

Relatório de Atividades Práticas em Python

Aluna: Regiane Lubanco de Souza

Disciplina: Programação em Python

Data: 10/06/2025

Objetivo Geral

Este relatório visa documentar a execução de microatividades práticas utilizando estruturas fundamentais da linguagem Python, com foco em condicionais, laços de repetição e funções. Ao final, aplica-se o conhecimento adquirido para refatorar uma calculadora simples, promovendo maior organização e reutilização de código.

Microatividade 1: Estruturas Condicionais – if e else

Objetivo: Compreender a utilização das estruturas de decisão simples if e compostas com else.

Código Base:

```
temperatura = 29
if temperatura < 30:
    print("A temperatura hoje está amena")
else:
    print("Hoje está fazendo calor")
```

Resultados: O programa respondeu corretamente às variações da variável, demonstrando o uso funcional das estruturas de decisão.

Microatividade 2: Estrutura elif

Objetivo: Utilizar múltiplas condições com if, elif e else.

Código Base:

```
tempoExperiencia = 5
if tempoExperiencia < 2:
    print("Nível de conhecimento júnior.")
elif 2 < tempoExperiencia < 5:
    print("Nível de conhecimento pleno.")
else:
    print("Nível de conhecimento sênior.")
```

Resultados: Os testes com valores 5, 1 e 3 confirmaram o correto uso das estruturas condicionais encadeadas.

Microatividade 3: Estrutura de Repetição while

Objetivo: Aprender o uso da estrutura de repetição while.

Código Base:

```
entrada_idade = ""
while str(entrada_idade) != '0':
    entrada_idade = input("Digite um número qualquer ou 0 para sair: ")
    print("Número digitado:", entrada_idade)
```

Resultados: A estrutura repetiu corretamente até o valor 0 ser digitado, validando o uso da lógica de repetição condicional.

Microatividade 4: Estrutura de Repetição for

Objetivo: Explorar a estrutura de repetição for aplicada a strings e intervalos numéricos.

Código Base:

```
texto = 'Olá, laço for.'
for item in texto:
    print("Caractere:", item)

for numero in range(1, 11):
    print("Número do intervalo:", str(numero))
```

Resultados: Impressão dos caracteres da string e dos números de 1 a 10, comprovando o uso adequado do for.

Microatividade 5: Declaração de Funções

Objetivo: Definir e invocar funções simples.

Código Base:

```
def imprimir_variavel():
    texto = "Olá, funções em Python"
    print(texto)

imprimir_variavel()
```

Resultados: A função foi corretamente definida e chamada, demonstrando encapsulamento de instruções.

Microatividade 6: Argumentos em Funções

Objetivo: Utilizar argumentos em funções para personalizar o comportamento do código.

Código Base:

```
def loginUsuario(perfil):  
    if perfil.lower() == 'admin':  
        print("Bem-vindo, Administrador")  
    else:  
        print("Bem-vindo, Usuário")
```

```
loginUsuario("Admin")  
loginUsuario("user")
```

Resultados: A função adaptou sua resposta de acordo com a entrada, mostrando entendimento de parâmetros e manipulação de strings.

Trabalho Prático – Refatoração de Calculadora

Objetivo: Aplicar os conceitos de estrutura condicional, repetição e funções na criação de uma calculadora otimizada.

Código Base (Resumo):

```
def adicao(a, b):  
    return a + b  
  
def subtracao(a, b):  
    return a - b  
  
def multiplicacao(a, b):  
    return a * b  
  
def divisao(a, b):  
    if b == 0:  
        return "Não foi possível realizar a divisão por 0"  
    return a / b  
  
def calculadora(a, b, operacao):  
    operacao = operacao.lower()  
    if operacao in ['+', 'adicao']:  
        return adicao(a, b)  
    elif operacao in ['-', 'subtracao']:  
        return subtracao(a, b)
```

```

elif operacao in ['*', 'multiplicacao']:
    return multiplicacao(a, b)
elif operacao in ['/', 'divisao']:
    return divisao(a, b)
else:
    return "Operação inválida"

saida = ""
while saida.lower() != 'n':
    n1 = float(input("Digite o primeiro número: "))
    n2 = float(input("Digite o segundo número: "))
    op = input("Digite a operação (+, -, *, /): ")
    resultado = calculadora(n1, n2, op)
    print("Resultado da operação:", resultado)
    saida = input("Deseja continuar? (S/N): ")

```

Resultados: A aplicação interativa executa operações matemáticas conforme a entrada do usuário, e permite múltiplas execuções. Utiliza funções e estrutura de repetição while, promovendo organização e robustez no código.

Conclusão

As microatividades permitiram a construção gradual do conhecimento prático sobre estruturas de controle, repetição e modularização em Python. A atividade prática final demonstrou a integração desses conhecimentos em um programa funcional e organizado. A experiência reforçou a importância da lógica de programação e da clareza na escrita de código.