

ESTÁCIO DE RIBEIRÃO PRETO

Curso: Desenvolvimento Full Stack

Disciplina: Lógica, Algoritmos e Programação de Computadores

Professor: Alessandro

Aluno: Daniel Victor

ESTRUTURAS DE CONDIÇÃO EM PYTHON

Ribeirão Preto — Junho de 2025

1. Introdução

Este relatório tem como finalidade apresentar o desenvolvimento de microatividades propostas na disciplina de Lógica, Algoritmos e Programação de Computadores. O foco está na aplicação prática dos conceitos fundamentais de programação com a linguagem Python, especialmente no uso de estruturas de decisão, laços de repetição e funções.

O trabalho foi desenvolvido com o objetivo de fixar o conteúdo estudado em sala de aula, proporcionando uma abordagem interativa com o uso do terminal para entrada e saída de dados, favorecendo a compreensão do funcionamento do código.

2. Objetivos

Objetivo Geral:

Aplicar estruturas básicas de programação em Python através de microatividades que estimulem a prática da lógica computacional.

Objetivos Específicos:

- Implementar estruturas condicionais (if, elif, else) para tomadas de decisão.
- Utilizar laços de repetição (for, while) para automatizar execuções.
- Criar e utilizar funções, com e sem parâmetros, retornando valores.

3. Desenvolvimento

Durante a execução das microatividades, foram abordados diversos conceitos fundamentais da linguagem Python:

3.1 Estruturas Condicionais

As estruturas if, elif e else foram utilizadas para controlar o fluxo do programa com base em condições. Um exemplo foi o uso dessas estruturas para exibir mensagens diferentes conforme a temperatura ou a experiência do usuário.

3.2 Laços de Repetição

Os laços for e while foram empregados para repetir ações, como a leitura de entradas ou a manipulação de strings. Essas estruturas foram essenciais para percorrer elementos e realizar interações controladas com o usuário.

3.3 Funções

Foram implementadas funções simples e compostas, com e sem retorno de valores. O uso de funções possibilitou a reutilização de código e uma melhor organização dos scripts.

3.4 Mini Calculadora

Uma das microatividades mais completas consistiu na criação de uma calculadora com as quatro operações básicas (soma, subtração, multiplicação e divisão), incluindo tratamento de exceções para evitar divisão por zero, proporcionando uma experiência mais segura ao usuário.

3.5 Interação via Terminal

Todas as atividades foram feitas com entrada e saída de dados via terminal, promovendo um aprendizado prático e direto.

4. Como Executar o Projeto

Para rodar os códigos desenvolvidos, siga os passos abaixo:

1. Clone o repositório:

CopiarEditar

git clone <https://github.com/nielreis07/microatividades.git>

2. Acesse a pasta do projeto:

CopiarEditar

cd microatividades

3. Execute os arquivos desejados:

CopiarEditar

```
python3 nome_do_arquivo.py
```

5. Considerações Finais

O desenvolvimento deste trabalho foi uma excelente oportunidade de consolidar o aprendizado teórico por meio da prática. A execução das microatividades permitiu não apenas a compreensão das estruturas de programação em Python, como também o desenvolvimento do raciocínio lógico e a organização de código.

Com esse projeto, foi possível explorar as bases da linguagem Python de maneira acessível e interativa. A experiência servirá como alicerce para desafios mais avançados no decorrer do curso e da futura atuação profissional.

6. Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023: Informação e documentação – Referências – Elaboração**. Rio de Janeiro, 2018.

PYTHON SOFTWARE FOUNDATION. *The Python Tutorial*. Disponível em: <https://docs.python.org/3/tutorial/>. Acesso em: 03 jun. 2025.

GITHUB. Daniel Victor – Microatividades em Python. Disponível em: <https://github.com/nielreis07/microatividades>. Acesso em: 03 jun. 2025.