Sumário

[1. Introdução: 2](#_Toc181025009)

[2. Implementação: 2](#_Toc181025010)

[3. Testes 2](#_Toc181025011)

[3.1 Teste n° 01 2](#_Toc181025012)

[3.2 Teste n° 02 2](#_Toc181025013)

[3.3 Teste n° 03 2](#_Toc181025014)

[3.4 Teste n° 04 2](#_Toc181025015)

[3.5 Teste n° 05 3](#_Toc181025016)

[4. Conclusão 3](#_Toc181025017)

[Referências 3](#_Toc181025018)

[Anexos 3](#_Toc181025019)

[calculadora.h 4](#_Toc181025020)

[calculadora.c 4](#_Toc181025021)

[main.c 4](#_Toc181025022)

## Introdução:

#### Este trabalho se trata de uma solução para um problema de gerenciamento estudantil, foi nos pedidos para que desenvolvêssemos um sistema de gerenciamento estudantil, seguindo algumas métricas especificas.

## Implementação:

O sistema foi implementado em linguagem Java como pedido pela matéria, seguindo métricas especificas, como gerar um relatório contendo professores e estudantes, um menu principal com 3 opções além da opção de sair, um menu aluno dentro da classe aluno contendo 2 opções, e dentro de uma das opções um outro menu de consulta, tendo o menu professor seguindo o mesmo padrão, um menu curso seguindo o mesmo modelo do menu aluno e professor, mas com um menu a mais contendo vinculação, e um menu relatório com 2 opções. Seguimos métricas especificas como *getters e setters*  em todas as classes e métodos de manipulação, que foram utilizados principalmente para facilitação de leitura e limpeza do menu principal, onde principalmente podemos manipular opções dentro da classe aluno, professor e curso ainda dentro do menu principal.

## Conclusão

Com este trabalho concluímos que somos capazes de construir um sistema gerencialmente estudantil eficiente em sua base, mesmo sendo um sistema “bruto”, é uma base ótima que poderia ser implementada efetivamente caso trabalhada, como o conectar a um banco de dados e uma interface mais intuitiva a um usuário final, que muitas vezes não é alguém de TI que esteja acostumado com sistemas mais difíceis.