

Carlos Eduardo Silva de Oliveira

Pesquisa - Estrutura de Dados (Pilha e Fila)

Joinville
2023

1. Explicação sobre Estrutura de Dados:

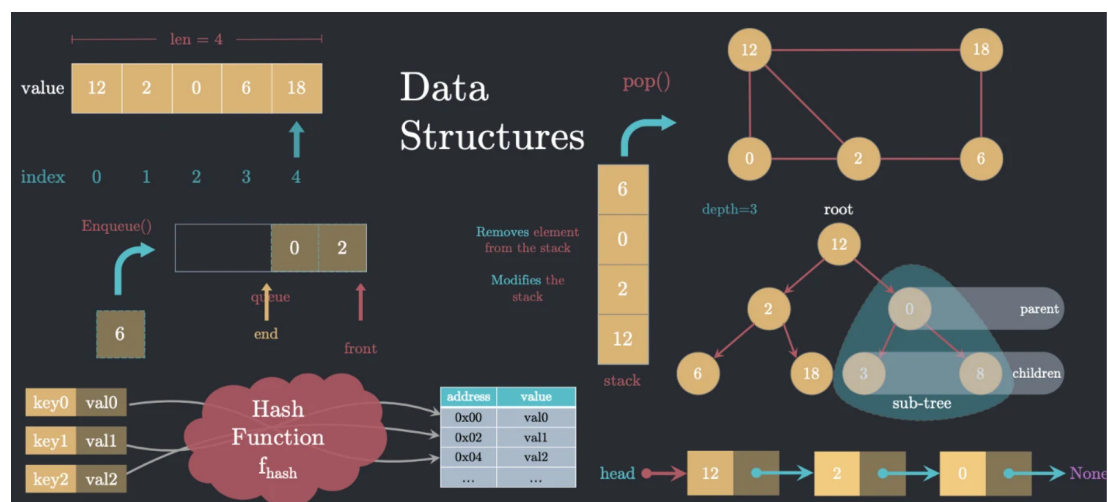
A Estrutura de Dados é uma disciplina fundamental na programação e no desenvolvimento de software. Em termos simples, refere-se à organização e armazenamento de dados de forma eficiente para facilitar o acesso e manipulação. Uma estrutura de dados adequada é essencial para otimizar algoritmos, tornando o software mais rápido e eficiente. Essa disciplina permite aos programadores escolher a melhor maneira de organizar e manipular dados com base nos requisitos específicos de cada problema.

As estruturas de dados podem ser:

- lineares (ex. arrays) ou não lineares (ex. grafos);
- homogêneas (todos os dados que compõe a estrutura são do mesmo tipo) ou heterogêneas (podem conter dados de vários tipos);
- estáticas (têm tamanho/capacidade de memória fixa) ou dinâmicas (podem expandir).

Exemplo de estruturas:

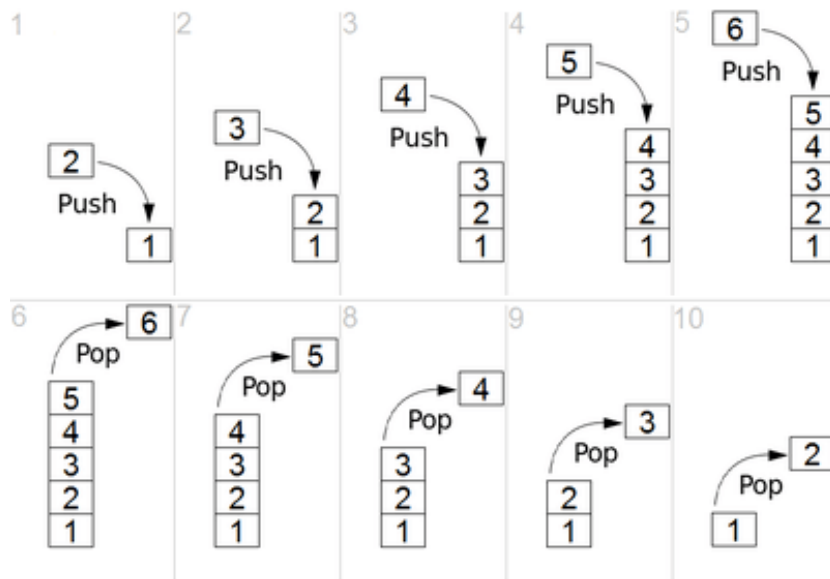
- Array;
- Pilha;
- Fila;
- Deque;
- Fila circular;
- Lista ligada;
- Conjunto;
- Dicionário ou hashmap;
- Árvore;
- Heap binário;
- Grafo.



Fonte: https://www.fiverr.com/toms_rozkalns
[Estruturas de dados: uma introdução | Alura](#)

2. Pilhas em Estrutura de Dados:

As pilhas são estruturas de dados lineares que seguem o princípio LIFO (Last In, First Out), onde o último elemento inserido é o primeiro a ser removido. Suas operações fundamentais incluem a inserção de elementos no topo (push) e a remoção do topo (pop). Pilhas são frequentemente utilizadas em situações em que a ordem inversa de processamento é necessária, como na reversão de strings ou no rastreamento de chamadas de funções.



Fonte: [Estruturas de dados: Pilha](#)

Exemplo de uso de pilha em Notação Polonesa Inversa

Por exemplo: $((1 + 2) * 4) + 3$ em notação pós-fixa:

1 2 + 4 * 3 +

[Algoritmos e Estruturas de Dados/Pilhas - Wikilivros](#)

Utilizando uma pilha consideramos:

empilhar quando encontrar um operando;

sacar dois operandos e achar o valor quando encontrar uma operação.

empilhar o resultado.

Entrada	Operação	Pilha
1	push operando	1
2	push operando	1, 2
+	adicionar	3
4	Push operando	3, 4
*	multiplicar	12
3	push operando	12, 3
+	adicionar	15

O resultado final 15 estará no topo da pilha no fim do cálculo.

[O que é e como funciona a Estrutura de Dados Pilha | Blog TreinaWeb](#)

3. Exemplos de Aplicações Reais de Pilhas:

- Navegação na Web:

Ao pressionar o botão "Voltar" em um navegador, as páginas visitadas são gerenciadas por uma pilha. Cada nova página visitada é empilhada, e ao pressionar "Voltar", a página é desempilhada, proporcionando uma navegação intuitiva.

- Undo em Editores de Texto:

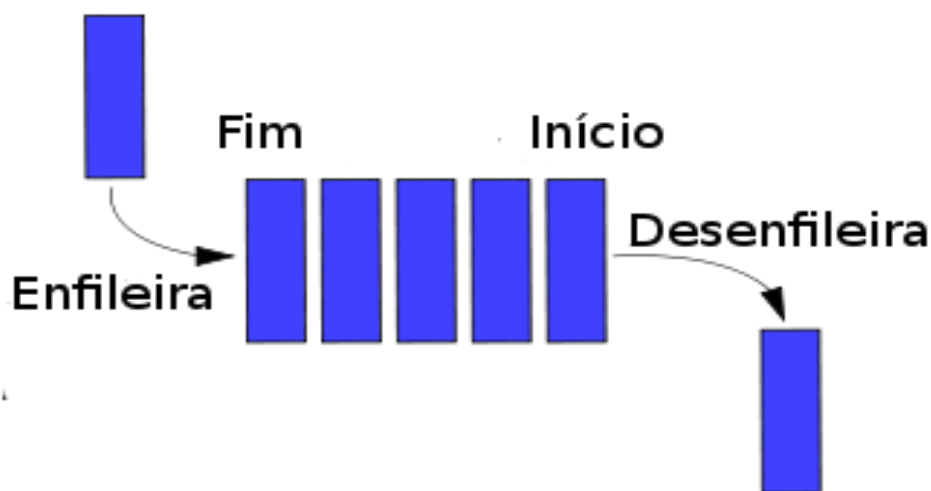
Editores de texto frequentemente utilizam pilhas para implementar a funcionalidade "Desfazer" (Undo). Cada ação realizada é empilhada, permitindo que o usuário reverta as ações na ordem inversa em que foram feitas.

[Estrutura de Dados e Algoritmos](#)

4. Filas em Estrutura de Dados:

Filas são estruturas de dados lineares que seguem o princípio FIFO (First In, First Out), onde o primeiro elemento inserido é o primeiro a ser removido. Operações fundamentais incluem a inserção no final (enqueue) e a remoção do início (dequeue). Filas são comumente usadas em situações onde a ordem de chegada é crucial, como em sistemas de processamento de pedidos.

[O que é e como funciona a Estrutura de Dados Fila | Blog TreinaWeb.](#)



Fonte: <https://wiki.sj.ifsc.edu.br/images/a/ad/Prg-queue.png>

Operações com Fila:

Todas as operações em uma fila podem ser imaginadas como as que ocorre numa fila de pessoas num banco, exceto que o elementos não se movem na fila, conforme o primeiro elemento é retirado. Isto seria muito custoso para o computador. O que se faz na realidade é indicar quem é o primeiro.

- criação da fila (informar a capacidade no caso de implementação sequencial - vetor);
- enfileirar (enqueue) - o elemento é o parâmetro nesta operação;
- desenfileirar (dequeue);
- mostrar a fila (todos os elementos);
- verificar se a fila está vazia (isEmpty);
- verificar se a fila está cheia (isFull - implementação sequencial - vetor).

[Estrutura de Dados e Algoritmos](#)

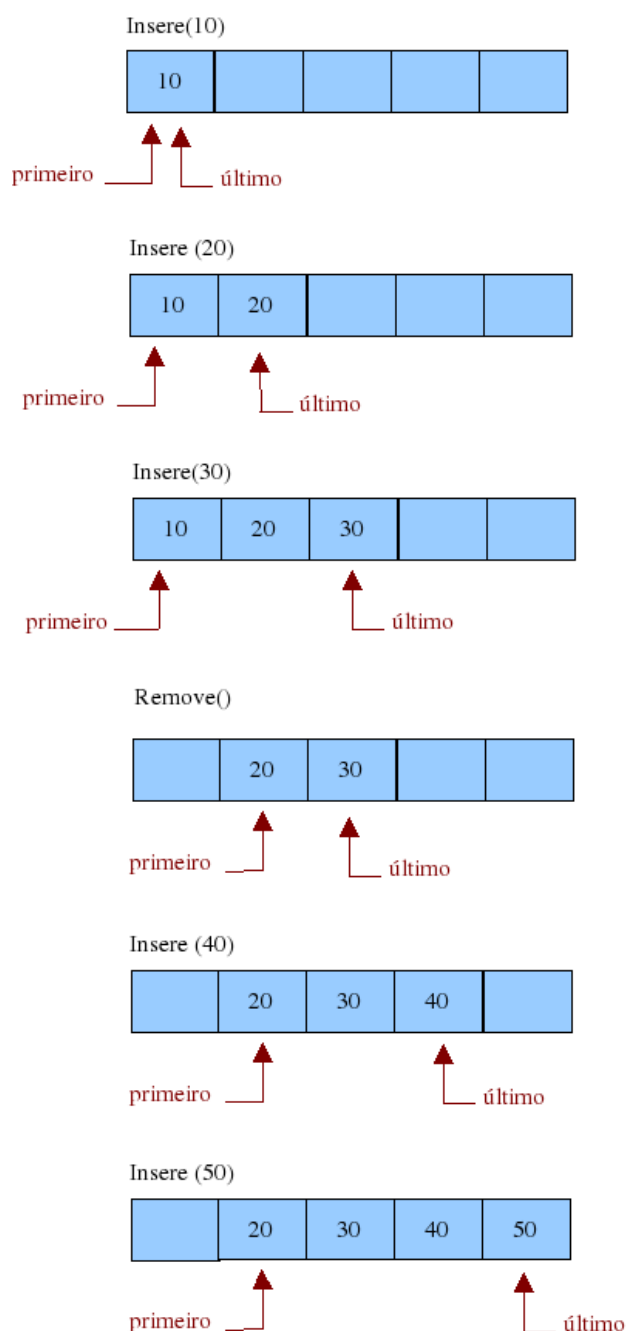
5. Exemplos de Aplicações Reais de Filas:

- Fila em um Supermercado:

Os clientes que chegam primeiro são atendidos primeiro. A estrutura de fila garante uma distribuição justa e organizada do atendimento.

- Fila/Spool de Impressão:

Sistemas de spooling de impressão utilizam filas para organizar os trabalhos de impressão. Os documentos são enfileirados e processados conforme a ordem de chegada, esta função diminui o tempo de impressão, maximizando a eficácia da impressora.



Fonte: <https://www.cos.ufrj.br/~rfarias/cos121/operacoesFila1.png>

6. Importância da Escolha da Estrutura de Dados:

Escolher a estrutura de dados correta é crucial para o desempenho eficiente de algoritmos. Uma escolha inadequada pode resultar em perda de tempo e recursos. Por exemplo, o uso de uma fila em vez de uma pilha em um contexto que requer ordem inversa pode levar a uma implementação menos eficiente. A compreensão das características e aplicações de cada estrutura de dados é essencial para tomar decisões mais eficientes.

[Introdução Completa À Estrutura de Dados: Conceitos e Aplicações.](#)

Exemplo onde foi prejudicial não utilizar estrutura de dados:

Sistema de controle de estoque:

Em uma loja que não utiliza estrutura de dados. Nesse caso, se cada item do estoque fosse tratado como uma variável independente, sem uma estrutura organizada, poderia resultar em uma bagunça difícil de gerenciar.

Sem uma estrutura de dados, teria que lidar com variáveis individuais para cada produto, como `quantidade_produto_1`, `quantidade_produto_2`, e assim por diante. Isso tornaria as operações de atualização, busca e remoção de itens do estoque complicadas e propensas a erros.

Com o uso de uma estrutura de dados, como um array, lista ou banco de dados, seria mais fácil organizar e manipular as informações do estoque de forma eficiente. A estrutura adequada permitiria a adição e remoção de produtos de maneira mais intuitiva, facilitando a manutenção do estoque e reduzindo a probabilidade de erros.

Utilizando estrutura:

- Evita duplicidade de dados, gerenciamento de múltiplas cópias dos dados, possibilidade de existirem dados conflitantes;
- Cada alteração na estrutura de dados implica em menos modificações, e menor chance de erros de desenvolvimento;
- Facilita a manutenção geral do código, reduzindo o tempo e o custo de manutenção.

Não utilizando estrutura:

- Dificuldade de manutenção;
- Execução de tarefas repetidas;
- Causando efeito totalmente contrário às vantagens de utilizar a estrutura.

[A Importância das Estruturas de Dados - Lab C++ - Codificação Eficiente](#)

7. Referências:

[Estruturas de dados: uma introdução | Alura](#)

[Estruturas de dados: Pilha](#)

[Algoritmos e Estruturas de Dados/Pilhas - Wikilivros](#)

[O que é e como funciona a Estrutura de Dados Pilha | Blog TreinaWeb](#)

[Estrutura de Dados e Algoritmos - Pilha](#)

[O que é e como funciona a Estrutura de Dados Fila | Blog TreinaWeb.](#)

[Estrutura de Dados e Algoritmos - Fila](#)

[Introdução Completa À Estrutura de Dados: Conceitos e Aplicações.](#)

[A Importância das Estruturas de Dados - Lab C++ - Codificação Eficiente](#)