



## Labirinto: aplicações de Pilhas/Filas:

### Objetivo

Colocar em prática os conceitos aprendidos sobre listas encadeadas numa aplicação em Python para gerenciar a lista de espera de candidatos aprovados em concurso público.

### Problema

Considere um labirinto representado por uma matriz  $m \times n$ . O caminho livre é marcado na matriz com " " (sem aspas), enquanto os caminhos ocupados são marcados com "#" (sem aspas).

O ratinho está na posição (livre) (1, 0) e quer chegar à posição (livre) ( $m - 2$ ,  $n - 1$ ) passando pelo labirinto.

Ele pode se mover para uma casa livre que esteja à direita, à esquerda, acima ou abaixo da casa em que está.

### Exemplo de Labirinto

```
#####
# # # # #
#       # # #
#       # # #
# # # # ###
# # ##
#####
```

### Atividade

1. Escreva uma função `eh_possivel_sair()` que retorna True se houver um caminho de saída do labirinto, False caso contrário.

#### Dicas

1. Use pilhas e/ou filas para realizar o processamento do caminho, bem como para guardar o caminho já visitado
2. Marque as casas visitadas
3. Se escolher um caminho sem saída, volte e tente outro caminho (pilha é a melhor solução para voltar)

#### Gerador de labirintos:

- Utilize o código disponível em <https://replit.com/@ricardorubens/gera-lab-mp2> para gerar um novo labirinto de teste .
- Para usar o código acima, salve o código em um arquivo, por exemplo, `gera_lab.py`:

```
from gera_lab import gera_lab, print_lab #importa as funções do módulo gera_lab

lab = gera_lab(7,14) #gera um labirinto com 7 linhas e 14 colunas
print_lab(lab) #imprime o lab
```

#### Atividade - Teste 1

```
def eh_possivel_sair(lab):
    #seu codigo aqui
    return False

#Teste 1
labirinto1 = [
    ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
    [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '#'],
    ['#', ' ', ' ', ' ', ' ', '#', '#', ' ', ' ', '#'],
    ['#', '#', '#', ' ', ' ', '#', ' ', ' ', '#', '#'],
    ['#', '#', ' ', '#', ' ', ' ', ' ', ' ', ' ', ' '],
    ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#']]

resultado_esperado1 = True
resultado1 = eh_possivel_sair(labirinto1)

print(resultado1 == resultado_esperado1) #deve ser True
```

## Atividade - Teste 2

```
def eh_possivel_sair(lab):
    #seu codigo aqui
    return False

#Teste 1
labirinto2 = [['#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
               [' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', '#'],
               ['#', '#', ' ', ' ', ' ', ' ', '#', '#', ' ', '#'],
               ['#', ' ', ' ', '#', '#', '#', '#', ' ', '#', ' '],
               ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#']]

resultado_esperado1 = False
resultado1 = eh_possivel_sair(labirinto2)

print(resultado1 == resultado_esperado1) #deve ser True
```

## Realização e Entrega

- Pode ser realizado *individualmente* ou em equipe de até 3 estudantes
- Conteúdo a ser entregue:
  - arquivos do código fonte ou um link (github ou replit)
  - arquivo leia.me.txt contendo:
    - integrantes da equipe (pode estar no link)
    - lista com todo o conteúdo de consultado para realização do projeto (sites, livros, artigos, códigos prontos, etc.)
    - comentário da equipe sobre se conseguiu ou não realizar tudo o que foi proposto.
    - comentário destacando possíveis problemas identificados no código, dificuldades encontradas na escrita e/ou funcionalidades que deveriam ser implementadas, mas não foram.
- Apresentar o projeto funcionando ao professor

## Critérios de Avaliação

1. Executar sem erros
2. Apresentar a função `eh_possivel_sair()` de acordo com o que foi solicitado

## Para estudar

- Pilhas
  - Seções 3.1 a 3.9 do livro [https://panda.ime.usp.br/panda/static/pythonds\\_pt/index.html](https://panda.ime.usp.br/panda/static/pythonds_pt/index.html)
  - Capítulo sobre pilha em qualquer livro sobre estruturas de dados
- Filas

- Seções 3.10 a 3.14 do livro [https://panda.ime.usp.br/panda/static/pythonds\\_pt/index.html](https://panda.ime.usp.br/panda/static/pythonds_pt/index.html)
- Capítulo sobre fila em qualquer livro sobre estruturas de dados