

# Travessia do Amostradinho no Labirinto

Curso técnico de Desenvolvimento de Sistemas  
(TDS 1º ano – IFMA, Viana-MA, 2024)

Estudo de caso proposto como exercício de programação:  
variáveis, controle de fluxo, laços de repetição, funções, etc. na  
disciplina de Algoritmos e Programação.

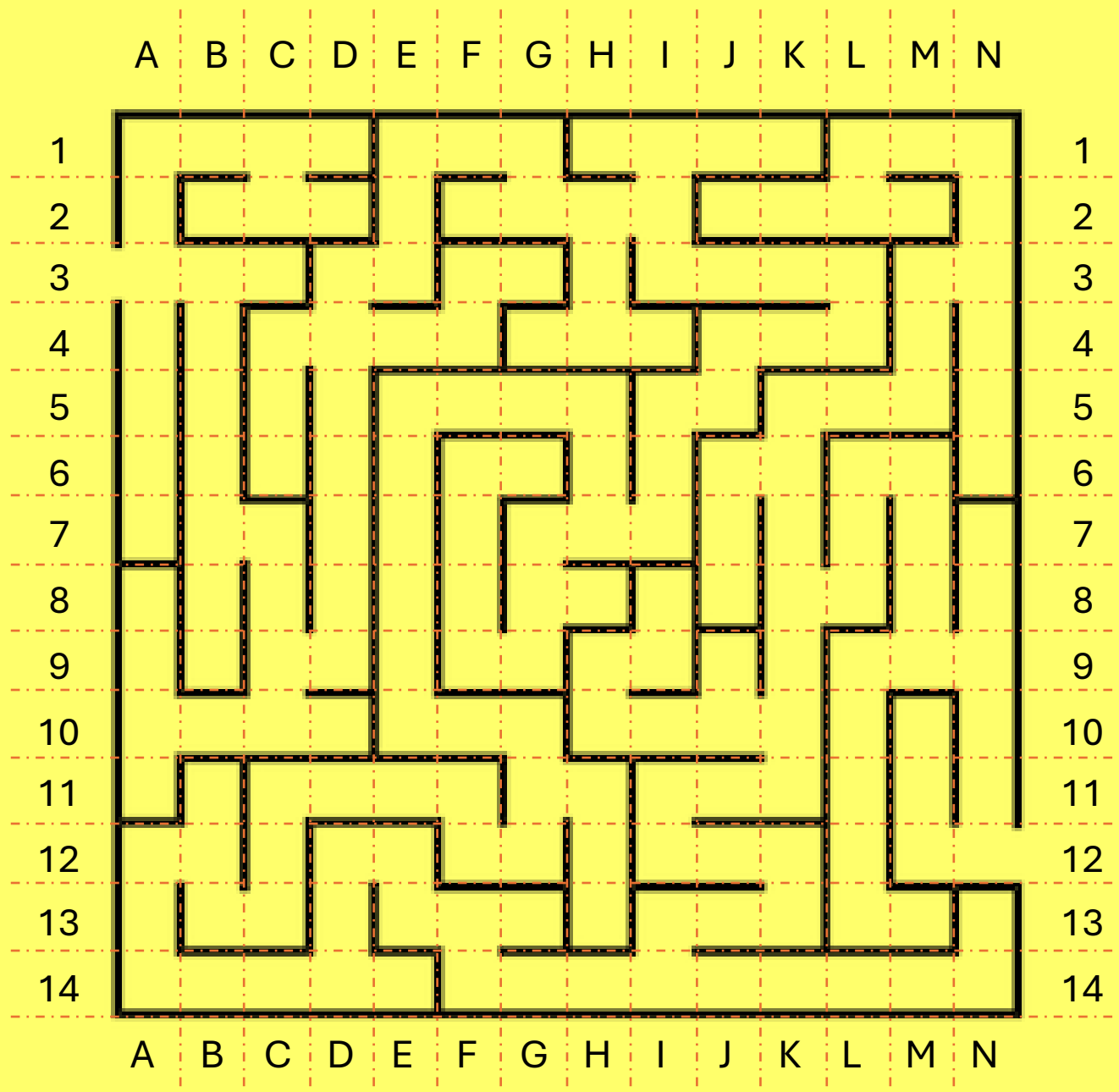
Professor Douglas Gonçaze

# Travessia do Amostradinho no Labirinto



Amostradinho é o personagem que aceitou o desafio de atravessar o labirinto com a nossa ajuda.

Amostradinho é um androide cujo idioma é a linguagem Python. Em outras palavras, precisamos passar as orientações usando a programação em Python.



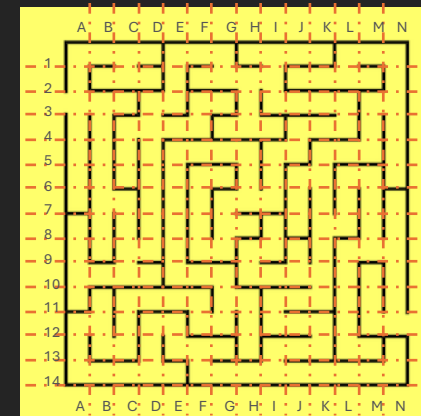
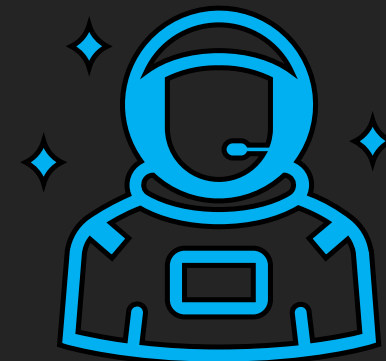
# Regras do Labirinto

1. O início do desafio do Labirinto começa com o Amostradinho se posicionando em A3, ou seja, na coluna da letra A e na linha de número 3.
2. Os movimentos dentro do Labirinto devem ser realizados considerando uma posição por vez. Proibido se mover de A3 para C3 sem primeiro passar por B3.
3. O caminhar do Amostradinho sempre deverá ser para frente. Se o Amostradinho quiser ir para a esquerda ou para a direita deve primeiramente virar-se na direção desejada e somente depois caminhar.
4. As paredes do Labirinto são intransponíveis, ou seja, o Amostradinho não tem o poder para: atravessar paredes, escalar paredes, voar sobre as paredes e nem teletransportar-se.
5. A travessia será concluída a partir do momento em que Amostradinho se posicionar na coluna de letra N e na linha de número 12.
6. O professor recomenda que o estudante tente se divertir neste exercício. Faça em grupo para ser ainda mais divertido!

# Desenvolva a sua estratégia

Desenvolva a sua estratégia de travessia do labirinto usando as capacidades da linguagem de programação Python:

- Variáveis;
- Estruturas condicionais ou controle de fluxo (IF, ELIF, ELSE);
- Estruturas de repetição (FOR e WHILE);
- Funções (DEF);



# Desenvolva a sua estratégia

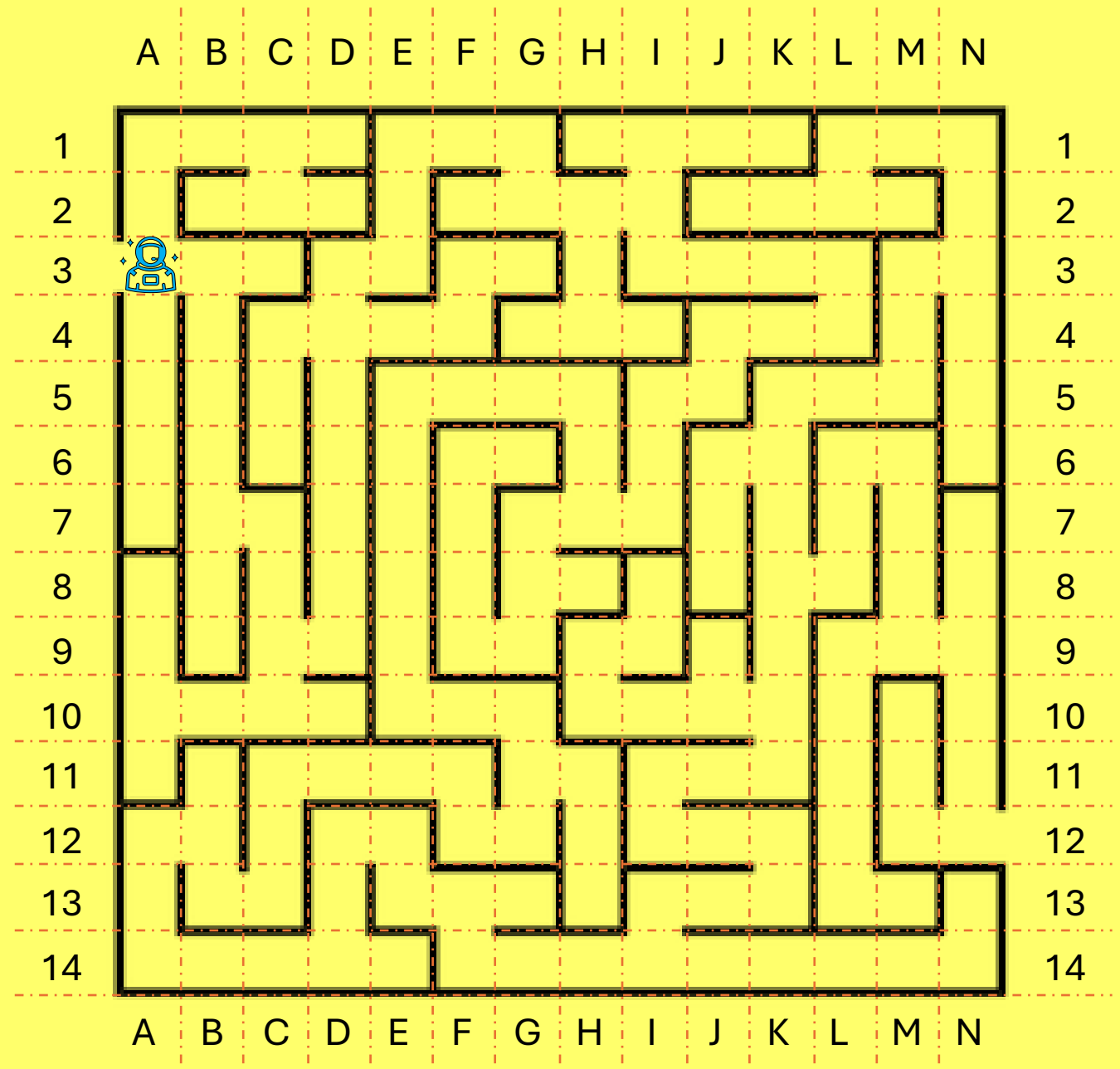
A partir de agora você deve desenvolver a sua estratégia em linguagem Python para ajudar o Amostradinho atravessar a o labirinto.

Por exemplo, crie uma função para determinar a posição do Amostradinho:

```
def posicao_atual(posicao)
```

Outra função para informar a direção para o Amostradinho:

```
def apontar_para()
```



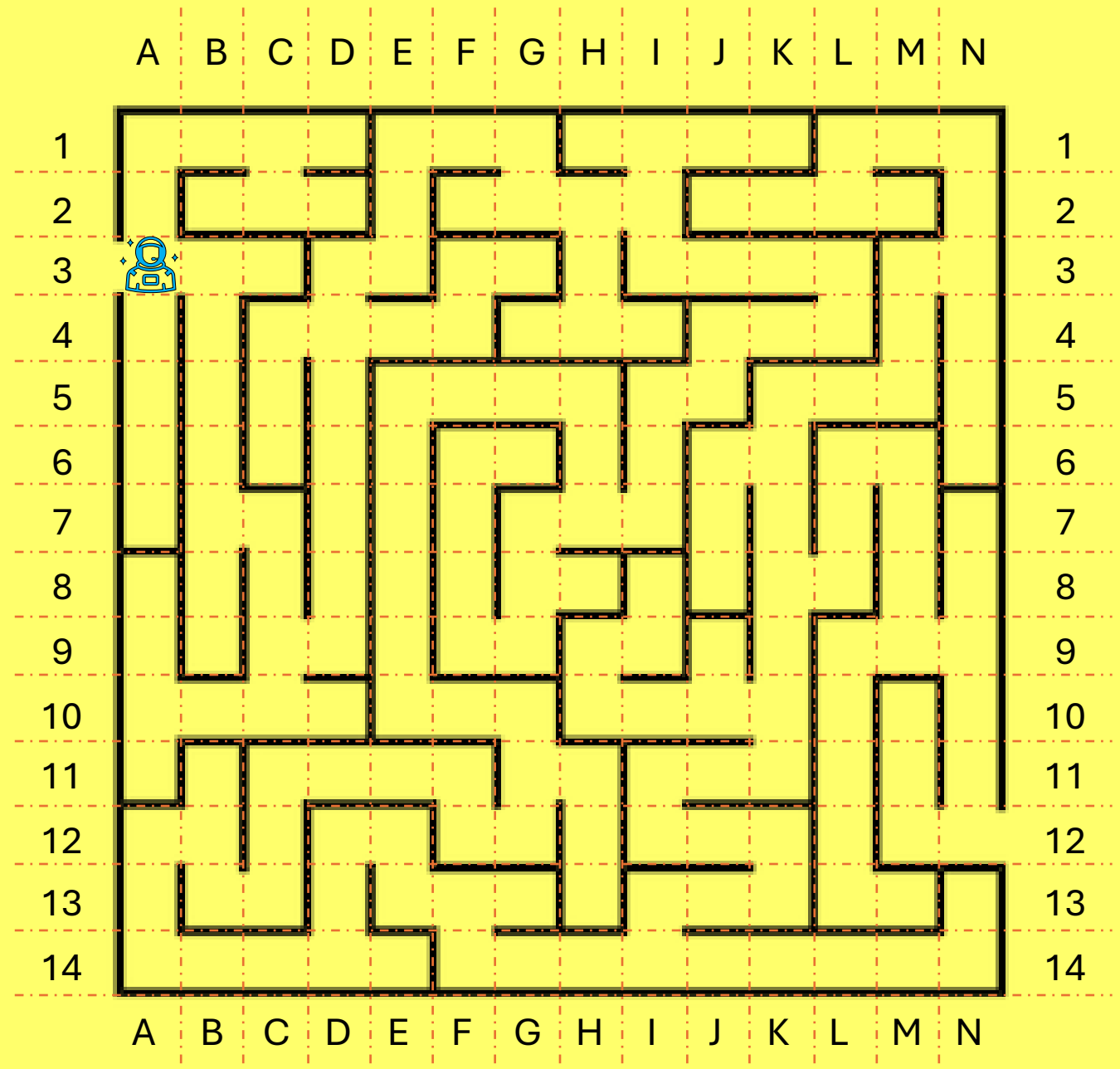
# Desenvolva a sua estratégia

Outra estratégia que poderia ser adotada no lugar da função `apontar_para()` poderia ser:

**`def va_para()`**

Informando assim a letra da coluna e o número da linha. Por exemplo, B3.

Mas, fique a vontade para desenvolver com liberdade a sua estratégia.



# Exemplo de execução



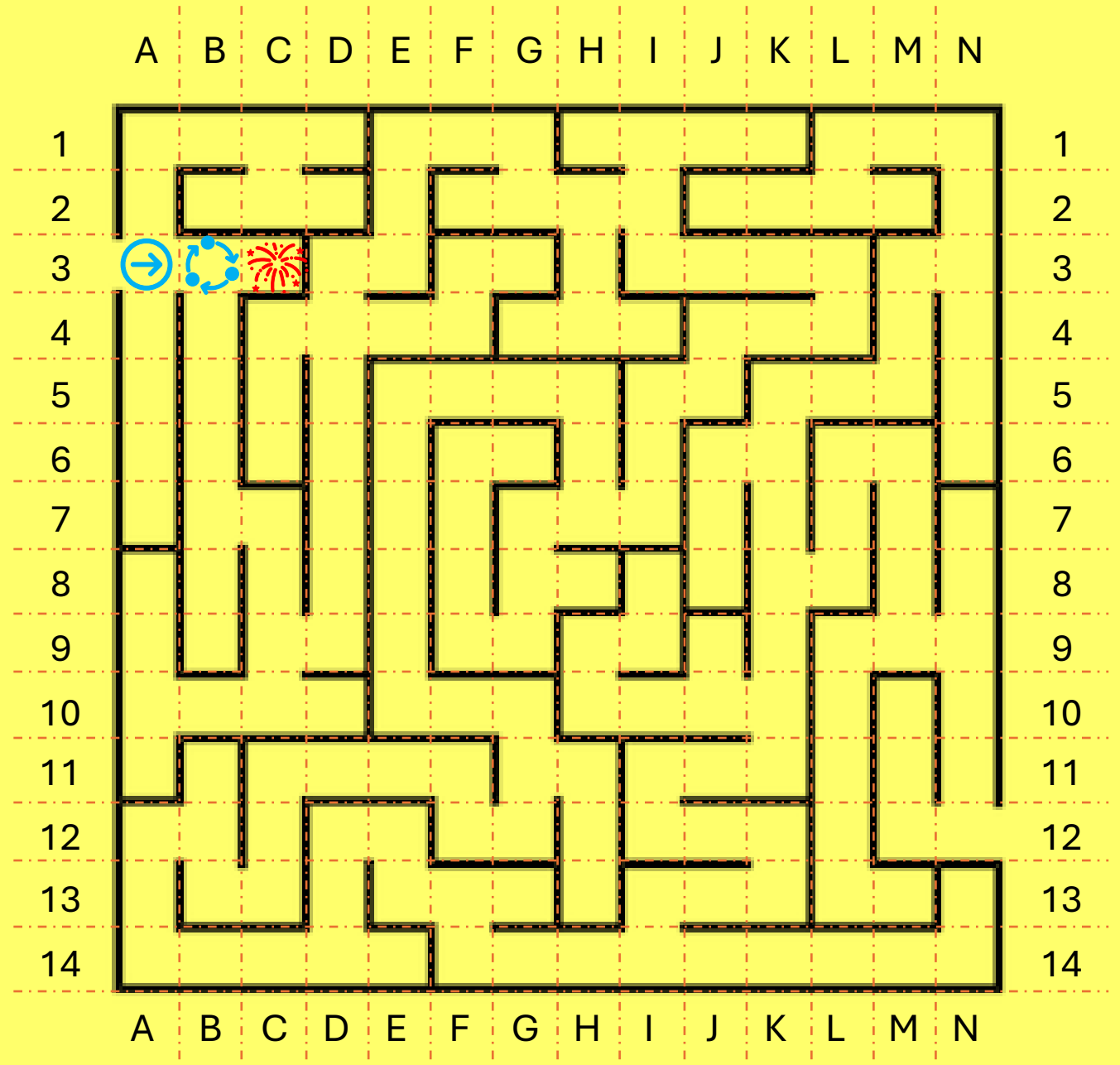
Girar: uma vez, duas vezes, três vezes.



Seguir para: frente, esquerda, direita.



Batendo na parede.



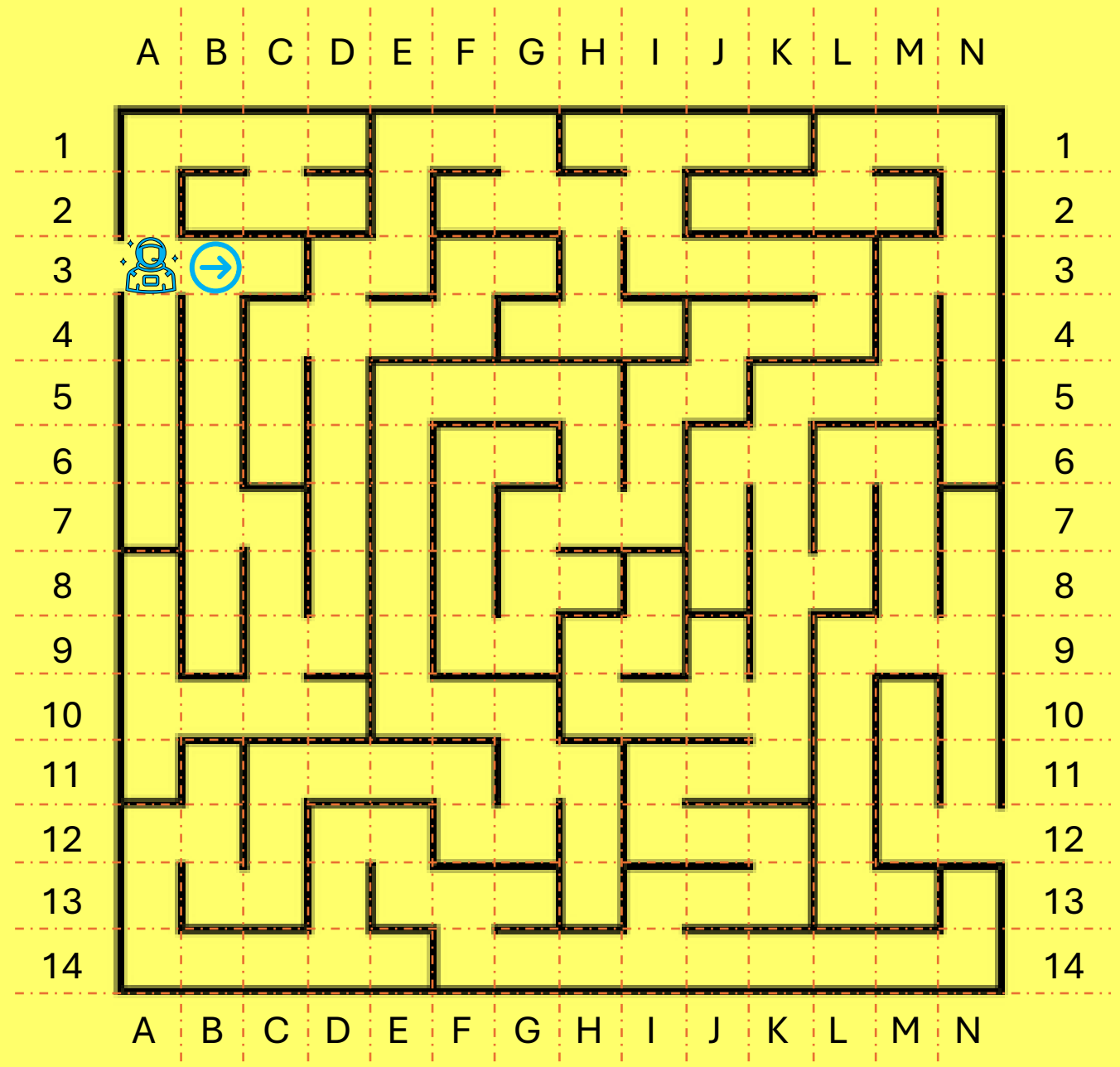
# Exemplo de execução

**def posicao\_atual("A3")** 🚀

A função recebe a informação da posição do Amostradinho.

**def va\_para()** ➡

A função imprime a coordenada para a qual o Amostradinho deve seguir caminho. Por exemplo: B3.

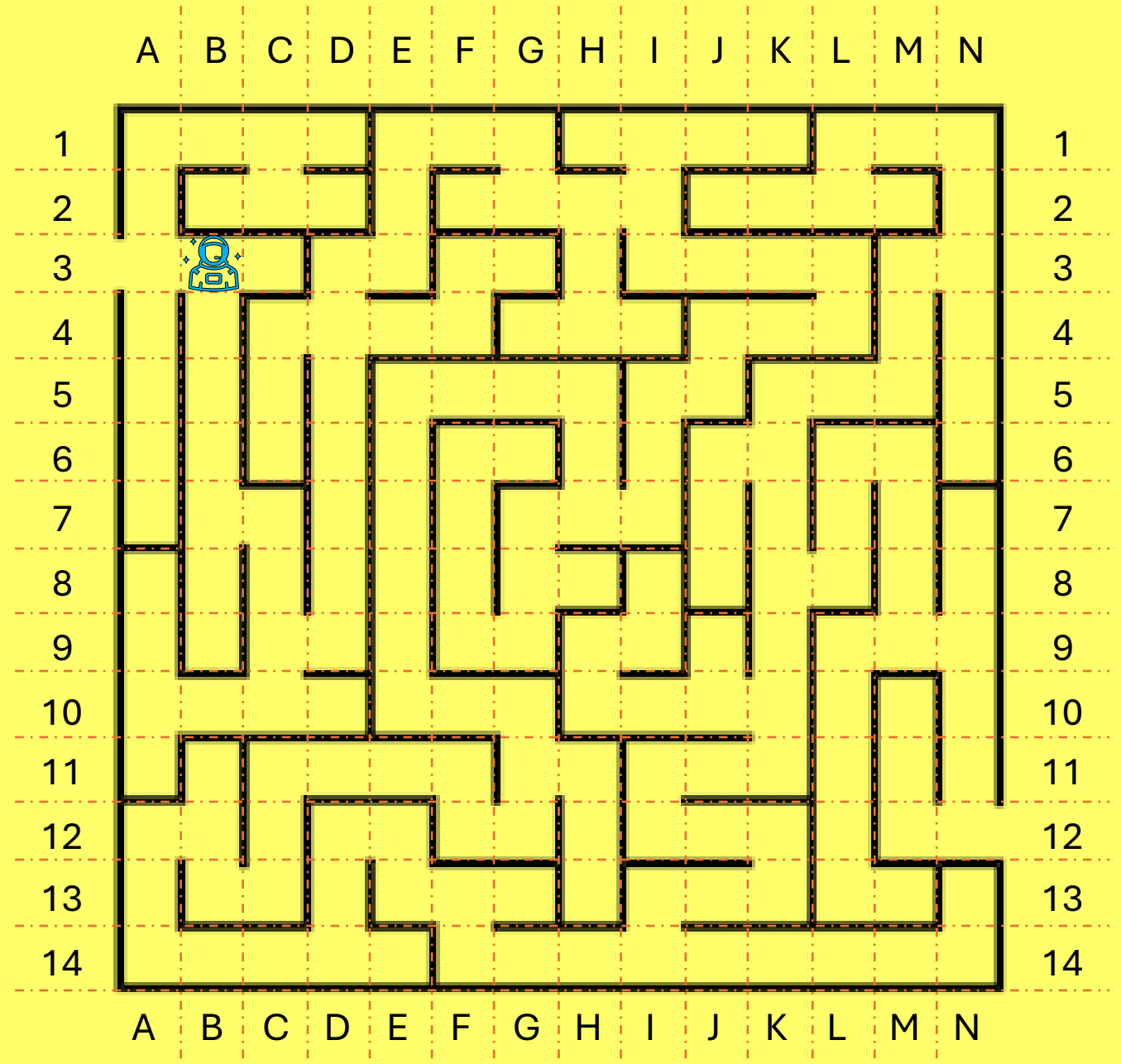


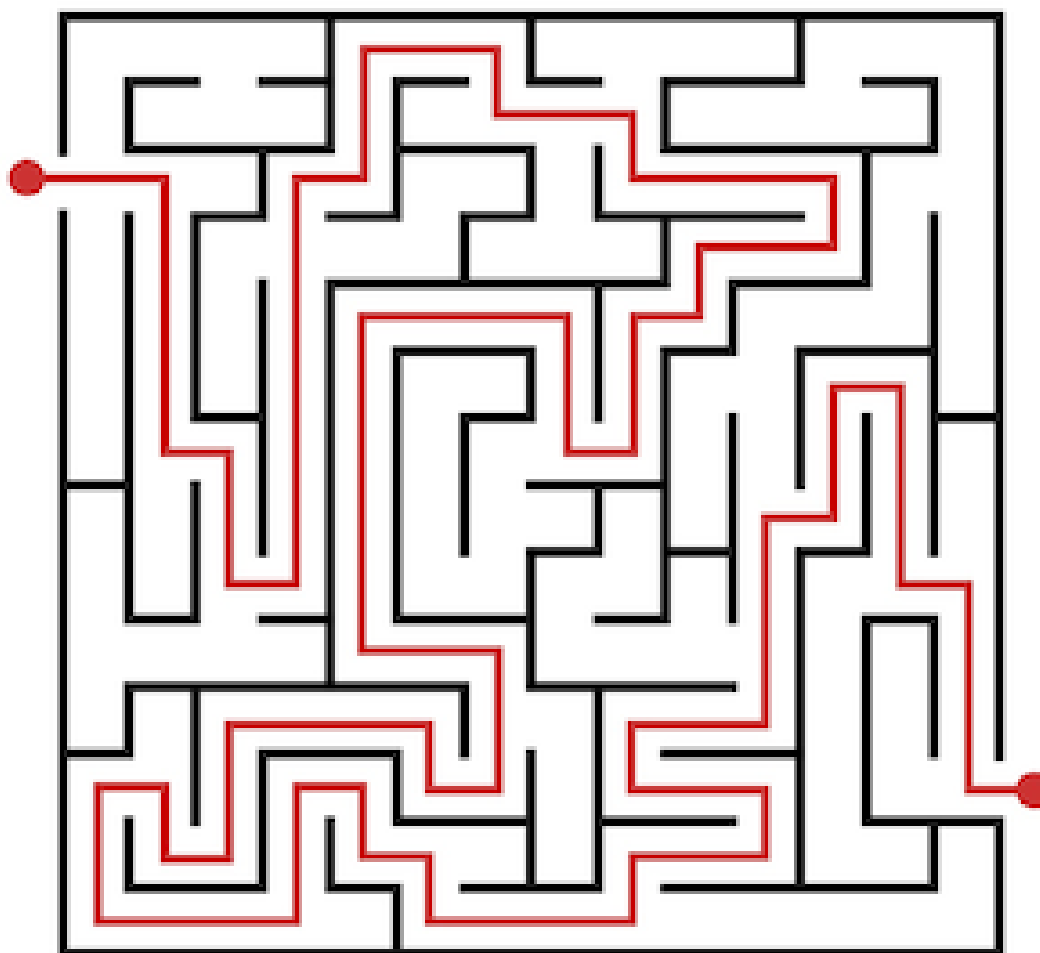
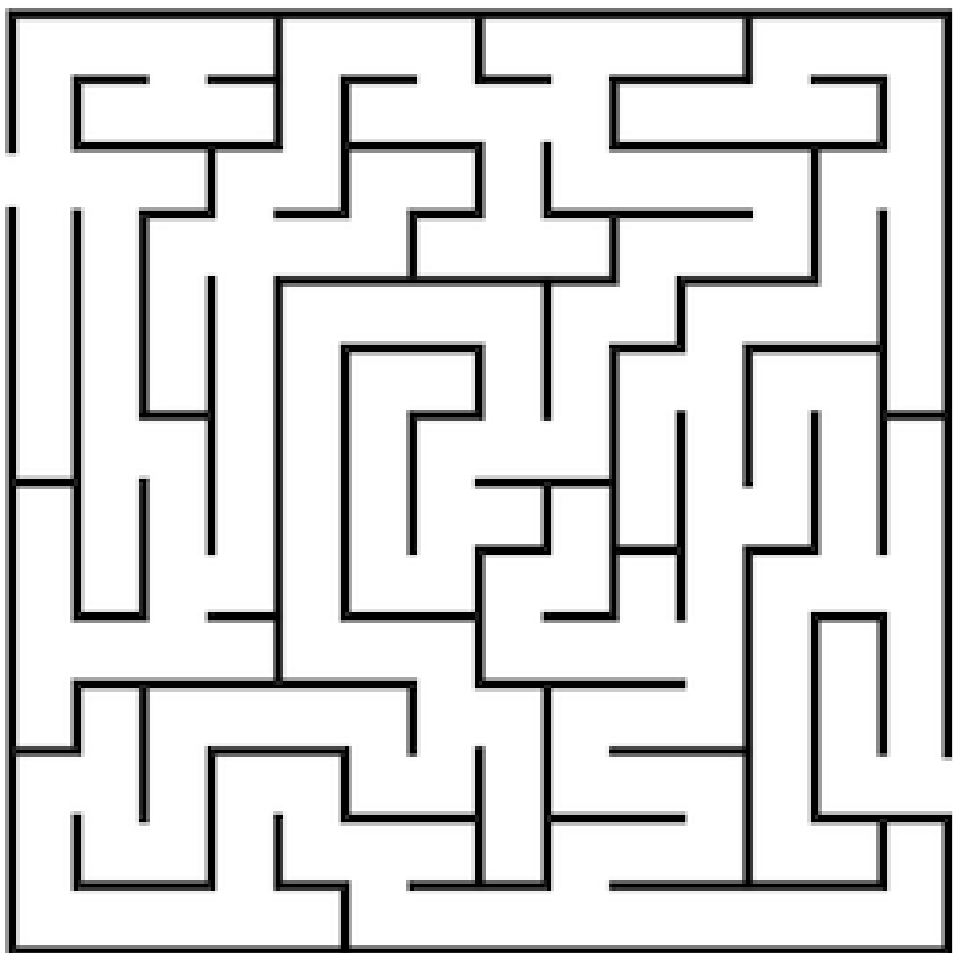


# Exemplo de execução

def va\_para() ➡

A função imprime a coordenada para a qual o Amostradinho deve seguir caminho. Por exemplo: B3.





# Controle do Menu

## Exemplo de como pode ser feito

O código a seguir trata-se de apenas uma sugestão. O estudante tem liberdade para usar a sua criatividade!

```
# importar módulo O.S. para limpeza da tela
import os

# Menu de opções
menu = """
    [0] - Desistir da travessia
    [1] - Posição atual do Amostradinho
    [2] - Caminhar
    [3] - Girar para esquerda
    [4] - Girar para a direita
    """

# laço de repetição infinito
while(True):
    # faz a limpeza da tela
    os.system("cls")

    # imprime o menu de opções
    print(f"\n{menu}\n")

    # captura a resposta do usuário
    resposta = input("Selecione uma opção: ")

    # testa a resposta do usuário
    if resposta == "0":
        # faz a limpeza da tela
        os.system("cls")
        # imprime mensagem de desistência
        print("Desistindo da travessia! :(")
        print() # print vazio para pular uma linha
        # interrompe o loop infinito
        break

    elif resposta == "1":
        print() # print vazio para pular uma linha
        print("Crie a função e faça aqui a invocação para receber a posição atual do Amostradinho!")
        input("") # Input vazio para fazer leitura da mensagem impressa

    elif resposta == "2":
        print() # print vazio para pular uma linha
        print("Crie a função e faça aqui a invocação para Amostradinho caminhar para a nova posição.")
        input("") # Input vazio para fazer leitura da mensagem impressa

    elif resposta == "3":
        print() # print vazio para pular uma linha
        print("Crie a função e faça aqui a invocação para Amostradinho girar para a esquerda.")
        input("") # Input vazio para fazer leitura da mensagem impressa

    elif resposta == "4":
        print() # print vazio para pular uma linha
        print("Crie a função e faça aqui a invocação para Amostradinho girar para a direita.")
        input("") # Input vazio para fazer leitura da mensagem impressa

    else:
        print() # print vazio para pular uma linha
        print("\nOpção inválida")
```