



# CIÊNCIAS DA COMPUTAÇÃO

FACULDADE DE CIÊNCIAS | UNIVERSIDADE AGOSTINHO NETO

TEMA:

## Análise de expressões em C

DOCENTE

---

Amândio de Jesus Almada

Luanda Junho de 2025



# CIÊNCIAS DA COMPUTAÇÃO

FACULDADE DE CIÊNCIAS | UNIVERSIDADE AGOSTINHO NETO

TEMA:

## Análise de expressões em C

INTEGRANTES DO GRUPO 16:

- ✓ Eurico Kiala António : N° :932267987 [euricokialaantonio0@gmail.com](mailto:euricokialaantonio0@gmail.com)
- ✓ Maria Bibiana Kianguebeni António: N° :956821101  
[bibiana.kianguebeni@gmail.com](mailto:bibiana.kianguebeni@gmail.com)
- ✓ Osvaldo Salvador Manuel: N°: 930915016
- ✓ Sara Dimene: N°: 950995368 \_ [dimenesara@gmail.com](mailto:dimenesara@gmail.com)

## 1. Introdução

O presente relatório tem como objetivo documentar o desenvolvimento de um programa em linguagem C capaz de processar um arquivo de texto contendo expressões matemáticas, analisá-las sintática e semanticamente, avaliá-las e substituir seu conteúdo pelo resultado numérico correspondente, este procedimento é chamado de **análise de expressões**.

## 2. Objetivos do projeto

- **Objetivo principal**

O principal objetivo deste projeto é desenvolver um programa capaz de processar um arquivo de texto << in.txt >>, contendo expressões matemáticas, analisá-las e criar um novo arquivo de texto << out.txt >>, contendo o resultado numérico correspondente de cada expressão matemática ( linha/linha) contido no arquivo de texto << in.txt >>. Em caso de erro, o programa deverá identificar o tipo de erro e registrar uma mensagem adequada na linha correspondente.

- **Objetivos específicos**

- ✓ Facilitar o utilizador na resolução de problemas relacionados a análise de expressões matemáticas.

### 3. Principais Funcionalidades

O programa recebe como parametro, via linha de comando, o caminho para um arquivo in.txt contendo expressões matemáticas, uma por linha. Cada linha do arquivo é analisada individualmente. As expressões devem conter números inteiros de um único dígito, operadores aritméticos básicos (+, -, \*, /, ^) e parênteses para definir precedência.

#### Como funciona o programa?

- Cada expressão é avaliada (linha por linha) e o resultado da operação é gravado no arquivo de saída << out.txt >> na linha correspondente.
- Caso a expressão contenha erros, a linha correspondente a expressão no arquivo out.txt é substituída por uma mensagem descrevendo o tipo de erro.

### 4. Estrutura do projeto

#### 4.1. Organização dos ficheiros

Os ficheiros estão contidos na pasta G16, entre eles temos:

- index.c
- index.exe
- Out.txt

#### 4.2. Módulos

No presente projeto utilizou-se módulos para organizar o código-fonte em unidades menores e reutilizáveis.

Utilizou-se funções para modularizar o projeto.

### 5. Metodologia e desenvolvimento

O presente projeto foi desenvolvido por 4 estudantes, utilizando o IDE Dev-C++, utilizamos o bloco de texto do Dev-C++ para escrever um conjunto de passos finitos para resolver e estruturar o projeto.

Declarou-se funções tais como:

- Pilha\* inserir(Pilha \*p, char valor) {} – Esta função insere um dado carácter na pilha;
- float calcularExpressao(float n1, float n2, char sinal) {} – Esta função recebe dois números (n1 e n2) e o sinal aritmético (sinal), para verificar qual operação será realizada;
- float lerArquivo(char caminho[]) {} – Esta função recebe o caminho onde está guardado o arquivo << in.txt >> para posteriormente fazer a leitura das expressões;

- Etc.

Utilizamos a estrutura de dados **struct** para agrupar variáveis de diferentes tipos (como: int, char, float) sob um único nome e arrays para agrupar variáveis de um mesmo tipo sob uma única variável, etc.

Encontramos dificuldade no momento da criação da pilha e ao criar a função que calcula a expressão pós-fixada e, com estudos, união e pesquisa conseguimos superar as dificuldades.

## 6. Resultados e testes

Todas as funcionalidades foram implementadas com sucesso. O processo de empilhar e desempilhar funcionou bem, a manipulação de arquivos também funcionou bem, o processo de verificação de precedência de sinais também funcionou bem, etc.

Podemos melhorar mais ainda o programa criando um programa que analisa não só expressões matemáticas básicas, mais também as mais complexas, tais como, expressões algébricas e transcendentais.

## **7. Conclusão**

O trabalho foi concluído com sucesso e atende aos requisitos que nos foi proposto. Com realização do presente trabalho foi uma mais valia para nós, pois, graças a este trabalho aprendemos a lidar com arquivos em C, a manipular ponteiros, a trabalhar com pilhas, etc.

Link para o repositório no github

<https://github.com/euricokialaantonio/grupo16>