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](#_Toc529721938)

[2. Introdução 5](#_Toc529721939)

[3. Referencial teórico 6](#_Toc529721940)

[3.1 Bubble sort 6](#_Toc529721941)

[3.2 Select sort 6](#_Toc529721942)

[3.3 Insert sort 6](#_Toc529721943)

[3.4 Quick sort 6](#_Toc529721944)

[4. Desenvolvimento 7](#_Toc529721945)

[5. Resultados e Discussão 8](#_Toc529721946)

[6. Considerações Finais 9](#_Toc529721947)

[7. Bibliografia 10](#_Toc529721948)

[8. Código Fonte 11](#_Toc529721949)

# 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

# Referencial teórico

## Bubble sort

## Select sort

## Insert sort

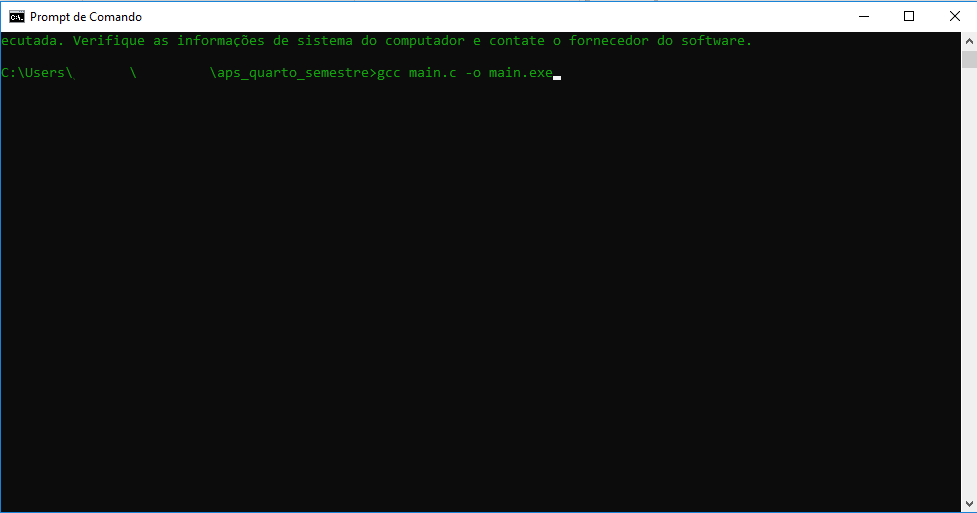
## Quick sort

# 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, foi criado métodos auxiliares para apenas para inserir valores aleatórios em cada vetor, sempre tendo um retorno da função o próp

# Resultados e Discussão

# Considerações Finais

# Bibliografia

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

# Código Fonte