**Calculadora de expressões aritméticas**

Estruturas de Dados I

*Prof. Francisco da Fonseca Rodrigues*

**RA:** 18173 **|** Ariane Paula Barros

**RA:** 18343 **|** Lorenna Joanna Leal Nunes

Campinas, 2019

**Índice de conteúdo**

Introdução...................................................................................................... 03

Desenvolvimento............................................................................................ 03

Conclusão...................................................................................................... 06

**Introdução**

Este projeto tem como objetivo a criação de uma calculadora que resolva expressões aritméticas, cumprindo com as especificações contadas no enunciado do projeto. Ele deve através dos conceitos de notação pósfixa e infixa implementar métodos que por exemplo: Gere sequencias (pósfixa e infixas) com os operandos e letras no lugar dos números; Calcule o resultado de uma sequencia pósfixa. O programa foi desenvolvido na linguagem de programação C#, com a utilização de pilha, uma estrutura de dados apresentada na disciplina estrutura de Dados I do 3º semestre do curso técnico de Campinas – COTUCA.

**Desenvolvimento**

14/04 – Criação do Projeto no GitHub.

15/04 – Criação do formulário da calculadora

16/04 – Criação da classe Expressao e desenvolvimento de seus métodos obrigatórios

17/04 – Exibição dos caracteres no txtVisor, configurar mensagens de erro e elaborar raciocínio da classe Expressao e sua função

20/04 – Começo do desenvolvimento do método ParaInfixa() e SeEhSinal()

22/04 – Desenvolvimento do método SeTemPreferencia(), de acordo com as informações da apostila

23/04 - Desenvolvimento do método VerificarParenteses()

24/04 – Término do desenvolvimento do método ParaInfixa()

25/04 – Começo do desenvolvimento do método ParaPosfixa()

26/04 – Desenvolvimento do método ParaPosfixa()

29/04 – Raciocínio da chamada de métodos pelo programa, determinamos que o método ParaInfixa() deve ser chamado primeiro, para preencher o vetor infixo, que será usado pelo método ParaPosfixa() e Resolver()

30/04 – Término do desenvolvimento do método ParaPosfixa()

02/05 – Começo do desenvolvimento do método Resolver()

03/05 – Desenvolvimento do método Resolver() e SubExpressao()

05/05 – Desenvolvimento do método Ordenar()

06/05 – Aprimoramento dos métodos ParaInfixa(), ParaPosfixa() e Resolver()

07/05 – Mudanças no código em relação ao uso de parênteses

08/05 – Término do relatório e comentários no código

09/05 – Arrumar a leitura de parênteses e não deixar o usuário digitar mais de um sinal seguido.

**Conclusão**

O projeto foi essencial para melhorarmos nosso raciocínio em relação ao funcionamento de uma sequência posfixa e infixa e ao uso de pilhas, desde como funcionam basicamente até como usá-la para transformar uma sequência matemática antes infixa, na forma pósfixa e calcular seu resultado. Aprendemos também e exercitamos nosso conhecimento e raciocínio sobre o uso de pilhas. Conseguimos transformar as sequencias infixas para pósfixa e calcular seu resultado.