamente preenchido. A função “Interface” para deixar a parte front-end mais user-friendly, e por final, a função “OpcaoSort” que é basicamente o core do programa. Ele irá servir tanto para pegar os resultados de desempenho entre as ordenações, assim como a execução de cada função que foi criada no projeto. Se fosse Orientação a objetos, poderíamos chama-la de classe pai basicamente, já que é ela que está gerenciando os ponteiros e as funções do programa. Por final, temos as funções de ordenação e a função de busca binária, no qual é o foco principal do trabalho. Durante o desenvolvimento foi adicionado um código um pouco “diferente” como experimento. O nosso convidado ilustre é o código espaguete, utilizado na linha 210 e 285 da fonte do código. Ele foi utilizado para ter um aumento de desempenho. Pois graças a ele, não houve necessidade de criar um novo método, ou repetir a mesma string várias vezes para apenas dizer que a opção já tinha sido escolhida.

# Resultados e Discussão

# Considerações Finais

# Bibliografia

**Não há fontes bibliográficas no documento atual.**

# Código Fonte