

Pilhas (Stack) - Guia Didático

O que é uma Pilha?

Pense em uma **pilha de pratos** na sua cozinha. Você sempre:

- Coloca um novo prato **em cima** (no topo)
- Pega um prato para usar **sempre do topo**

Essa é a essência da pilha: **LIFO** - Last-In, First-Out (Último a Entrar, Primeiro a Sair).

Operações Básicas

1. Push() - Empilhar

- **Ação**: Adiciona um elemento no topo da pilha
- **Analogia**: Colocar mais um prato na pilha
- **Exemplo**:
 - Pilha: [A, B, C]
 - Push(D) → [A, B, C, D]

2. Pop() - Desempilhar

- **Ação**: Remove e retorna o elemento do topo
- **Analogia**: Pegar o prato do topo para usar
- **Exemplo**:
 - Pilha: [A, B, C, D]
 - Pop() → Retorna D, Pilha: [A, B, C]

3. Peek() - Espiar

- **Ação**: Mostra o elemento do topo sem remover
- **Analogia**: Olhar qual prato está no topo sem pegá-lo
- **Exemplo**:
 - Pilha: [A, B, C, D]
 - Peek() → Retorna D, Pilha continua: [A, B, C, D]

4. isEmpty() - Está Vazia?

- **Ação**: Verifica se a pilha não tem elementos
- **Retorno**: True (vazia) ou False (com elementos)

Como Implementar?

Com Array (Vetor)

- **Vantagem**: Acesso rápido e simples
- **Desvantagem**: Tamanho fixo (pode estourar)
- **Uso**: Quando sabemos o tamanho máximo

Com Lista Encadeada

- **Vantagem**: Tamanho ilimitado e flexível
- **Desvantagem**: Usa mais memória (ponteiros)
- **Uso**: Quando não sabemos quantos elementos teremos

Complexidade - Eficiência

Todas as operações principais são **$O(1)$** - tempo constante:

- **Push**: Sempre adiciona no topo - rápido
- **Pop**: Sempre remove do topo - rápido
- **Peek**: Sempre olha o topo - rápido

Aplicações no Mundo Real

1. Histórico UNDO/REDO

- **Como funciona**: Cada ação é "empilhada"
- **UNDO**: Pop() - desfaz a última ação
- **REDO**: Push() - refaz a ação

2. Chamada de Funções (Call Stack)

- **Como funciona**: Quando uma função chama outra:
 - Empilha a função atual
 - Executa a nova função
 - Quando termina, desempilha e volta para a anterior

3. Navegador Web - Voltar/Avançar

- **Como funciona**: Cada página visitada vai para a pilha
- "Voltar": Pop() - volta para página anterior
- "Avançar": Push() - vai para próxima página

4. Verificação de Expressões Matemáticas

- **Como funciona**: Para verificar parênteses balanceados:
 - Abre parênteses: Push()
 - Fecha parênteses: Pop()
 - No final, pilha deve estar vazia

5. Calculadora RPN (Notação Polonesa Reversa)

- **Como funciona**:
 - Números: Push()
 - Operações: Pop() dos operandos, calcula, Push() do resultado

Exemplo Prático - Verificação de Parênteses

Expressão: (() ())

Passo a passo:

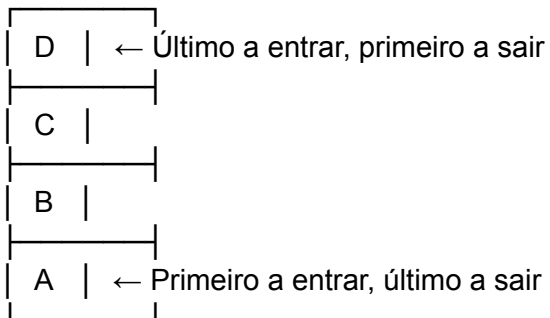
1. '(' → Push → Pilha: ['(']
2. '(' → Push → Pilha: ['(', '(']
3. ')' → Pop → Pilha: ['('] (removeu um '(')
4. '(' → Push → Pilha: ['(', '(']
5. ')' → Pop → Pilha: ['('] (removeu um '(')
6. ')' → Pop → Pilha: [] (removeu o último '(')

****Resultado****: Válido! Pilha ficou vazia.

Resumo Visual

...

TOPO



BASE

...

****Regra de Ouro****: Só trabalhamos com o topo! Não acessamos elementos do meio diretamente.

Pilhas são perfeitas para situações onde a ordem de processamento é "o último a chegar é o primeiro a ser atendido"!

Pergunta 1: Princípio Fundamental

Qual princípio define o funcionamento de uma pilha (stack)?

- a) FIFO (First-In-First-Out)
- b) LIFO (Last-In-First-Out)
- c) Ordenação por prioridade
- d) Acesso aleatório

Resposta: b) LIFO (Last-In-First-Out)

Pergunta 2: Aplicação Prática

Em qual dos seguintes cenários uma pilha seria a estrutura de dados MAIS adequada?

- a) Fila de impressão de documentos
- b) Sistema de atendimento ao cliente por ordem de chegada
- c) Histórico de navegação em um browser (botão voltar)
- d) Lista de contatos em ordem alfabética

Resposta: c) Histórico de navegação em um browser (botão voltar)