

## Documentação: Projeto Calculadora Phyton

**Autor:** Tiago Riêgo Dia Ribeiro

**Curso:** Desenvolvimento Full Stack

**Instituição:** Estácio

**Disciplina:** Logica, Algoritmos e Programação de Computadores

### 1. Objetivo do Projeto

O objetivo deste projeto é desenvolver uma calculadora funcional utilizando a linguagem Python. O programa implementa as quatro operações matemáticas básicas (adição, subtração, multiplicação e divisão) e é estruturado em torno de funções modulares, estruturas de controle de fluxo e um laço de repetição principal para permitir múltiplas operações sequenciais.

### 2. Funcionalidades Implementadas

O programa atende a todos os requisitos solicitados, incluindo:

- **Operações Básicas:** Realiza adição, subtração, multiplicação e divisão.
- **Entrada Flexível:** O usuário pode escolher a operação digitando tanto o símbolo (+, -, \*, /) quanto o nome da operação (adicao, subtracao, etc.).
- **Tratamento de Erro (Divisão por Zero):** O programa valida especificamente se o divisor é zero e, caso seja, retorna uma mensagem de erro amigável ao usuário, evitando o "crash" da aplicação.
- **Loop de Execução Contínua:** O programa roda dentro de um laço while, permitindo que o usuário realize quantos cálculos desejar.
- **Controle de Saída:** Ao final de cada cálculo, o programa pergunta ao usuário se ele deseja continuar (S) ou sair (N), tratando a entrada em maiúsculas ou minúsculas.
- **Tratamento de Erro (Entrada de Dados):** Foi implementado um bloco try...except ValueError para garantir que a aplicação não quebre caso o usuário digite um valor não numérico (ex: "abc") nos campos de número.

### 3. Estrutura do Código (Funções)

O código foi modularizado em várias funções para garantir a clareza, o reuso e a fácil manutenção. A estrutura principal é a seguinte:

#### Funções de Operação

- adicao(a, b): Recebe dois números e retorna a soma.
- subracao(a, b): Recebe dois números e retorna a subtração.
- multiplicacao(a, b): Recebe dois números e retorna a multiplicação.
- divisao(a, b): Recebe dois números. Contém uma estrutura if que verifica se b é igual a 0. Em caso positivo, retorna uma string de erro; do contrário, retorna a divisão.

## **Funções de Controle**

- calculadora(a, b, operacao):
  - Utiliza uma estrutura if/elif/else para analisar o parâmetro operacao (após convertê-lo para minúsculas com .lower()).
  - Chama a função de operação matemática correspondente (ex: adicao()) e armazena o valor em uma variável resultado.
  - Retorna a variável resultado para o laço principal.

## **Bloco Principal (Laço while)**

- Inicia um laço while que continua executando enquanto a variável saida for diferente de 'n'.
- Dentro do laço, pede ao usuário o a\_input, b\_input e op\_input.
- Chama a função calculadora() com os valores recebidos.
- Imprime o resultado formatado.
- Pergunta ao usuário se deseja continuar e atualiza a variável saida, controlando assim o fluxo do laço.

## **4. Instruções de Execução**

1. Garanta que o interpretador Python (versão 3.x) esteja instalado no sistema.
2. Salve o código-fonte em um arquivo com a extensão .py (ex: calculadora.py).
3. Abra um terminal ou prompt de comando.
4. Navegue até o diretório onde o arquivo foi salvo.
5. Execute o script com o comando:

Bash

`python calculadora.py`

6. Siga as instruções apresentadas no console para inserir os números e a operação.
7. Ao ser perguntado se deseja continuar, digite S para realizar um novo cálculo ou N para encerrar o programa.

## **5. Conclusão**

O projeto cumpriu com sucesso todos os requisitos do enunciado, resultando em uma aplicação de console funcional. A separação da lógica em funções distintas (como a verificação da divisão por zero na função divisao) e o tratamento de erros de entrada do usuário (try...except) foram fundamentais para a estabilidade do programa.