

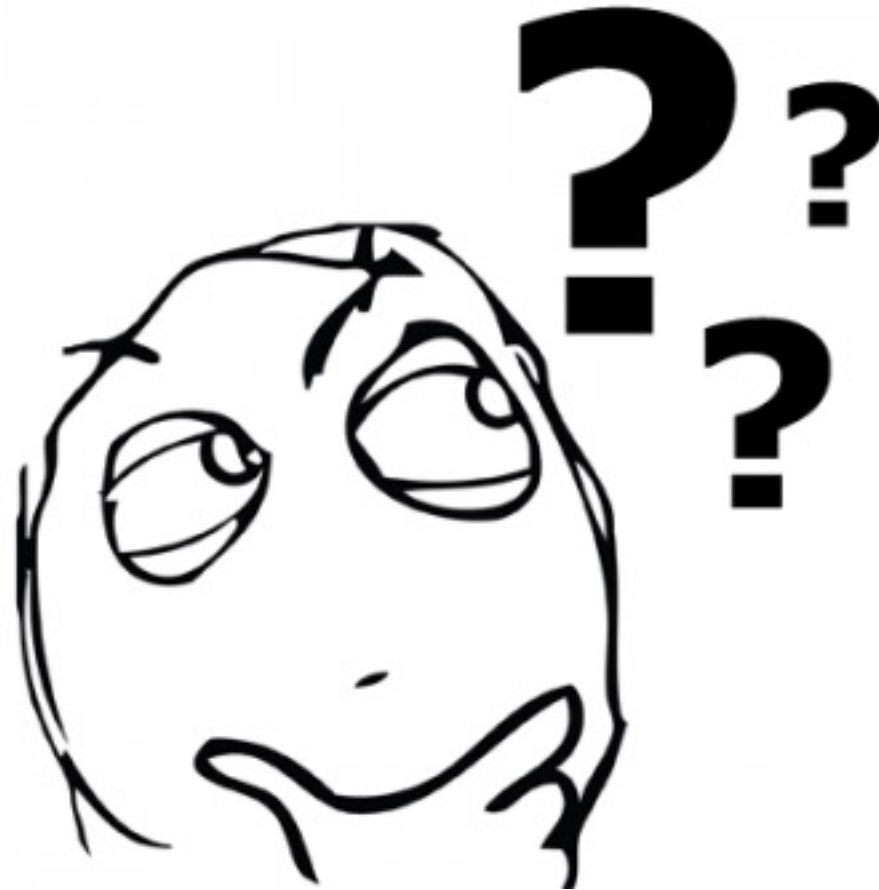


Estrutura de Dados

Aula 4

GUSTAVO FORTUNATO PUGA

Dados



Dados

Os dados (e seus diversos tipos) são os **blocos básicos da programação**. Eles representam uma unidade ou um **elemento de informação que pode ser acessado através de um identificador** - por exemplo, uma variável.

- **INT** ou **número inteiro**: valores numéricos inteiros (positivos ou negativos);
- **FLOAT** ou o chamado "**ponto flutuante**": valores numéricos com casas após a vírgula (positivos ou negativos);
- **BOOLEAN** ou **booleanos**: representado apenas por dois valores, "verdadeiro" e "falso". Também chamados de operadores lógicos;
- **TEXT**: **sequências ou cadeias de caracteres**, utilizados para manipular textos e/ou outros tipos de dados não numéricos ou booleanos, como hashes de criptografia.



Estrutura de Dados



Estrutura de Dados

Dados são **agregados e organizados** dependendo de como serão utilizados e processados, levando-se em consideração, por exemplo, a eficiência para buscas, o volume dos dados trabalhados, a complexidade da implementação e a forma como os dados se relacionam.

As diversas formas de organização são as chamadas estruturas de dados.



Estrutura de Dados

Cada estrutura de dados tem um conjunto de métodos próprios para realizar operações como:

- Inserir ou excluir elementos;
- Buscar e localizar elementos;
- Ordenar (classificar) elementos de acordo com alguma ordem especificada.



Características das Estrutura 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).



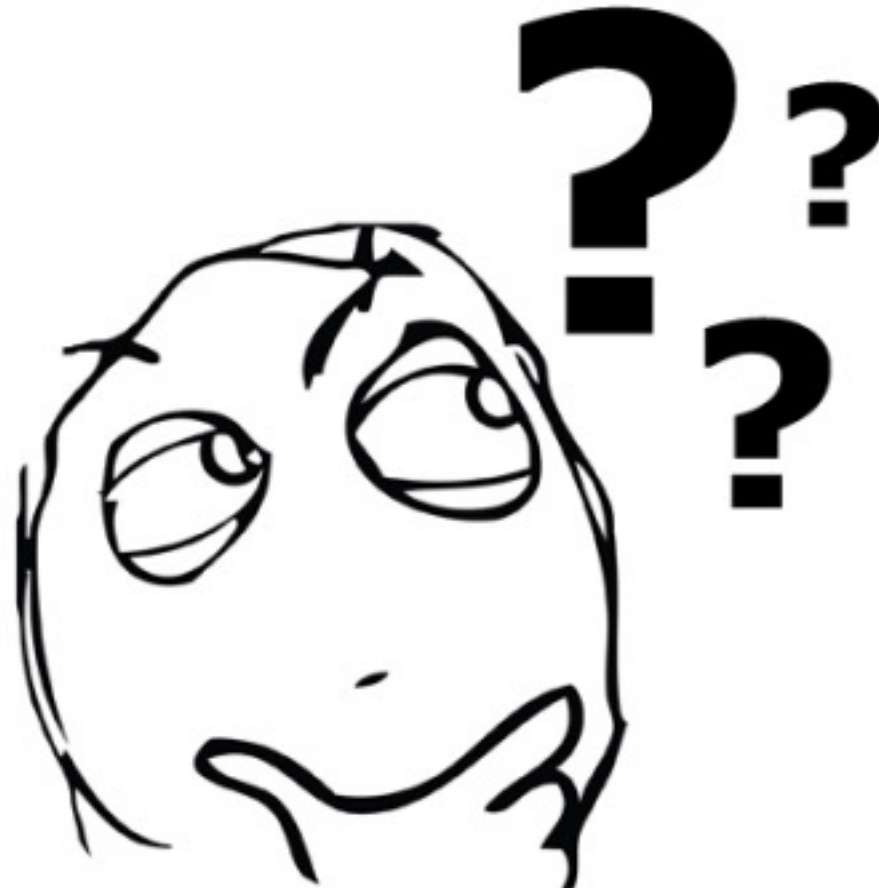
Array

- Também chamado de **vetor, matriz ou arranjo**, o **array** é a mais comum das estruturas de dados e normalmente é a primeira que estudamos.

Os arrays também podem ser multidimensionais, sendo utilizados sempre que há necessidade de tabular dados e os arrays de 2 dimensões (matrizes) são utilizados para processamento de imagens

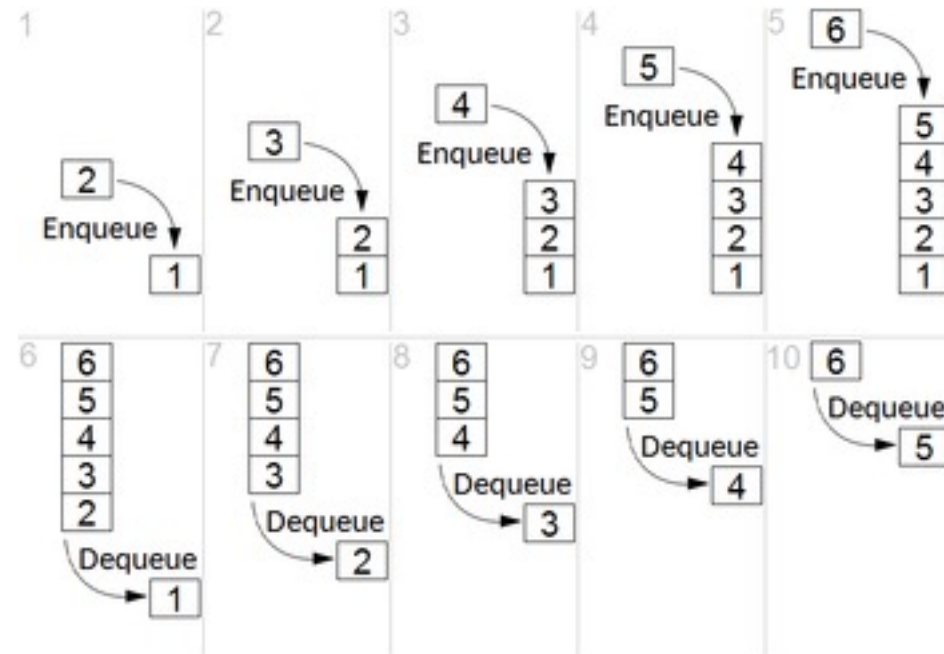


E qual o problema com Arrays?



Fila

Pense em uma fila de bilheteria, por exemplo. A pessoa que chegou antes vai ser atendida (e comprar seu ingresso) antes de quem chegou depois e ficou atrás na fila. A fila como estrutura de dados segue o mesmo princípio

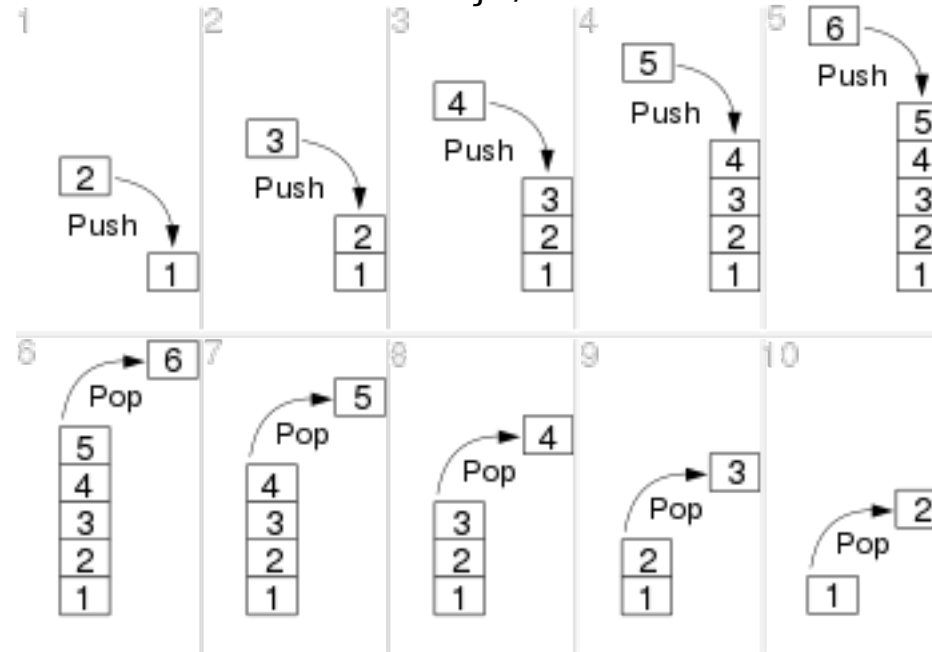


FIFO (*First In, First Out* = primeiro a entrar, primeiro a sair)



Pilha

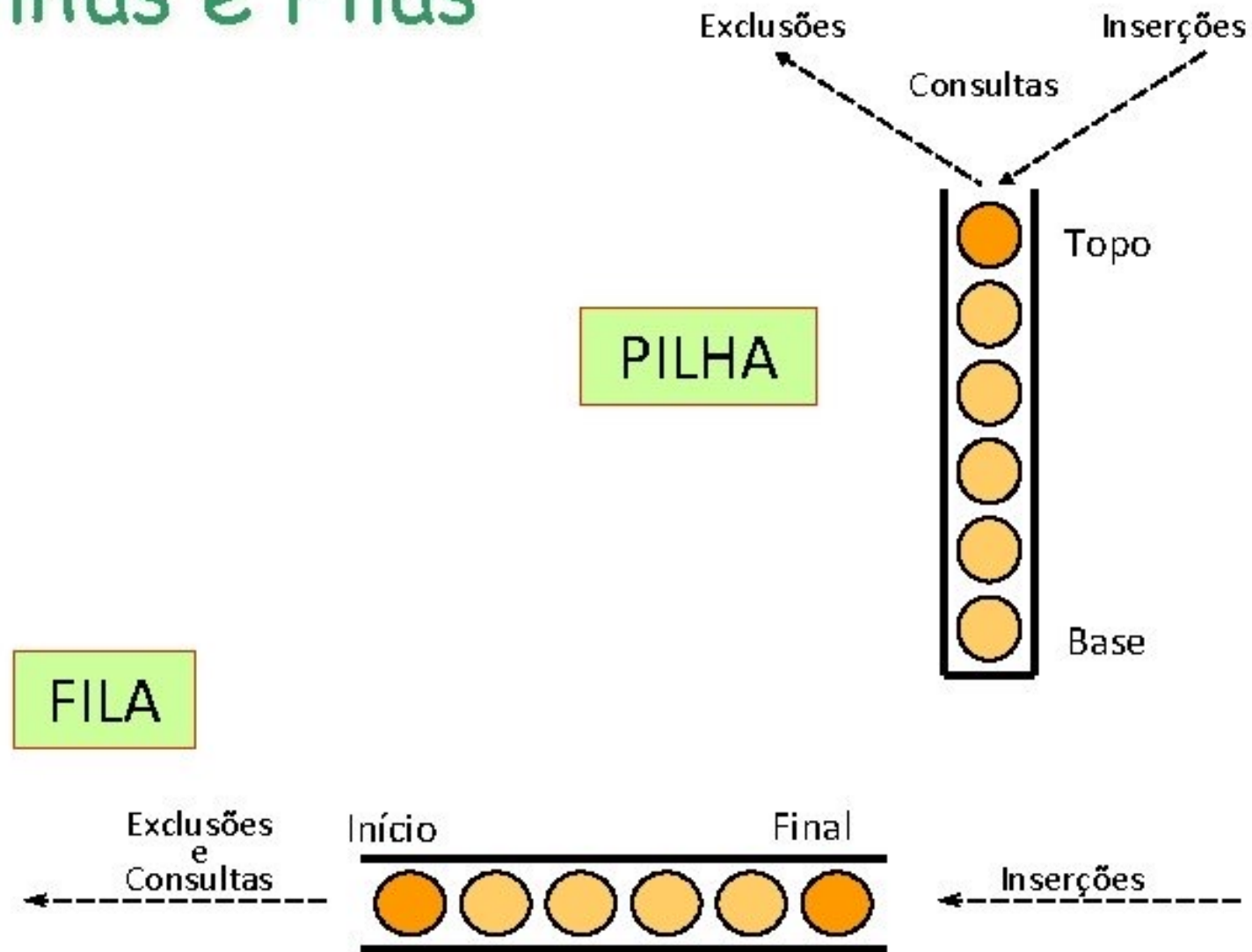
Pense em uma *pilha* de livros ou de pratos. Ao empilharmos livros, por exemplo, o primeiro livro a ser retirado da pilha é obrigatoriamente o último que foi colocado; se tentarmos retirar o último livro da pilha, tudo vai desabar. Ou seja, o último livro a ser empilhado é o primeiro a ser retirado.



LIFO (*Last In, First Out* = último a entrar, primeiro a sair)

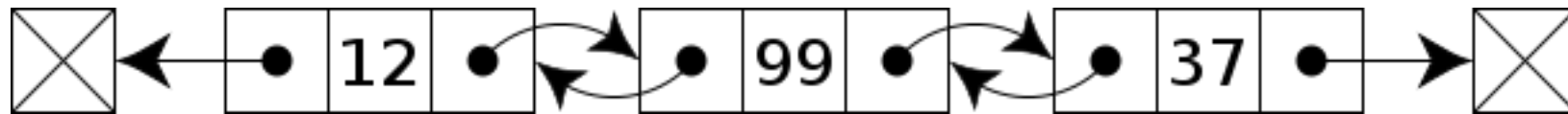


Pilhas e Filas



Lista duplamente ligada (Deque)

Variação da fila que aceita inserção e remoção de elementos tanto do início quanto do final da fila.



Conjuntos (*Set*)

A estrutura de dados conjunto (ou set) é uma lista não ordenada de elementos únicos. Ou seja, não é possível repetir o valor de um elemento dentro de um conjunto.

Vem da matemática, e também é possível fazer operações como união e intersecção em conjuntos de dados. Um dos usos mais comuns desta estrutura é em bancos de dados SQL.



Dicionário ou *hashmap*

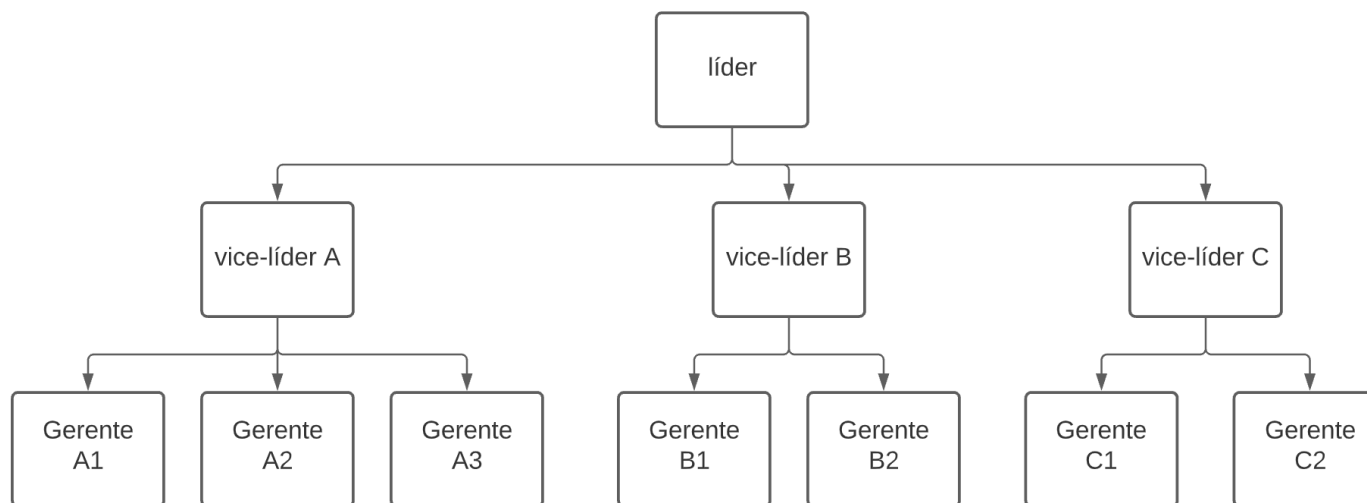
Também conhecido como **mapa** (*map*) é uma estrutura que guarda dados em pares de chave e valor e utiliza estas chaves para encontrar os elementos associados a elas, diferentemente das estruturas que vimos até agora, que trabalham com listas (sequenciais ou não) apenas de valores.



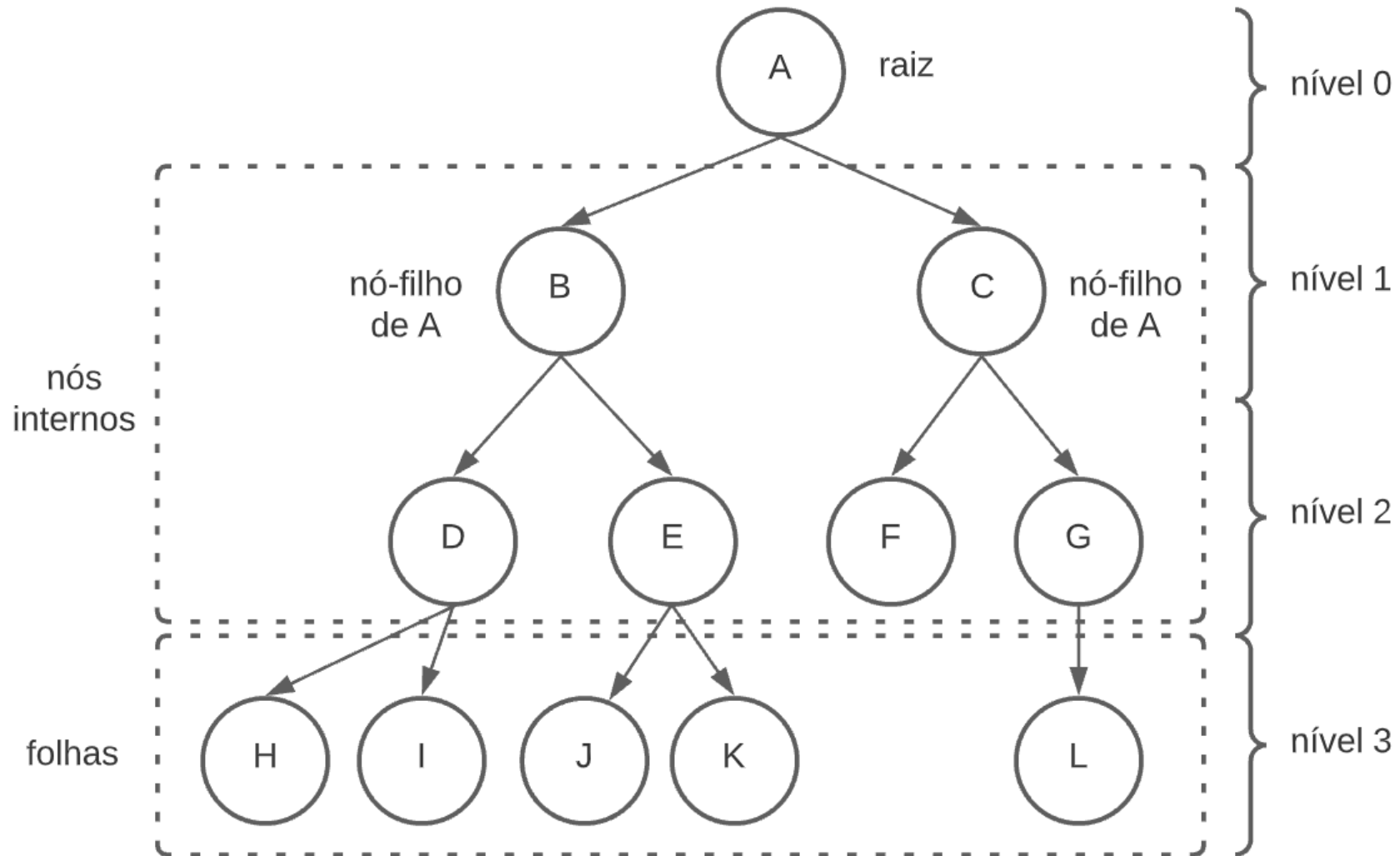
Árvore (tree)

Estrutura não-sequencial, muito útil para armazenar dados de forma hierárquica e que podem ser acessados de forma rápida.

Pode-se definir árvore como uma coleção de dados representados por nós e arranjos em níveis hierárquicos (ao invés de sequências como as estruturas vistas anteriormente).



Árvore (tree)

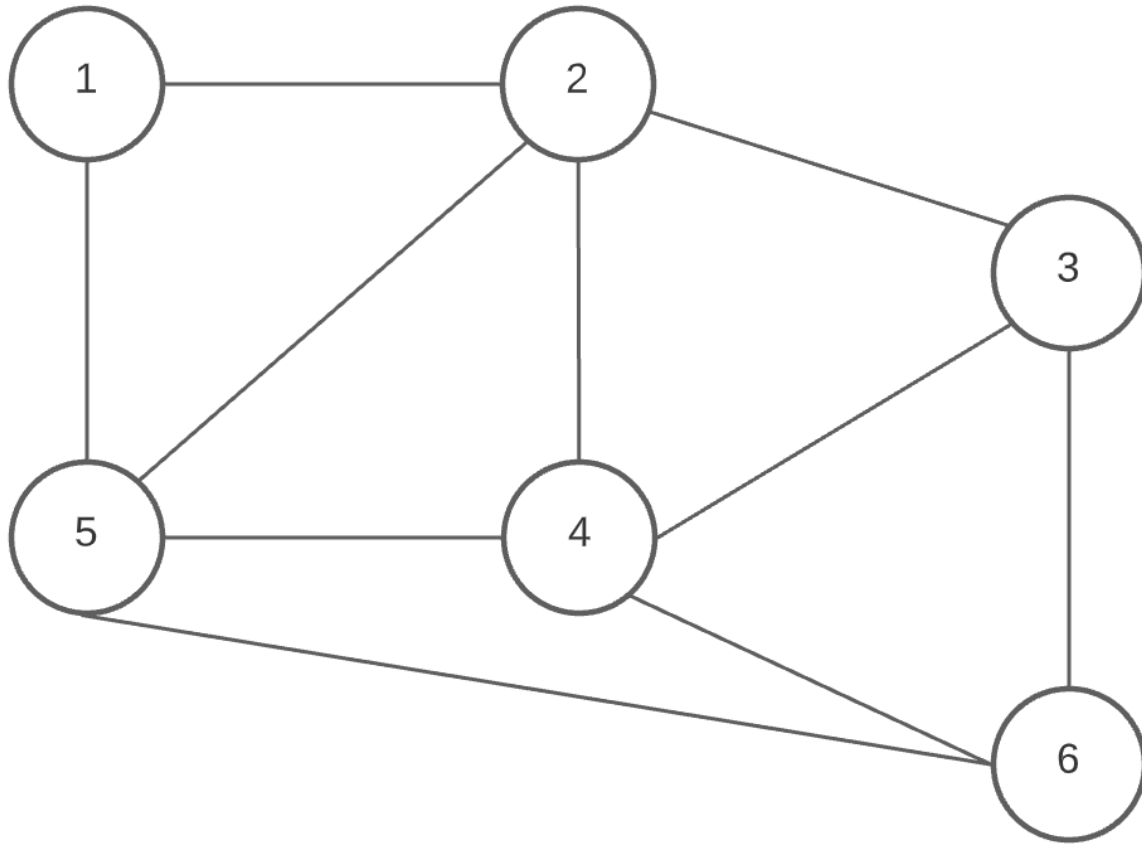


Árvore (tree)

Árvore tem vários usos diversos, como algoritmos de tomada de decisão em aprendizado de máquina, indexação de bancos de dados, indexação e exibição de arquivos e pastas no explorador de arquivos dos sistemas operacionais, entre vários outros casos.



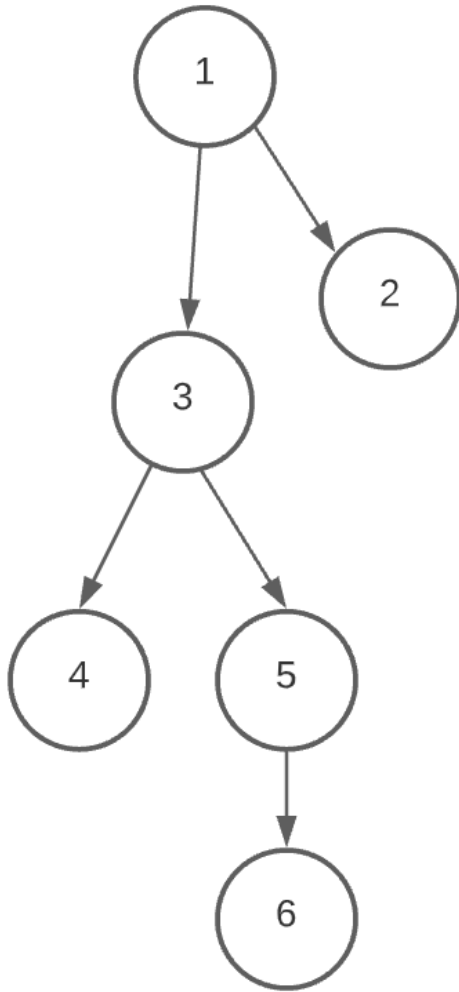
Grafo



Estrutura não-sequencial,
o **grafo** (*graph*) é um conjunto
de nós (ou vértices), ordenados
ou não e ligados por arestas,
formando uma estrutura em
forma de rede



Grafo



As redes sociais utilizam os grafos para manejar a grande quantidade de dados relacionados entre si que recebem a cada instante. O exemplo mais famoso, a linguagem de consulta GraphQL, foi criada pelo Facebook com o intuito de utilizar grafos para acessar e relacionar dados.

Outro uso famoso para os grafos é o sistema de navegação dos aplicativos de mapas/GPS, que utilizam grafos e o algoritmo de caminho mínimo (ou *shortest path*) para traçar rotas.



Bibliografia

<https://www.alura.com.br/artigos/estruturas-de-dados-introducao>

BARNES, David J.; KOLLING, Michael. **Programação orientada a objetos com Java**. 4. ed. São Paulo: Prentice Hall - Br, 2009.

Aditya Y. Bhargava. **Entendendo Algoritmos**. Novatec Editora. ISBN 9788575226629.

Algoritmos. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Campus.

Algorithms. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani. McGraw Hill.

Concrete Mathematics: A Foundation for Computer Science (2nd Edition). Ronald L. Graham, Donald E. Knuth, Oren Patashnik. Addison Wesley.

DEITEL, Paul; DEITEL, Harvey. **Java: como programar**. 10. ed. São Paulo: Pearson, 2017. ISBN 9788543004792. Disponível em: <https://ifsp.bv3.digitalpages.com.br/users/publications/9788543004792>. Acesso em: 14 jun. 2019.

Helder da Rocha.. **Gerência de memória em Java**. Argonavis: 2005.

Sierra, Kathy. **Use A Cabeça Java**. S.l: Alta Books, 2009. ISBN: 8576081733





**INSTITUTO
FEDERAL**

São Paulo

Câmpus
São Paulo

Obrigado!

Gustavo Fortunato Puga

gustavo.puga@ifsp.edu.br

