Lista 2: Introdução aos Princípios de Programação em Python

Aluno: Pedro Ailton Alves da Cunha

Disciplina: Princípios de Programação

Professor: Cleyton Magalhães

Entrega: 26/06/25

Link da Atividade no Github:
https://github.com/pedroailton/Principios_de_Programacao/tree/37a1c0767ffa7b206d5fc006ce842eb4c09caf81/Lista%20de%20Exerc%C3%ADcios%202.

1. Representação de Algoritmos
   Situação escolhida: Colocar o lixo para fora de casa.

   ■ Em descrição narrativa (texto narrativo com o passo a passo).

   Se for terça-feira, quinta-feira ou sábado

   Vá na cozinha até a gaveta de sacolas de compras

   Se houver 3 sacolas ou mais,

   Pegue 3 sacolas;

   Senão,

   Pegue a quantidade de sacolas que houver

   E Procure o restante das sacolas em outro lugar para completar 3;

   Ponha as 3 sacolas de plástico na mesa da cozinha;

   Pegue uma das sacolas;

   Vá no primeiro banheiro da casa;

   Abra o lixeiro;

   Recolha o saco de lixo de dentro do lixeiro;

   Coloque o novo saco no lixeiro;

Feche o lixeiro;

Repita o processo para os outros 2 lixeiros da casa;

Amarre juntos os 3 sacos de lixo;

Se a porta estiver fechada,

E se a porta estiver trancada, procure a chave para destrancar a porta e destranque a porta

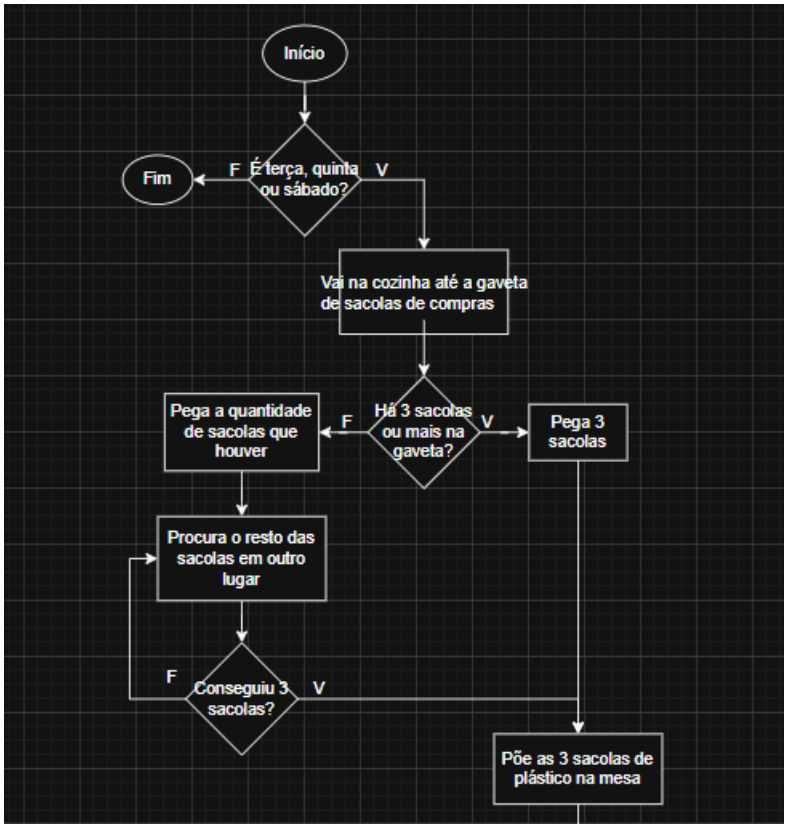
Abra a porta

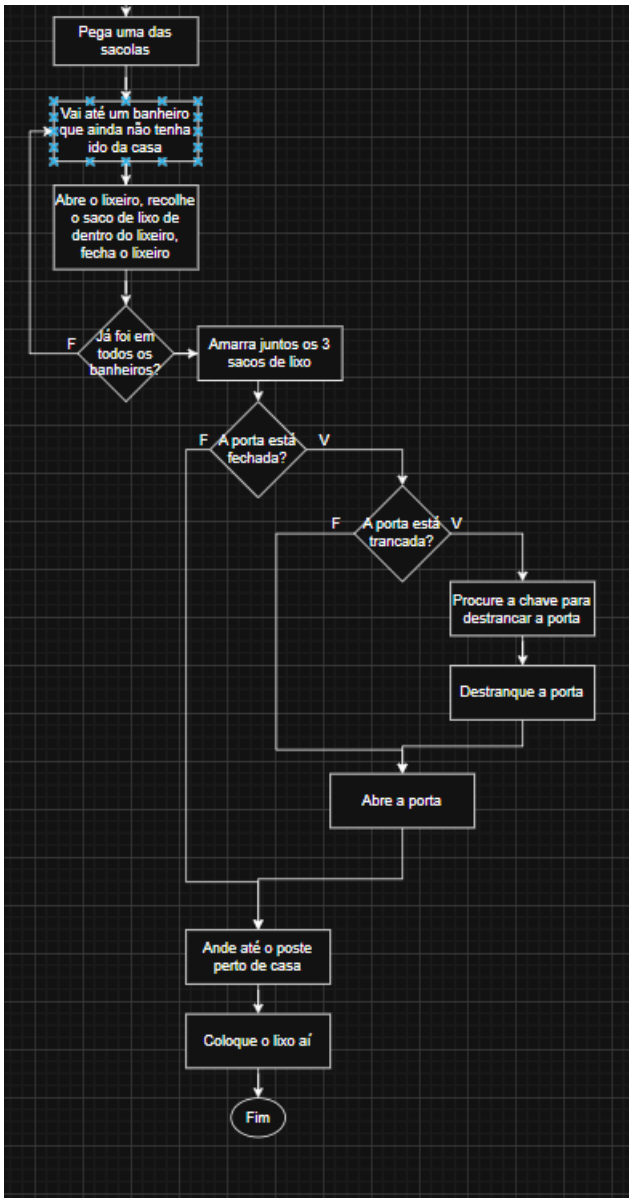
Ande até perto do poste perto de casa;

Coloque o lixo aí.

■ Em forma de fluxograma (criar um fluxograma para a tarefa).

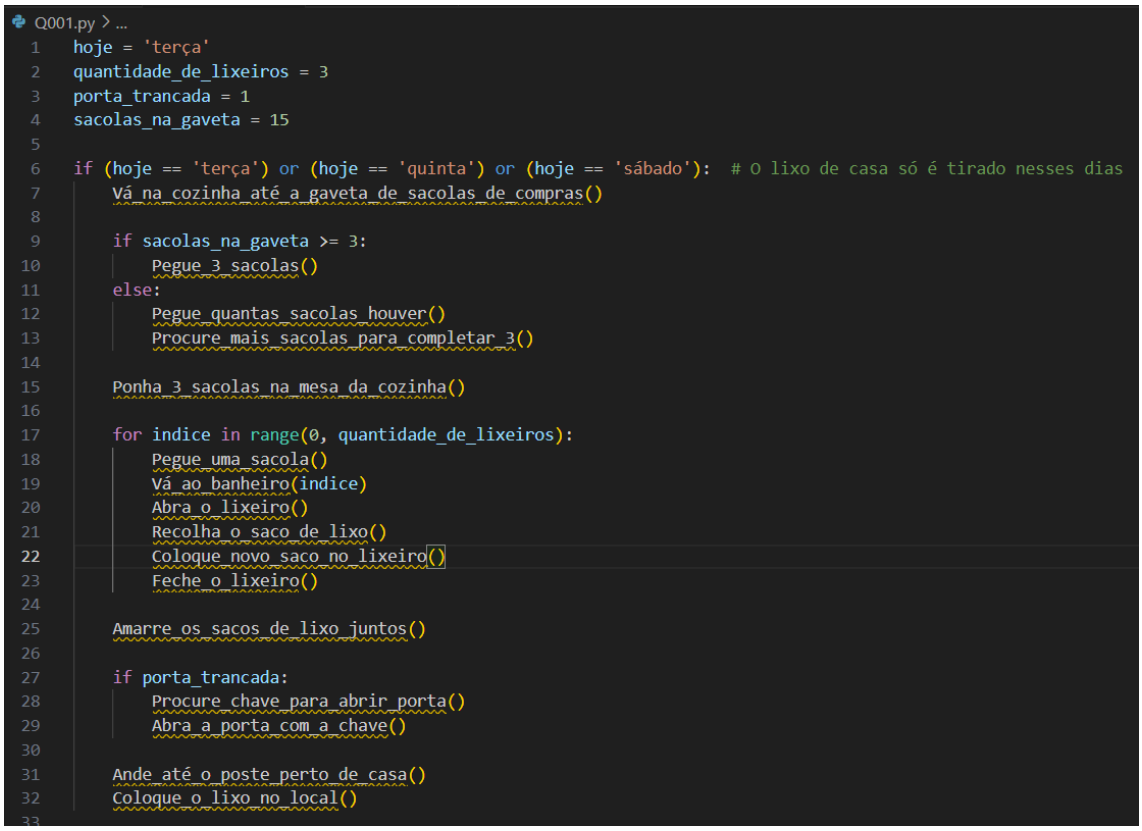
O arquivo .drawio do Fluxograma pode ser encontrado na pasta Q001.





■ Em código Python

O arquivo .py do Código pode ser encontrado na pasta Q001.



```python
hoje = 'terça'
quantidade_de_lixeiros = 3
porta_trancada = 1
sacolas_na_gaveta = 15

if (hoje == 'terça') or (hoje == 'quinta') or (hoje == 'sábado'):  # O lixo de casa só é tirado nesses dias
    Vá_na_cozinha_até_a_gaveta_de_sacolas_de_compras()

    if sacolas_na_gaveta >= 3:
        Pegue_3_sacolas()
    else:
        Pegue_quantas_sacolas_houver()
        Procure_mais_sacolas_para_completar_3()

    Ponha_3_sacolas_na_mesa_da_cozinha()

    for indice in range(0, quantidade_de_lixeiros):
        Pegue_uma_sacola()
        Vá_ao_banheiro(indice)
        Abra_o_lixeiro()
        Recolha_o_saco_de_lixo()
        Cologue_novo_saco_no_lixeiro()
        Feche_o_lixeiro()

    Amarre_os_sacos_de_lixo_juntos()

    if porta_trancada:
        Procure_chave_para_abrir_porta()
        Abra_a_porta_com_a_chave()

    Ande_até_o_poste_perto_de_casa()
    Coloque_o_lixo_no_local()
```

2. Sintaxe da Linguagem: Palavras-chave, Identificadores e Indentação

○ Atividade 2: Responda:

■ O que são palavras-chave em Python. Inclua alguns exemplos.

Palavras-chave (ou *keywords*) são palavras reservadas pela linguagem Python com significados especiais. Elas são reconhecidas pelo interpretador e não podem ser usadas como nomes de variáveis, funções ou identificadores. Essas palavras controlam o fluxo do programa, definem estruturas, funções, classes, e realizam diversas instruções essenciais.

Exemplos: if, else, elif – usadas em estruturas condicionais; for, while – usadas em laços de repetição; def – usada para definir funções; return – usada para retornar valores de uma função.

■ O que são identificadores e explique as regras para nomeação em Python.

Identificadores são os nomes dados a variáveis, funções, classes, objetos e módulos em um programa Python. Eles servem para identificar elementos no código.

Regras para nomeação de identificadores em Python:

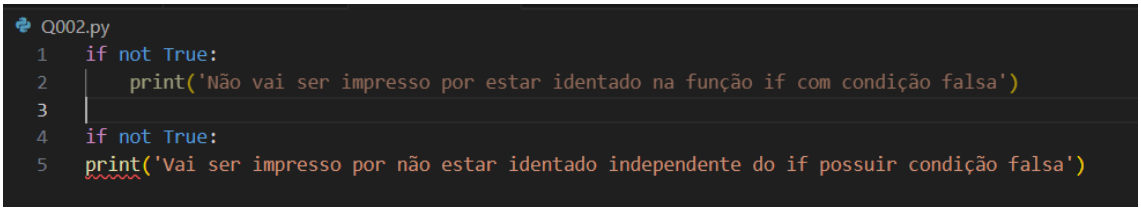
1. Devem começar com uma letra (a–z, A–Z) ou com sublinhado _.
2. Podem conter letras, números e sublinhados (não podem conter espaços ou símbolos especiais como @, %, $, etc.).
3. Não podem usar palavras-chave da linguagem, como def, class, if, entre outras.
4. São sensíveis a maiúsculas e minúsculas – ou seja, nome e Nome são identificadores diferentes.
5. Não podem começar com números (por exemplo, 2variavel é inválido).

Boas práticas:

- Variáveis e funções devem ser nomeadas usando snake_case.
- Classes devem usar CamelCase.
- Constantes são escritas em MAIÚSCULAS.

■ A importância da indentação e como ela é usada em Python. Use um exemplo.

Ela é importante para alocar blocos de código inseridos em uma estrutura de controle, função ou classe. Um exemplo é com a função “if”, a seguir:



■ Cite exemplos de problemas de sintaxe que você teve ao tentar programar em Python.

- Esquecer de colocar “:” no fim da linha de uma estrutura de controle;

- Misturar espações e tabulações;

- Errar a quantidade de parênteses em funções dentro de outras

- Não fechar aspas;

- Indentação incorreta ou faltando, fazendo o bloco de código não se comportar da maneira planejada.

3. Simulação de um processo seletivo

FASE 1:

3.1. Descrição narrativa, uma vez que vai se tratar de uma simples transcrição de como o sistema está entendido na minha mente, de modo que eu possa confirmar o entendimento com o possível usuário ou cliente e debater sobre o sistema a ser desenvolvido de maneira rápida e simples.

3.2. Utilizaria os fluxogramas, de modo a destacar cada parte e funcionalidade separadamente, apresentando os fluxos (incluindo os de erro, facilmente percebidos com essa forma de representação de sistema) de maneira descomplicada para os stakeholders nas reuniões e entregas.

FASE 2:

3.3. Fluxogramas em anexo na subpasta Fluxogramas, dentro da subpasta Q003.

FASE 3:

3.4. Código em Python “cardume_kanban.py” em anexo na subpasta Q003.

Obs: Esse código funciona com o banco de dados SQLite, que vai gerar um arquivo chamado “cardume.db” na execução do programa.