

# Trabalho Prático | DGT2817 LÓGICA, ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

Material de **orientações** para desenvolvimento do **Trabalho Prático** da disciplina  
DGT2817 Lógica, Algoritmos e Programação de Computadores..

**ATENÇÃO:** a nota deste trabalho prático representa 100% da nota desta disciplina.  
Esta disciplina **NÃO** tem simulados ou alternativas de recuperação de nota. Desta  
forma, interaja com o tutor via fórum da SAVA quantas vezes forem necessárias  
para você obter o entendimento necessário para o desenvolvimento deste  
trabalho. Fique atento ao prazo de entrega.

**Objetivos da prática**

- Descrever a utilização das estruturas de condição if e else em Python;
- Descrever a utilização da estrutura de condição else if (elif) em Python;
- Descrever a utilização da estrutura de repetição while em Python;
- Descrever a utilização da estrutura de repetição for em Python;
- Descrever a utilização de funções em Python;
- Descrever a utilização argumentos de funções no Python;
- Refazer a calculadora utilizando estrutura condicional e funções;

## Entrega e Progresso

- As microatividades irão dar suporte para o desenvolvimento do Trabalho Prático. Elas têm apoio/gabarito para resolução no próprio documento;
- Teremos uma entrega intermediária das Microatividades 1-2-3 e a entrega final da disciplina que é o Trabalho Prático;
- Após concluir as microatividades, compartilhe com o professor o link do GITHUB
- ATENÇÃO: a nota deste trabalho representa 100% da nota desta disciplina. Esta disciplina NÃO tem simulados ou alternativas de recuperação de nota. Desta forma, interaja com o tutor via fórum da SAVA quantas vezes forem necessárias para você obter o entendimento necessário para o desenvolvimento deste trabalho. Fique atento ao prazo de entrega.

## Atividades práticas

# Microatividade 1: Descrever a utilização das estruturas de condição if e else em Python

## - Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

## - Procedimentos

1. Abra a IDE VS Code;
2. No menu File, selecione a opção “Open Folder”;
3. Crie uma nova pasta em seu computador para armazenar os códigos desse conjunto de microatividades e a selecione a partir do VS Code;
4. No VS Code, menu Explorer, clique na pasta do projeto e crie um novo arquivo/script chamado “estruturas\_condicao1.py”;
5. No script criado:
  - a. Crie uma variável chamada temperatura e atribua a ela o valor 29;
  - b. Crie uma verificação, utilizando a condição if, para checar se o valor da variável temperatura é menor que 30;
  - c. Caso positivo, imprima na tela o texto ‘A temperatura hoje está amena’;
  - d. Caso contrário, e utilizando a condição else, imprima na tela o texto ‘Hoje está fazendo calor’;
6. Salve o arquivo/script;
7. Na barra superior direita você verá um ícone no formato de um triângulo deitado, semelhante à imagem abaixo:

semelhante a imagem abaixo:

8. Clique no ícone. A seguir, deverá ser aberta uma janela, na parte inferior do VS Code, exibindo o terminal e a saída dos comandos acima, que é a exibição da frase digitada após o comando “print”;
9. Caso ocorra algum erro na execução do script, verifique o seu código, corrigindo eventuais erros, e tente executá-lo novamente;
10. Ainda no script, altere o valor da variável temperatura para 31, salve a alteração e execute novamente o script.

#### - Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre as estruturas de condição if e else e se é capaz de criar instruções simples utilizando-as.

## Microatividade 2: Descrever a utilização da estrutura de

.. ~ . . . . .

# condição else if (elif) em Python

## - Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

## - Procedimentos:

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada na microatividade anterior, crie um novo arquivo/script chamado “estruturas\_condicao2.py”;
3. Nesse novo script:
  - a. Crie uma variável chamada tempoExperiencia e atribua a ela o valor 5;
  - b. Crie uma verificação, utilizando a condição if, para checar se o valor da variável tempoExperiencia é menor que 2;
  - c. Caso positivo, imprima na tela o texto ‘Nível de conhecimento júnior.’;
  - d. Após as instruções acima, crie uma outra condição utilizando elif (else if) para verificar se o valor da variável tempoExperiencia é maior que 2 e menor que 5. Em caso positivo, imprima o texto ‘Nível de conhecimento pleno.’
  - e. Por fim, crie uma condição else e imprima o texto ‘Nível de conhecimento sênior.’;
4. Salve o arquivo/script e o execute;
5. Altere o script, modificando o valor da variável tempoExperiencia para 1. Salve e execute;
6. Por fim, altere novamente o script, modificando o valor da variável tempoExperiencia para 3. Salve e execute.

## - Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre as estruturas de condição if, else if e else e se é capaz de criar instruções simples utilizando-as.

# Microatividade 3: Descrever a utilização da estrutura de repetição while em Python

## - Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

## - Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada nas microatividades anteriores, crie um novo arquivo/script chamado “estruturas\_repeticao1.py”;
3. Nesse novo script:
  - a. Crie uma variável chamada entrada\_idade e atribua a ela o valor “”;
  - b. Crie uma instrução while que verifique se o valor atribuído à variável entrada\_idade é diferente de 0 (como o valor inicial atribuído à variável é “”, isso a definiu como tipo string. Logo, a verificação no While deve ser feita com auxílio da instrução str );
  - c. No escopo da instrução while, atribua à variável entrada\_idade um input de entrada de dados com o texto ‘Digite um número qualquer ou 0 para sair: ‘;
  - d. Imprima, na tela, o número digitado pelo usuário precedido do texto ‘Número digitado: ‘;

4. Salve o arquivo/script e o execute;
5. Teste diferentes valores como entrada de dados, incluindo o número 0 - que deverá fazer com que a execução do programa seja interrompida. Caso isso não ocorra, verifique seu código – sobretudo a comparação na instrução While.

#### - Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre a estrutura de repetição while, sendo capaz de utilizá-la num programa simples.

## Microatividade 4: Descrever a utilização da estrutura de repetição for em Python

#### - Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

#### - Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada nas microatividades anteriores, crie um novo arquivo/

script chamado “estruturas\_repeticao2.py”;

3. Nesse novo script:
  - a. Crie uma variável chamada texto e atribua a ela o valor ‘Olá, laço for.’;
  - b. Crie uma instrução for que itere sobre a variável texto atribuindo cada um de seus caracteres a uma variável chamada item;
  - c. Imprima, na tela, dentro do escopo do laço for, o valor da variável item precedido do texto ‘Caractere: ‘;
4. Salve o arquivo/script e o execute;
5. Crie, no mesmo script, uma nova instrução for que:
  - a. Itere sobre um intervalo numérico entre 1 e 10 (dica: use a instrução range);
  - b. Imprima, na tela, dentro do escopo do laço for, o valor de cada número no intervalo acima precedido do texto ‘Número do intervalo: ‘;
  - c. Lembre-se de utilizar a instrução str para concatenar o valor inteiro com a string no momento de imprimir o valor pedido na tela.

#### - Resultados Esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos sobre a estrutura de repetição for e se é capaz de usá-la em diferentes cenários.

## Microatividade 5: Descrever a utilização de funções em Python



#### - Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

#### - Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada nas microatividades anteriores, crie um novo arquivo/script chamado “funcoes1.py”;
3. No script recém criado:
  - a. Defina uma função chamada “imprimir\_variavel”. Tal instrução não receberá parâmetros;
  - b. No escopo da função acima, crie uma variável chamada texto e atribua a ela o valor ‘Olá, funções em Python’;
  - c. A seguir, ainda no escopo da função, imprima na tela o valor da variável texto;
  - d. Por último, fora do escopo da função, faça a chamada da mesma.
4. Salve o arquivo/script e o execute. Você deverá ver na tela o conteúdo da variável texto.

#### - Resultados esperados ✨

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos básicos sobre a declaração e chamada de funções na linguagem Python.

# Microatividade 6: Descrever a utilização argumentos de funções no Python

## - Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

## - Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta utilizada nas microatividades anteriores, crie um novo arquivo/script chamado “funcoes2.py”;
3. No script acima:
  - a. Defina uma função chamada “loginUsuario”. Tal instrução receberá cini parâmetro a variável perfil;
  - b. No escopo da função, verifique se o valor do parâmetro perfil é igual a ‘admin’.  
Dica: considere que o usuário poderá digitar letras maiúsculas e/ou minúsculas na entrada de dados. Portanto, utilize a instrução lower no momento de fazer a verificação;
  - c. Caso o valor do parâmetro seja igual a ‘admin’, imprima na tela: ‘Bem-vindo, Administrador’. Do contrário, imprima: ‘Bem-vindo, Usuário’;
  - d. Por último, fora do escopo da função, faça a chamada da mesma passando diferentes valores como parâmetro. Ex:
    - i. Admin
    - ii. admin
    - iii. User
    - iv. usuário

v. etc.

4. Salve o arquivo/script a cada alteração no valor do parâmetro na chamada da função e o execute.

#### - Resultados esperados

O resultado esperado dessa microatividade é verificar se o aluno possui conhecimentos básicos sobre a declaração e chamada de funções com a utilização de parâmetros na linguagem Python.

## Trabalho Prático



Através dessa atividade o aluno irá refatorar o código de uma calculadora, acrescentando ao mesmo alguns dos recursos da linguagem Python vistos ao longo das microatividades.

### Contextualização

Após o lançamento da primeira versão da calculadora (desenvolvida por você anteriormente), será necessário refatorar a aplicação, modificando e otimizando fragmentos do código a partir do uso de recursos como estruturas de condição, estruturas de repetição e funções. Todos os requisitos para essa tarefa podem ser

encontrados no Roteiro de Prática.

## Roteiro de prática

### - Material necessário para a prática

- Interpretador Python instalado no Sistema Operacional;
- IDE VS Code instalada no Sistema Operacional;

### - Procedimentos

1. Abra a IDE VS Code;
2. Na mesma pasta onde criou os scripts utilizados nas microatividades, crie um novo script chamado “calculadora\_v2.py”;
3. No script:
  - a. Crie uma variável chamada `saida` e atribua a ela o valor “”;
  - b. Crie uma função chamada `adicao`. Tal função deverá receber dois parâmetros e retornar a soma entre ambos;
  - c. Crie uma função chamada `subtracao`. Tal função deverá receber dois parâmetros e retornar a subtração entre ambos;
  - d. Crie uma função chamada `multiplicacao`. Tal função deverá receber dois parâmetros e retornar a multiplicação entre ambos;
  - e. Crie uma função chamada `divisao`. Tal função deverá receber dois parâmetros, verificar se um deles é igual a 0. Em caso positivo, deverá retornar a mensagem “Não foi possível realizar a divisão por 0”. Em caso negativo, deverá retornar a divisão entre ambos;
  - f. Crie uma função chamada `calculadora`. Tal função deverá receber três parâmetros, sendo eles: os dois números que serão usados para os cálculos e a operação matemática que se deseja realizar. Sobre esse último parâmetro, você poderá utilizar tanto o sinal da operação quanto o seu nome;
  - g. No corpo da função `calculadora` você deverá verificar qual a operação desejada pelo usuário, checando o valor do parâmetro correspondente. Utilize estruturas

de condição para isso e, dependendo da operação desejada, você deverá chamar a função relativa a ela, passando as variáveis contendo os dois números para serem utilizados no cálculo. Armazene o resultado da chamada às funções de cálculo numa variável chamada resultado. Ao final da função calculadora você deverá retornar a variável resultado;

- h. Crie um laço while e, como condição do mesmo, verifique se o valor da variável saída é diferente de n. Lembre-se de que o usuário poderá inserir tanto N quanto n;
- i. No escopo do laço while peça ao usuário para digitar o primeiro número e armazene seu valor numa variável. Faça o mesmo para o segundo número e para a operação matemática. Passe essas três variáveis para o método calculadora, armazenando o retorno dessa chamada numa variável também chamada resultado. Imprima na tela o valor da variável resultado precedido pelo texto 'Resultado da operação: '. Por fim, pergunte ao usuário se ele deseja continuar ou não executando o programa. Armazene tal input na variável saída;
- j. Tome cuidado com a condição de verificação do laço for em relação à entrada do usuário armazenada na variável saída. Em outras palavras, deixe claro para o usuário as respostas possíveis para a pergunta se ele deseja sair. Use, por exemplo, S/N. Com isso você poderá considerar um desses dois valores na verificação do laço para saber se deve continuar executando o programa ou se deve encerrá-lo.

4. Salve as alterações no script e o execute via VS Code;

5. Teste o aplicativo interagindo com ele através do prompt, fornecendo os dados necessários para a sua execução.

### - Resultados esperados ✨

Os passos anteriores visam a construção de um aplicativo completo com a estrutura básica do Flutter, garantindo que as interfaces de usuário sejam responsivas e acessíveis em dispositivos móveis, tablets e desktops. Essa plataforma permitirá que os clientes compartilhem avaliações e comentários sobre suas experiências de viagem. Além disso, a Agência de viagens utilizará widgets interativos do Flutter para exibir as avaliações dos clientes e interagir com eles de forma envolvente, compartilhando suas opiniões e proporcionando uma experiência interativa e informativa aos usuários do site. A imagem a seguir ilustra um exemplo de aplicativo desenvolvido. Adapte os códigos apresentados para que você possa criar a sua solução.

## 📌 Referências

Não foram utilizadas referências bibliográficas para a elaboração das atividades.

# Entrega da prática

## Chegou a hora, gamer!

👉 Armazene o projeto em um repositório no GIT.

👉 Anexar a documentação do projeto (PDF) no GIT.

👉 Compartilhe o link do repositório do GIT com o seu tutor para correção da prática, por meio da **Sala de Aula Virtual**, na aba "**Trabalhos**" do respectivo nível de conhecimento.

👉 **Ei, não se esqueça de entregar este trabalho na data estipulada na aba TRABALHOS da SAVA!**

Feito com o Microsoft Sway

Crie e compartilhe apresentações, histórias pessoais,  
relatórios interativos e muito mais.

Introdução

