

EDITAL DE INSCRIÇÃO (RESUMO EXPANDIDO) - ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

**A IMPORTÂNCIA DE CRIAR UMA CALCULADORA EM PYTHON PARA
ESTUDANTES: LÓGICA E INICIAÇÃO NA LINGUAGEM.**

Graziely De Assis Santos (grazy.assis.santos@gmail.com)

Milena Gomes Nunes (milenanunes_ims@a.metodista.br)

Matheus Barreto Soares (barretozenilda585@gmail.com)

Matheus Gomes Costa (matheusgomes2077@gmail.com)

Tiago Misawa Calixto (tiago.calixto@metodista.br)

A criação de uma calculadora em Python é uma das atividades mais comuns e fundamentais para estudantes que estão começando sua jornada na programação. Embora pareça uma tarefa simples à primeira vista, ela representa um marco importante no processo de aprendizagem, pois envolve uma série de conceitos essenciais da lógica computacional e do funcionamento básico da linguagem Python. Desenvolver uma calculadora não é apenas aprender a somar, subtrair, multiplicar e dividir, mas sim aplicar raciocínio lógico, estruturar o pensamento e compreender o comportamento de um programa de computador.

Python é amplamente reconhecida por sua sintaxe clara e objetiva, características que a tornam ideal para o ensino de programação (ZELLE, 2016). O aluno, ao construir uma calculadora, lida com variáveis, operadores

aritméticos, estruturas condicionais (if, else, elif), laços de repetição (while, for), funções e manipulação de entrada e saída. Esses elementos são considerados essenciais em praticamente todas as linguagens de programação (SEBESTA, 2018).

Segundo Aho e Ullman (1995), a clareza na estruturação de algoritmos é um dos pilares para o aprendizado eficaz da programação. A calculadora, nesse sentido, permite que o estudante exercite a decomposição de um problema maior — como realizar diferentes operações matemáticas — em partes menores, compreensíveis e reutilizáveis. Trata-se, portanto, de um exercício de pensamento computacional, que também envolve a antecipação de erros e a formulação de soluções automatizadas para problemas simples.

À medida que o estudante avança no projeto, pode incluir melhorias como tratamento de exceções, menus interativos, validação de entrada e até o uso de bibliotecas como math, ou interfaces gráficas com tkinter. De acordo com Sweigart (2020), aplicar a programação em tarefas práticas do dia a dia é uma forma eficiente de consolidar conhecimentos e de entender o propósito real da codificação. A evolução de uma calculadora simples para uma mais robusta estimula a curiosidade, a autonomia e a criatividade, competências importantes na formação de futuros desenvolvedores.

Além disso, a atividade de criação de uma calculadora introduz um elemento essencial ao aprendizado de programação: o processo de depuração de código. Lutz (2013) ressalta que aprender a programar envolve também a capacidade de testar, errar e corrigir. Ao cometer erros comuns, como esquecer um operador ou digitar uma variável incorretamente, o estudante é desafiado a investigar o problema e encontrar uma solução, desenvolvendo habilidades investigativas e paciência.

Outro aspecto relevante é a acessibilidade da linguagem Python. Como destaca Zelle (2016), Python permite que o iniciante concentre sua atenção nos conceitos lógicos, em vez de se perder na complexidade sintática. Isso acelera o aprendizado e proporciona uma experiência mais fluida, especialmente para

quem nunca teve contato com programação. A clareza da linguagem também favorece o trabalho colaborativo, já que o código se torna mais compreensível para outros estudantes e professores.

Sebesta (2018) aponta que a escolha da linguagem influencia diretamente na forma como os conceitos são assimilados. Quando se usa uma linguagem de alto nível e de fácil leitura como Python, a motivação do aluno tende a aumentar, pois ele percebe rapidamente os resultados de seu trabalho. Isso reforça a importância de projetos simples, como a calculadora, como primeiro passo no ensino de programação.

Em síntese, a construção de uma calculadora em Python representa muito mais do que uma simples aplicação matemática. Ela é uma ferramenta pedagógica poderosa, capaz de introduzir o estudante nos conceitos fundamentais da programação, estimular o raciocínio lógico, promover a resolução de problemas e consolidar habilidades digitais essenciais no mundo contemporâneo. Além disso, permite o desenvolvimento de competências como autonomia, criatividade, trabalho em equipe e pensamento crítico — qualidades indispensáveis para a atuação profissional em um cenário cada vez mais orientado pela tecnologia.

REFERÊNCIAS

AHO, Alfred V.; ULLMAN, Jeffrey D. Fundamentals of programming languages. Reading: Addison-Wesley, 1995.

LUTZ, Mark. Learning Python. 5. ed. Sebastopol: O'Reilly Media, 2013.

SEBESTA, Robert W. Conceitos de linguagens de programação. 10. ed. Porto Alegre: Bookman, 2018.

SWEIGART, Al. Automatize tarefas maçantes com Python: programe soluções para problemas do dia a dia. 2. ed. São Paulo: Novatec, 2020.

ZELLE, John M. Python programming: an introduction to computer science. 3. ed. Wilsonville: Franklin, Beedle & Associates, 2016.

Palavras-chave: python; lógica de programação; calculadora; ensino de programação; pensamento computacional.

