

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

**A EXPERIENCIA DE CRIAR UMA CALCULADORA EM PYTHON:
DESVENDANDO A PROGRAMAÇÃO DE ALTO NÍVEL NO SEGUNDO
SEMESTRE**

Guilherme Oliveira (guilhermeoliveiramanso@gmail.com)

Kaio Souza Aires (kaioimhacker@gmail.com)

Gabriel Henry Oliveira Santos (gabrielhenryolliv@gmail.com)

Heitor Wennyd (heitorwennyd56@gmail.com)

Gabriel Casonatto (casonattog0@gmail.com)

O segundo semestre do curso de Análise e Desenvolvimento de Sistemas desvela um universo de possibilidades para o jovem programador, onde a teoria se entrelaça com a prática de forma indelével. É nesse cenário que a linguagem de alto nível, com sua sintaxe elegante e poder expressivo, emerge como a ferramenta primordial para transformar ideias em código funcional. A jornada de um grupo de cinco estudantes – Guilherme Oliveira Manso, Kaio Souza Aires, Gabriel Casonatto, Gabriel Henry e Heitor Wennyd – ilustra de forma vívida essa transição, culminando na criação de uma calculadora em Python. Este projeto, aparentemente simples, tornou-se um microcosmo dos desafios e triunfos inerentes ao aprendizado da programação, servindo como um catalisador para a compreensão de conceitos fundamentais.

Segundo Zelle (2016), a programação em Python oferece uma abordagem acessível e didática para o ensino da ciência da computação, permitindo que os alunos se concentrem na resolução de problemas em vez de em detalhes técnicos complexos da linguagem. Nesse sentido, a escolha do Python mostrou-se acertada, permitindo que os estudantes se concentrassem na lógica sem se perderem em complexidades sintáticas.

O desenvolvimento da calculadora em Python foi um mergulho profundo nos fundamentos da programação. Inicialmente, a equipe se deparou com a necessidade de compreender a lógica por trás das operações matemáticas e como traduzi-las para o código. A primeira etapa envolveu a criação de funções básicas para adição, subtração, multiplicação e divisão. Contudo, o verdadeiro desafio surgiu ao integrar essas funções em uma interface de usuário interativa. A exploração de bibliotecas como Tkinter para a criação de uma GUI (Graphical User Interface) revelou a importância de conceitos como eventos, callbacks e layouts.

De acordo com Sweigart (2020), a automação e o uso de bibliotecas em Python facilitam a construção de soluções práticas e didáticas, ao mesmo tempo em que aproximam o estudante do desenvolvimento de software do mundo real. Assim, cada botão da calculadora, cada campo de entrada, representava uma oportunidade de aplicar e consolidar o conhecimento recém-adquirido.

A depuração de erros, uma etapa inevitável em qualquer projeto de software, transformou-se em um exercício de paciência e raciocínio lógico. Como destacam Deitel e Deitel (2016), os erros de programação não devem ser vistos apenas como obstáculos, mas como etapas naturais do processo de aprendizado, uma vez que “cada erro é uma oportunidade de compreender mais profundamente o funcionamento do programa” (DEITEL; DEITEL, 2016, p. XX).

A colaboração entre Guilherme Oliveira Manso, Kaio Souza Aires, Gabriel Casonatto, Gabriel Henry e Heitor Wennyd foi crucial. As discussões sobre a

melhor forma de implementar uma funcionalidade, a revisão de código entre pares e a divisão de tarefas não apenas aceleraram o processo, mas também enriqueceram a experiência de aprendizado. Como apontam Sedgewick e Wayne (2010), a computação deve ser ensinada como uma ciência colaborativa, em que a troca de ideias é essencial para a construção do conhecimento.

A experiência com a calculadora demonstrou que a programação não é um ato solitário, mas uma construção coletiva, onde cada contribuição, por menor que seja, é fundamental para o sucesso do projeto. A superação dos desafios técnicos, como a manipulação de strings para exibir expressões complexas ou a validação de entradas do usuário, reforçou a resiliência e a capacidade de adaptação do grupo.

Em retrospectiva, a jornada da calculadora em Python transcendeu a mera construção de um software; ela se tornou uma metáfora para o aprendizado da programação de alto nível. A experiência de Guilherme Oliveira Manso, Kaio Souza Aires, Gabriel Casonatto, Gabriel Henry e Heitor Wennynd no segundo semestre do curso de Análise e Desenvolvimento de Sistemas sublinha a importância de projetos práticos e colaborativos como pilares para a formação de programadores competentes.

A capacidade de transformar um problema abstrato em uma solução concreta, de depurar erros com persistência e de colaborar efetivamente em equipe são habilidades que se forjam nesse tipo de vivência. Nesse sentido, como reforça Sweigart (2020), o aprendizado prático por meio de projetos “não apenas fixa o conhecimento técnico, mas motiva o estudante ao mostrar resultados imediatos e úteis”.

A programação, longe de ser uma disciplina puramente técnica, revela-se uma arte que exige criatividade, lógica e, acima de tudo, a paixão por construir. A calculadora em Python, com sua simplicidade e funcionalidade, é um testemunho do poder do aprendizado ativo e da colaboração, pavimentando o caminho para desafios ainda maiores no horizonte da programação. Este

trabalho reafirma que o domínio de uma linguagem de alto nível não se dá apenas pela absorção de sintaxes, mas pela imersão em projetos que demandam raciocínio, persistência e a inestimável troca de saberes entre pares.

BIBLIOGRAFIA:

SEEDGEWICK, Robert; WAYNE, Kevin. Introdução à computação com Java: uma abordagem interdisciplinar. São Paulo: Pearson, 2010.

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

DEITEL, Paul; DEITEL, Harvey. Java: como programar. 10. ed. São Paulo: Pearson, 2016.

(obs.: existe também versão para C, C++ e Python com títulos semelhantes)

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

Palavras-chave: python; programação; interface gráfica; colaboração; aprendizado prático.