

# Exercícios de Implementação: Pilhas e Filas

**Disciplina:** Estrutura de Dados

**Foco:** Conceitos Fundamentais de Pilhas e Filas

---



## Parte 1 — Exercícios com Pilhas (Stack)

O princípio da pilha é: o **último a entrar é o primeiro a sair** (*LIFO – Last In, First Out*).

Pense em uma pilha de pratos para visualizar o conceito.

---

### 1. Inverter uma Palavra

Crie uma função que receba uma string (palavra) e a retorne invertida.

Para isso, insira cada caractere da palavra em uma pilha e, em seguida, remova-os um a um para formar a nova palavra.

- **Entrada:** "programar"
  - **Saída:** "ramargorp"
- 

### 2. Verificar se uma Pilha está Vazia

Implemente uma função que cria uma pilha, adiciona alguns números (10, 20, 30) e depois os remove um a um.

A cada remoção, verifique e imprima se a pilha está vazia.

**Saída Esperada (no console):**

```
Removeu 30. A pilha está vazia? false
Removeu 20. A pilha está vazia? false
Removeu 10. A pilha está vazia? true
```

---

### 3. Acessar o Elemento do Topo

Crie uma função que adicione [5, 8, 12] a uma pilha.

Em seguida, use a operação adequada para “espiar” o elemento do topo **sem removê-lo**. Imprima o elemento, remova-o e espie o novo topo.

**Saída Esperada (no console):**

Elemento do topo: 12

Removendo o topo...

Novo elemento do topo: 8

---

## 4. Somar os Elementos de uma Pilha

Implemente uma função que receba uma pilha de números e retorne a soma de todos os seus elementos.

A pilha original ficará vazia ao final do processo.

- **Entrada:** Pilha com [4, 2, 7]
  - **Saída:** 13
- 

## 5. Jogo de Remover Pares

Crie uma função que receba uma pilha de números.

A regra é: se os dois números do topo forem iguais, remova ambos.

Repita o processo até que não seja mais possível remover pares.

Retorne a quantidade de elementos que sobraram.

- **Entrada:** Pilha com [1, 2, 2, 3, 3, 3, 1]
  - **Saída:** 3 (Sobraram [1, 3, 1])
- 



## Parte 2 — Exercícios com Filas (Queue)

O princípio da fila é: o **primeiro a entrar é o primeiro a sair (FIFO – First In, First Out)**.

Pense em uma fila de supermercado.

---

## 6. Simular Fila de Atendimento

Crie um programa que simule uma fila de atendimento.

Adicione os nomes "**Ana**", "**Carlos**" e "**Beatriz**" a uma fila.

Em seguida, simule o atendimento removendo um por um e imprimindo o nome de quem está sendo atendido.

**Saída Esperada (no console):**

Atendendo: Ana

Atendendo: Carlos

Atendendo: Beatriz

Fila vazia.

---

## 7. Ver o Próximo da Fila

Implemente uma função que adicione [ "**José**", "**Maria**", "**Pedro**" ] a uma fila.

Use uma operação para verificar quem é o primeiro da fila sem removê-lo.

Em seguida, atenda essa pessoa (remova-a) e verifique quem se tornou o novo primeiro.

**Saída Esperada (no console):**

Próximo a ser atendido: José

Atendendo José...

Agora, o próximo da fila é: Maria

---

## 8. Contar Elementos na Fila

Crie uma função que receba uma fila e retorne o número de elementos presentes nela **sem alterar a fila original**.

- **Entrada:** Fila com [ "a", "b", "c", "d" ]
  - **Saída:** 4
- 

## 9. Fila de Impressão

Simule uma fila de impressão.

Adicione **4 documentos** à fila.

Em seguida, imprima (remova) os dois primeiros e adicione um novo documento ao final.

Ao final, mostre os documentos restantes na fila.

**Passos:**

1. Adicionar: "Doc1.pdf", "Foto.png", "Trab.docx", "Planilha.xlsx"
2. Remover 2.
3. Adicionar: "Email.pdf"

**Saída Esperada (no console):**

Imprimindo: Doc1.pdf

Imprimindo: Foto.png

Documentos restantes na fila: Trab.docx, Planilha.xlsx, Email.pdf

---

## 10. Inverter os Primeiros K Elementos de uma Fila

Crie uma função que receba uma fila e um número **k**.

Sua tarefa é inverter a ordem dos **k** primeiros elementos da fila, mantendo o restante dos elementos em sua ordem original.

 *Dica: uma pilha pode ajudar aqui!*

- **Entrada:** Fila [10, 20, 30, 40, 50] e **k** = 3
- **Saída:** Fila [30, 20, 10, 40, 50]