

# Estruturas de Dados

## Pilhas

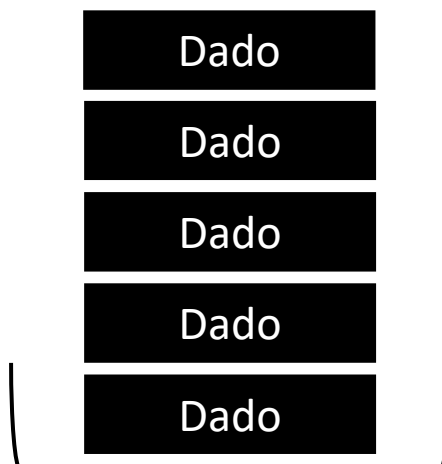
Universidade Federal do Rio Grande do Norte

# Pilha

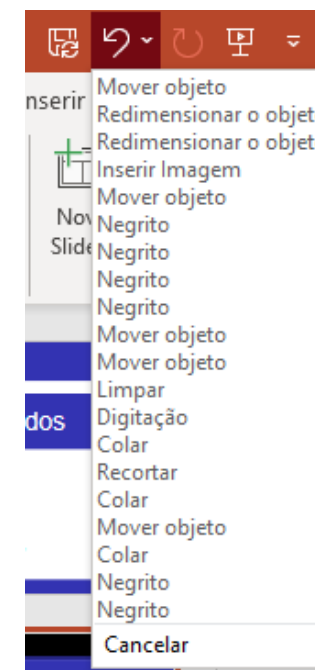
## Definição

Conjunto **ordenado de dados** no qual os dados podem ser **inseridos** ou **removidos** a partir de uma extremidade chamada **topo** da pilha.

Topo da pilha



[Esta Foto](#) de Autor Desconhecido  
está licenciado em [CC BY-SA](#)

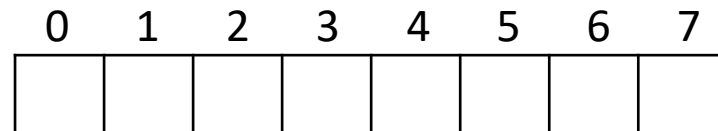


# Pilha

- Chamada de “stack” em inglês.
- É possível dizer que uma pilha é um tipo abstrato de dados baseado no princípio de **Last-In, First-Out**
- LIFO ou “o último a entrar é o primeiro a sair”.
- A ordem que os elementos são retirados da pilha é oposta a ordem de inserção.
- A operação de **INSERIR** sobre uma pilha é geralmente chamada de **PUSH** e possui como parâmetro o dado que será inserido na pilha.
- A operação de **REMOVER** sobre uma pilha é geralmente chamada de **POP** e não possui parâmetros.

# Pilha

- É possível implementar uma pilha de ***n*** elementos com um **array** para armazenar os dados e um atributo, chamado *topo*, para armazenar o **índice** do elemento inserido mais recentemente.



- Quanto  $\text{topo} = 0$  a pilha não contém nenhum elemento e está **vazia**.
- Quando  $\text{topo} = n$ , podemos dizer que a pilha está **cheia**.

# Pilha

- Se uma pilha vazia sofre uma operação de remoção (pop) dizemos que a pilha tem um **estouro negativo** (ou *stack underflow*)
- Se uma pilha cheia sofre uma operação de inserção (push) dizemos que a pilha tem um **estouro positivo** (ou *stack overflow*)
- Ambas as situações devem ser tratadas como **erro**.

# Principais Operações

**Push (dado)**

Adiciona um dado à pilha.

**Pop()**

Remove um dado da pilha.

**Top ()**

Retorna o dado que está no topo da pilha.

**isEmpty()**

Verifica se uma pilha está vazia.

# Outras Operações

**Clear ()**

Remove todos os dados da pilha.

**Size()**

Retorna a quantidade de dados presentes na pilha.

**toString()**

Retorna em formato de texto o conteúdo da pilha.

# Implementação

```
STACK-EMPTY(S)  
1 if topo[S] = 0  
2   then return TRUE  
3   else return FALSE
```

```
PUSH(S, x)  
1 topo[S]  $\leftarrow$  topo[S] + 1  
2 S[topo[S]]  $\leftarrow$  x
```

```
POP(S)  
1 if STACK-EMPTY(S)  
2   then error “underflow”  
3   else topo[S]  $\leftarrow$  topo[S] - 1  
4       return S[topo[S] + 1]
```

FONTE: CORMEN et al; Algoritmos: Teoria e Prática. 3º Edição. 2012



## Bibliografia Básica

- CORMEN, Thomas H et al. **Algoritmos: teoria e prática.** Rio de Janeiro: Elsevier, 2012. 926 p. ISBN: 9788535236996.
- ASCENCIO, Ana Fernanda Gomes. **Estruturas de dados: algoritmos, análise da complexidade e implementações em Java e C/C++.** São Paulo: Pearson, c2010. 432 p. ISBN: 9788576052216, 978857605816.
- PIVA JÚNIOR, Dilermando (et al). **Estrutura de dados e técnicas de programação.** 1. ed. Rio de Janeiro, RJ: Campus, 2014. 399 p. ISBN: 9788535274370.

## Bibliografia Complementar

- FERRARI, Roberto et al. **Estruturas de dados com jogos**. 1. ed. Rio de Janeiro: Elsevier, 2014. 259p. ISBN: 9788535278040.
- GRONER, Loiane. **Estruturas de dados e algoritmos em Javascript**: aperfeiçoe suas habilidades conhecendo estruturas de dados e algoritmos clássicos em JavaScript. São Paulo: Novatec, 2017. 302 p. ISBN: 9788575225530.
- SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. **Estruturas de dados e seus algoritmos**. 3. ed. Rio de Janeiro: LTC, 2010. xv, 302 p. ISBN: 9788521617501.
- GOODRICH, Michael T; TAMASSIA, Roberto. **Estruturas de dados e algoritmos em Java**. 5. ed. Porto Alegre: Bookman, 2013. xxii, 713 p. ISBN: 9788582600184.
- GUIMARÃES, Ângelo M. **Algoritmos e estruturas de dados**. LTC, 1994.