

# LISTA DE EXERCÍCIOS - ESSENCIAL

## PILHAS, FILAS e ÁRVORES

As questões resolvidas em sala estão no link:

<https://replit.com/@AlissonGomes/ListaDeExercicios-Estrutura-de-Dados#main.py>

Exemplo de recursividade: <https://replit.com/@AlissonGomes/Fatorial-Recursividade-While-e-For#main.py:27>

## PARTE 1 - PILHAS E FILAS

### Questões elementares:

- 1) Implemente uma classe ou struct que represente a estrutura de dados (Pilha 1x, Fila 1x), com o elemento atual (data) e o próximo elemento (next) - modelo Lista Encadeada
  - OPCIONAL:
    - Popular a estrutura de dados em Classe a partir de um array
    - Imprimir a estrutura de dados da Classe
- 2) Implemente uma função que recebe um elemento como parâmetro e o adiciona na estrutura de dados: pushWithClassLinkedList(e), pushWithArray(e), insertWithClassLinkedList(e) e insertWithArray(e)
- 3) Implemente uma função que deleta e retorna o elemento padrão, de acordo com a estrutura de dados: popWithClassLinkedList(e), popWithArray(e), removeWithClassLinkedList(e) e removeWithArray(e)

**CADA QUESTÃO ABAIXO POSSUI OS OS SEGUINTE ITENS, LOGO CRIE FUNÇÕES COM OS SUFIXOS DE CADA ITEM EM CADA QUESTÃO:**

- A) Implementação elementar utilizando Pilhas:** utilizando apenas as funções push e pop e criando outras pilhas quando necessário, sem utilizar nenhuma biblioteca
- B) Implementação diversa utilizando Pilhas:** utilizando quaisquer lógicas para resolver o problema
- C) Implementação elementar utilizando Filas:** utilizando apenas as funções insert e remove e criando outras filas quando necessário, sem utilizar nenhuma biblioteca
- D) Implementação diversa utilizando Filas:** utilizando quaisquer lógicas para resolver o problema

**Observação:**

- A implementação elementar de cada estrutura de dados deve utilizar, preferivelmente, uma classe ou struct Pilha ou Fila com o elemento atual (data) e o próximo elemento (next), ou, quem desejar tornar a implementação mais fácil e simples, pode representar a estrutura de dados através de um array
- Caso a implementação seja igual a diversa, deve-se reimplementar a elementar utilizando classes ou structs

### Questões elementares e diversas:

- 4) Implemente uma função que retorna o primeiro elemento, conforme a estrutura de dados (peek 2x e first 2x), sem alterar os dados ao final da execução
- 5) Implemente uma função que retorna se a estrutura de dados está vazia ou não (isEmpty 4x), sem alterar os dados ao final da execução
- 6) Implemente uma função que retorna a quantidade de elementos (len 4x), conforme a estrutura de dados, sem alterar os dados ao final da execução
- 7) Implemente uma função que retorna o último elementos (last 4x), conforme a estrutura de dados, sem alterar os dados ao final da execução
- 8) Implemente uma função que retorna o elemento em um determinado índice (getValueByIndex(i) 4x), conforme a estrutura de dados, sem alterar os dados ao final da execução
- 9) Implemente uma função que retorna o primeiro index de determinado valor de elemento (getIndexByValue(e) 4x), conforme a estrutura de dados, sem alterar os dados ao final da execução
- 10) Implemente uma função que retorna um array de indexs de determinado valor de elemento (getAllIndexByValue(e) 4x), conforme a estrutura de dados, sem alterar os dados ao final da execução
- 11) Implemente uma função que retorna um array de valores dos elementos conforme uma lista de índices (getValuesByIndexes(indexes) 4x), conforme a estrutura de dados, sem alterar os dados ao final da execução
- 12) Implemente uma função que retorna um array de valores dos elementos de um índice inicial até um índice final (getValuesBySlice(ii, if) 4x), conforme a estrutura de dados, sem alterar os dados ao final da execução
- 13) Implemente uma função que remove todos os elementos (removeAll 4x), conforme a estrutura de dados
- 14) Implemente uma função que remove um elemento em determinado índice (removeByIndex(i) 4x), conforme a estrutura de dados
- 15) Implemente uma função que remove o primeiro elemento com determinado valor (removeByValue(e) 4x), conforme a estrutura de dados
- 16) Implemente uma função que remove todos os elementos com determinado valor (removeAllByValue(e) 4x), conforme a estrutura de dados
- 17) Implemente uma função que remove todos os elementos no array de índices (removeAllByIndexes(indexes) 4x), conforme a estrutura de dados
- 18) Implemente uma função que remove todos os elementos do índice inicial até o índice final (removeAllBySlice(ii, if) 4x), conforme a estrutura de dados
- 19) Implemente uma função que altera o valor de um elemento em determinado índice (setValueInIndex(i, e) 4x), conforme a estrutura de dados
- 20) Implemente uma função que altera uma lista de valores na ordem de uma lista de índices (setValuesInIndexes(indexes, elements) 4x), conforme a estrutura de dados

## PARTE 2 - ÁRVORES BINÁRIAS - apenas uma implementação

21) Implemente uma classe ou struct que represente uma árvore binária, com o nó atual, o nó à esquerda e o nó direita

OPCIONAL:

- Popular a árvore em Classe a partir de um array em nível

22) Implemente uma função que retorna um array da árvore binária em pré-ordem

23) Implemente uma função que retorna um array da árvore binária em in-ordem

24) Implemente uma função que retorna um array da árvore binária em pós-ordem

25) Implemente uma função que retorna um array da árvore binária em nível

26) Implemente uma função que retorna a altura da árvore binária

27) Implemente uma função que retorna a quantidade de elementos da árvore binária

28) Implemente uma função que retorna as folhas da árvore binária

29) Implemente uma função que retorna se a árvore binária é completa ou não

30) Implemente uma função que retorna se a árvore binária é estritamente binária ou não