

## Aula 3: Estruturas de Dados

### Pilhas e Filas

Olá novamente, queridos aprendizes! Na nossa jornada pelas estruturas de dados, chegamos à Aula 3, onde vamos explorar duas estruturas muito interessantes: as "Pilhas" e as "Filas". Vamos descobrir o que são, como funcionam e onde podem ser aplicadas. Além disso, aprenderemos a implementar pilhas e filas em programação. Vamos começar!

#### Pilhas: Operações e Aplicações

Imagine uma pilha de pratos em um restaurante. Você sempre pega o prato que está no topo da pilha, certo? Isso é exatamente como uma "pilha" funciona em programação. Uma pilha é uma coleção de elementos onde o último elemento adicionado é o primeiro a ser removido. As operações básicas em uma pilha são "push" (empurrar) para adicionar um elemento e "pop" (retirar) para remover o elemento do topo.

As pilhas são úteis em situações onde a ordem de processamento é importante, como a função "desfazer" em editores de texto ou a verificação de parênteses em expressões matemáticas.

#### Filas: Operações e Aplicações

Agora, pense em uma fila de pessoas esperando em um caixa de supermercado. A primeira pessoa que entra na fila é a primeira a ser atendida. Isso é o que uma "fila" representa em programação.

Uma fila é uma coleção de elementos onde o primeiro elemento adicionado é o primeiro a ser removido. As operações básicas em uma fila são "enqueue" (enfileirar) para adicionar um elemento e "dequeue" (desenfileirar) para remover o elemento do início.

As filas são úteis em situações em que a ordem de chegada é importante, como em sistemas de gerenciamento de tarefas ou na impressão de documentos em uma impressora compartilhada.

#### Implementação de Pilhas e Filas em Programação

Vamos dar uma olhada em como podemos implementar pilhas e filas em Python:

### Exemplo de Implementação de Pilha em Python:

```
python Copy code

class Pilha:
    def __init__(self):
        self.itens = []

    def push(self, item):
        self.itens.append(item)

    def pop(self):
        if not self.esta_vazia():
            return self.itens.pop()
        else:
            return None

    def esta_vazia(self):
        return len(self.itens) == 0
```

### Exemplo de Implementação de Fila em Python:

```
python Copy code

from collections import deque

class Fila:
    def __init__(self):
        self.itens = deque()

    def enqueue(self, item):
        self.itens.append(item)

    def dequeue(self):
        if not self.esta_vazia():
            return self.itens.popleft()
        else:
            return None

    def esta_vazia(self):
        return len(self.itens) == 0
```

Esses são apenas exemplos simples de implementação de pilhas e filas em Python. Elas são estruturas de dados poderosas usadas em muitas aplicações da vida real.