

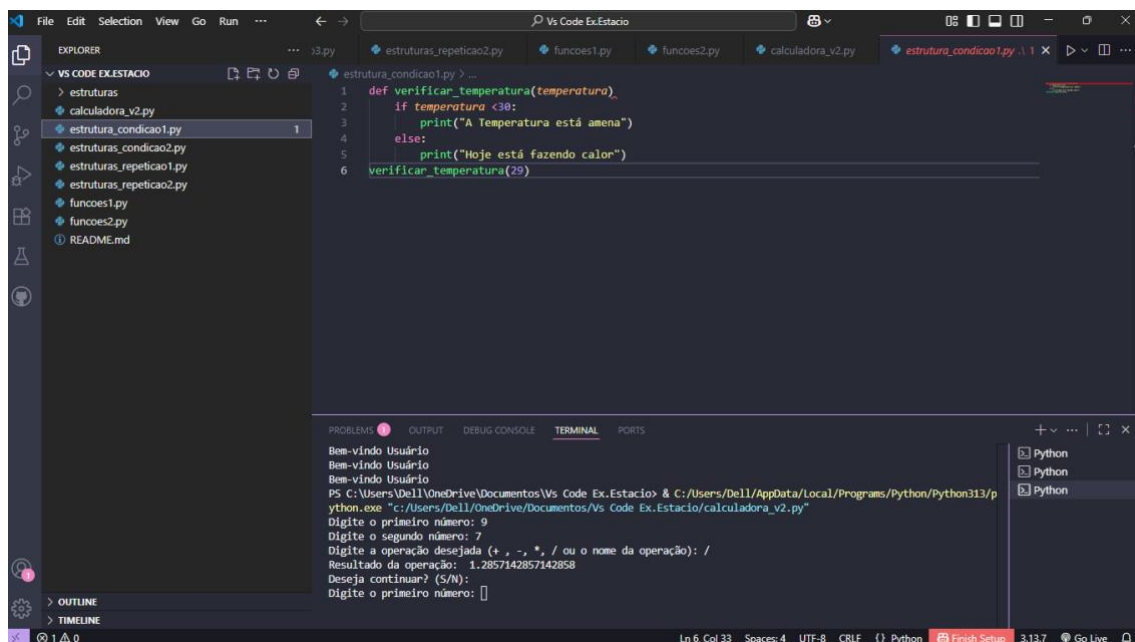
Relatório do Trabalho Prático

Lógica, Algoritmos e Programação de Computadores

1-Utilização das estruturas de Condição if e else em Python

As estruturas de condição if e else em Python são fundamentais para controlar o fluxo de execução de um programa com base em condições. Neste exemplo o programa verifica se a variável temperatura é menor que 30. Se for, imprime a mensagem: “A temperatura está amena. Se não, imprime a mensagem: “Hoje está fazendo calor”.

Comentário: A indentação é obrigatória para definir blocos de códigos.



The screenshot shows the VS Code interface with a file explorer on the left containing several Python files. The main editor displays a script named `estrutura_condicao1.py` with the following code:

```
1 def verificar_temperatura(temperatura):
2     if temperatura < 30:
3         print("A Temperatura está amena")
4     else:
5         print("Hoje está fazendo calor")
6     verificar_temperatura(29)
```

The bottom panel shows the terminal output, which includes a series of "Bem-vindo Usuário" messages and a calculation of the sum of two numbers (9 and 7) resulting in 1.2857142857142858.

2-Utilização das estruturas de Condição elif em Python

Elif significa “else if” e é utilizada para verificar várias condições diferentes. Se a primeira condição com if não for verdadeira, o Python verifica a próxima com elif e assim por diante. Neste exemplo da microatividade 2, utilizamos a condição if para checar se o valor da variável tempoExperiência é menor que 2. A condição elif oferece uma alternativa intermediária entre o if e o else, permitindo que outra ou múltiplas possibilidades sejam consideradas.

Comentário: Podemos utilizar elif quantas vezes forem necessárias. O Python avalia condições de cima para baixo e executa a primeira condição que for verdadeira.

The screenshot shows the VS Code interface with a file explorer on the left containing files like `estruturas_condicao1.py`, `estruturas_condicao2.py`, `estruturas_repeticao1.py`, `estruturas_repeticao2.py`, `funcoes1.py`, `funcoes2.py`, and `README.md`. The main editor displays `estruturas_condicao2.py` with the following code:

```
1 tempoExperiencia = 5
2
3 if tempoExperiencia < 2:
4     print("Nível de conhecimento júnior.")
5 elif tempoExperiencia > 2 and tempoExperiencia < 5:
6     print("Nível de conhecimento pleno.")
7 else:
8     print("Nível de conhecimento sênior.")
```

The bottom panel shows the TERMINAL with the following output:

```
Bem-vindo Usuário
 Bem-vindo Usuário
 Bem-vindo Usuário
 PS C:\Users\Dell\OneDrive\Documents\Vs Code Ex.Estacio> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/p
 ython.exe "C:/Users/Dell/OneDrive/Documents/Vs Code Ex.Estacio/calculadora_v2.py"
 Digite o primeiro número: 9
 Digite o segundo número: 7
 Digite a operação desejada (+, -, *, / ou o nome da operação): /
 Resultado da operação: 1.2857142857142858
 Deseja continuar? (S/N):
 Digite o primeiro número: 
```

3-Utilização das estruturas de repetição while em Python

A estrutura while permite que um trecho de código seja repetido enquanto a condição for verdadeira. O loop será interrompido quando a condição se tornar falsa.

Comentário: Se a condição nunca se tornar falsa o loop será executado infinitamente. É fundamental, por isso, verificar a condição dentro do while e garantir que o loop seja controlado corretamente dentro do bloco de códigos.

The screenshot shows the VS Code interface with the file explorer on the left. The main editor displays `estruturas_repeticao1.py` with the following code:

```
1 entrada_idade = 1
2 while entrada_idade != 0:
3     entrada_idade = int(input("Digite um número qualquer ou 0 para sair: "))
4     print("Número digitado: ", entrada_idade)
```

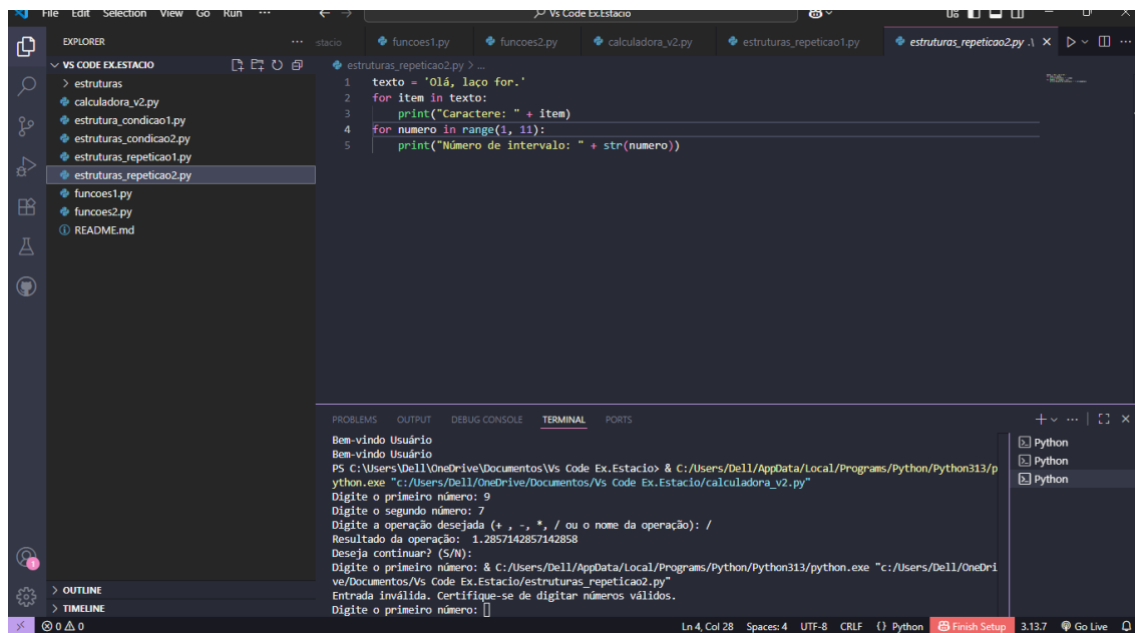
The bottom panel shows the TERMINAL with the following output:

```
Bem-vindo Usuário
 Bem-vindo Usuário
 Bem-vindo Usuário
 PS C:\Users\Dell\OneDrive\Documents\Vs Code Ex.Estacio> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/p
 ython.exe "C:/Users/Dell/OneDrive/Documents/Vs Code Ex.Estacio/calculadora_v2.py"
 Digite o primeiro número: 9
 Digite o segundo número: 7
 Digite a operação desejada (+, -, *, / ou o nome da operação): /
 Resultado da operação: 1.2857142857142858
 Deseja continuar? (S/N):
 Digite o primeiro número: 
```

4-Utilização das estruturas de repetição for em Python

A estrutura de repetição for em Python é usada para iterar sobre elementos de uma sequência (como listas, tuplas, strings, dicionários, conjuntos e objetos iteráveis) e executa um bloco de código para cada elemento. Ao chamar o método range, o Python cria uma sequência de números inteiros.

Comentário: O print dentro do loop for está indentado e será executado a cada repetição do loop; fora do loop, ele será executado apenas uma vez após o término do loop.



```
1 texto = 'Olá, laço for.'
2 for item in texto:
3     print("Caractere: " + item)
4 for numero in range(1, 11):
5     print("Número de intervalo: " + str(numero))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Bem-vindo Usuário
Bem-vindo Usuário
PS C:\Users\Dell\OneDrive\Documents\Vs Code Ex.Estacio> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/OneDrive/Documents/Vs Code Ex.Estacio/calculadora_v2.py"
Digite o primeiro número: 9
Digite o segundo número: 7
Digite a operação desejada (+, -, *, / ou o nome da operação): /
Resultado da operação: 1.2857142857142858
Deseja continuar? (S/N):
Digite o primeiro número: & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/OneDrive/Documents/Vs Code Ex.Estacio/estruturas_repeticao2.py"
Entrada inválida. Certifique-se de digitar números válidos.
Digite o primeiro número: [

5-Utilização das funções em Python

Funções são blocos de código reutilizáveis que nos permitem encapsular tarefas específicas e executá-las quando necessário. Neste exemplo a seguir, a função chamada fora do escopo executa sua tarefa sem depender de entradas externas para executar o que está nela.

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

imprimir_variavel()
```

Terminal output:

```
Bem-vindo Usuário
Bem-vindo Usuário
PS C:\Users\Dell\OneDrive\Documents\Vs Code Ex.Estacio> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/OneDrive/Documents/Vs Code Ex.Estacio/calculadora_v2.py"
Digite o primeiro número: 9
Digite o segundo número: 7
Digite a operação desejada (+, -, *, / ou o nome da operação): /
Resultado da operação: 1.2857142857142858
Deseja continuar? (S/N):
Digite o primeiro número: & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/OneDrive/Documents/Vs Code Ex.Estacio/estruturas_repeticao2.py"
Entrada inválida. Certifique-se de digitar números válidos.
Digite o primeiro número: 
```

6-Utilização de argumentos de funções em Python

Argumentos são valores passados para uma função quando ela é chamada. Permitem que a função receba dados externos e execute ações com base nesses dados.

Comentário: Quando os valores dos argumentos são fornecidos pelo usuário, estabelece-se uma interface dinâmica entre o sistema e o usuário, promovendo flexibilidade e personalização na execução de tarefas.

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

loginUsuario('Admin')
loginUsuario('admin')
loginUsuario('User')
loginUsuario('usuário')
loginUsuario('etc')
```

Terminal output:

```
Bem-vindo Usuário
Bem-vindo Usuário
PS C:\Users\Dell\OneDrive\Documents\Vs Code Ex.Estacio> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/OneDrive/Documents/Vs Code Ex.Estacio/calculadora_v2.py"
Digite o primeiro número: 9
Digite o segundo número: 7
Digite a operação desejada (+, -, *, / ou o nome da operação): /
Resultado da operação: 1.2857142857142858
Deseja continuar? (S/N):
Digite o primeiro número: & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/OneDrive/Documents/Vs Code Ex.Estacio/estruturas_repeticao2.py"
Entrada inválida. Certifique-se de digitar números válidos.
Digite o primeiro número: 
```

