

# Algoritmos - Aula 1

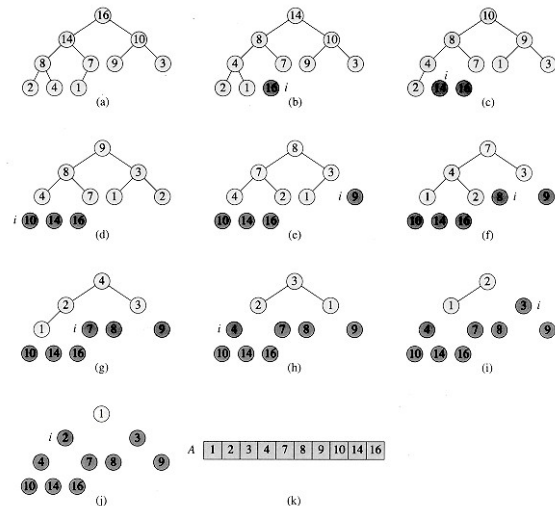
Fernando Raposo

# Apresentação

- Quem sou eu?
- Podem se apresentar?
  - Nome
  - Formação
  - Já trabalham?
  - Quais os seus interesses em computação?
    - Quais linguagens usam?

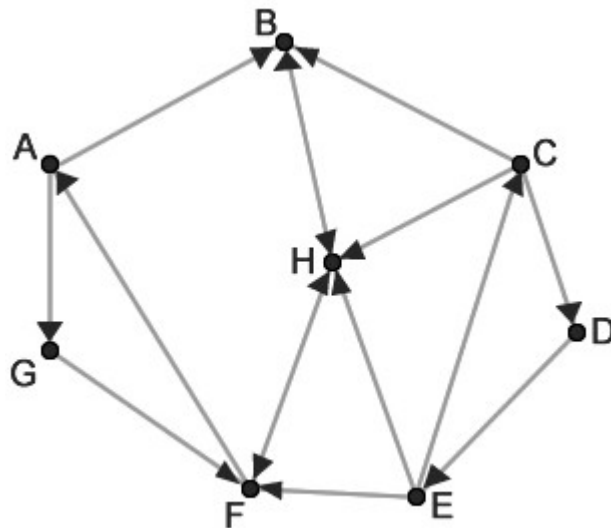
# Conteúdo Programático

- Revisão: Estruturas de Dados
- Java Crash Course
- Recursividade
  - Fatorial (!)
  - Torre de Hanói
- Algoritmos
  - Conceitos
  - Dividir para conquistar
  - Algoritmos de Ordenação
    - Bubble sort
    - Insertion sort
    - Selection sort
    - Merge sort
    - Heap sort



# Conteúdo Programático

- Análise de Algoritmos
- Complexidade Computacional
- Notação Big-O
- Grafos x Algoritmos
  - Busca em largura
  - Busca em profundidade
  - Algoritmo Dijkstra
  - Caixeiro Viajante
- Aplicação de Algoritmos em problemas de computação de dados
- Escolha de estruturas de dados
- Redução e Equivalência entre problemas



# Condução

- Aulas Teóricas
- Aulas Práticas (Laboratório)
- Google *Classroom*
- Listas de Exercícios
- Trabalhos valendo percentual da nota
- 2 provas

# Algoritmos

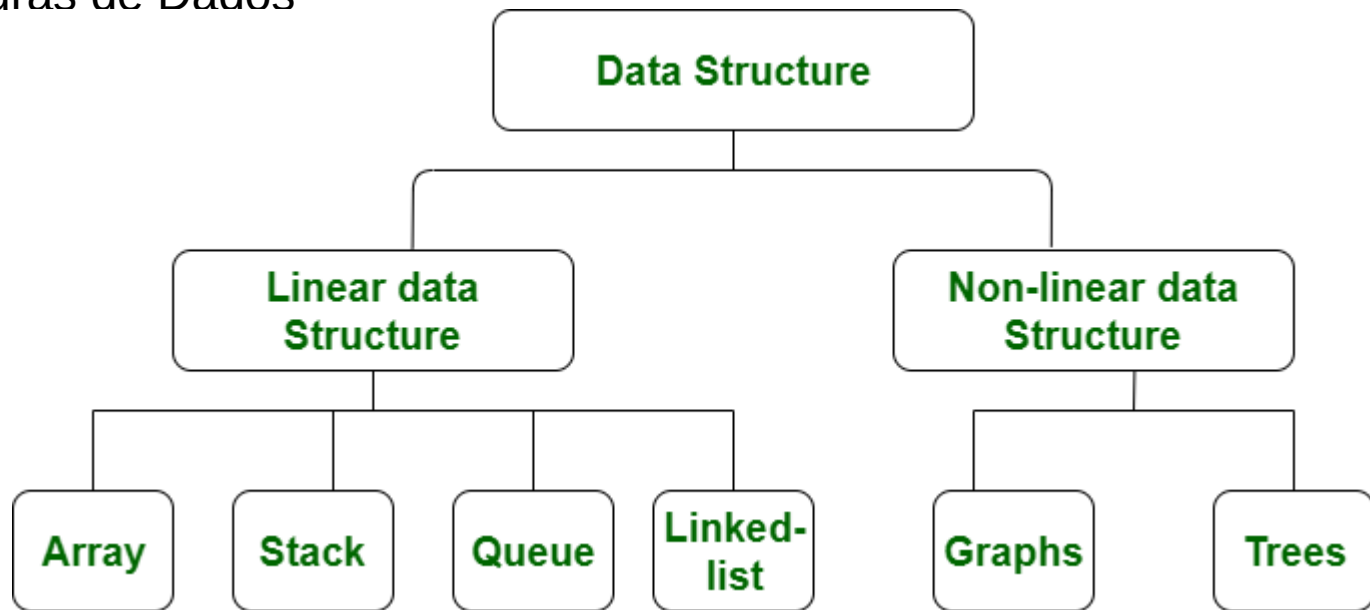
- Qual sua finalidade?
  - Auxiliar na resolução de problemas
  - Melhor gerenciamento de recursos
    - Memória;
    - Velocidade.



- Você sabe o que são as [Whiteboard Interviews](#)?
  - Estruturas de Dados e Algoritmos auxiliam em Whiteboard Interviews
    - Processo de seleção usado no [Google, Amazon, Microsoft...](#)
- Maratonas de Programação

# Revisão...

...Estruturas de Dados

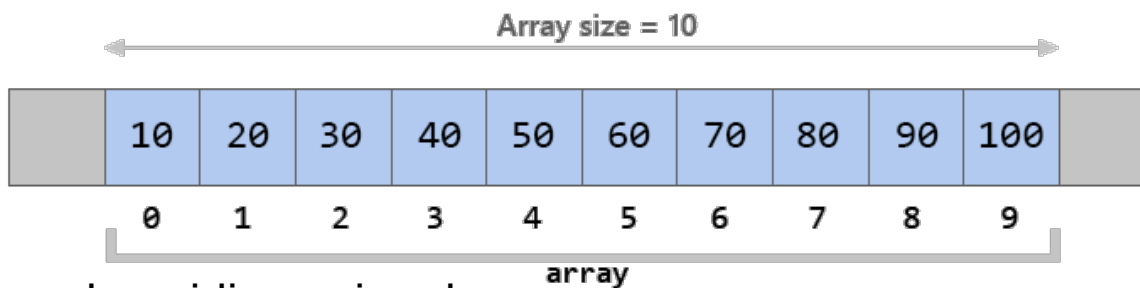


# Como Armazenamos Estes Dados?

- Resultados do Enem;
- Pedidos em uma cozinha de um restaurante;
- Resultado de busca do Google;
- Menor caminho entre dois pontos um mapa;



# Vetores



## Variável indexada unidimensional

- A partir de um único nome e de um número (o índice), permite o armazenamento e a localização de um conjunto de dados.

### Java

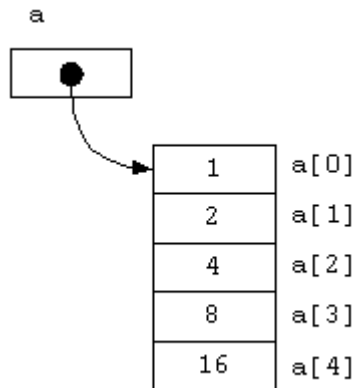
```
String[] carros = {"Ford", "Fiat", "VW",  
"Renault"};  
  
for (int i = 0; i < carros.length; i++) {  
  
    System.out.println(carros[i]);  
  
}
```

### Python

```
carros = ["Ford", "Fiat", "VW", "Renault"]  
  
for x in carros:  
    print(x)
```

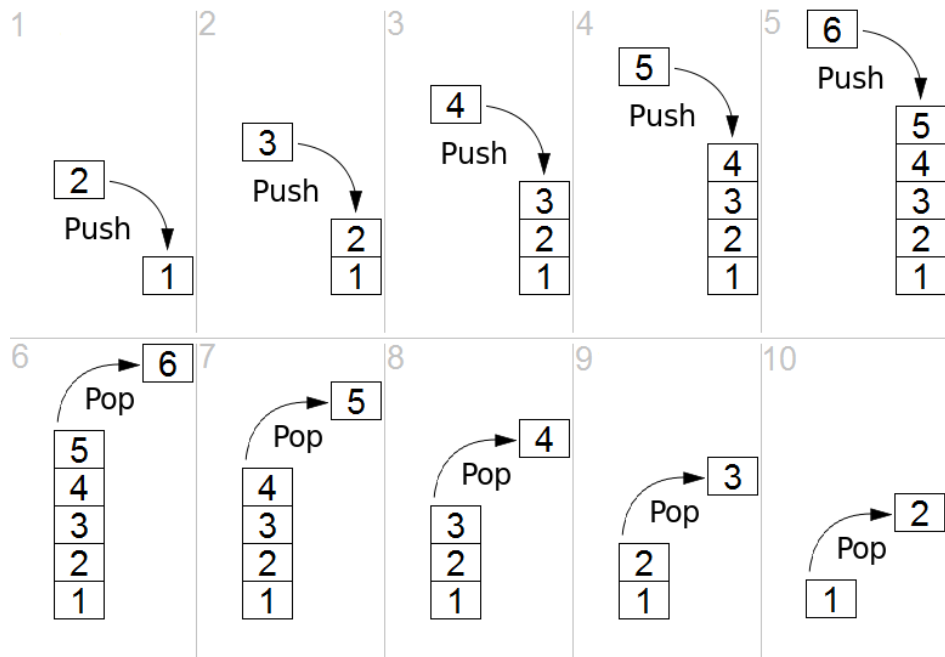
# Vetores

- Os Elementos são todos do mesmo tipo (ou não);
- Os Elementos do vetor (array) estão em posições subsequentes na memória;
- Permite acesso aleatório sem atravessarmos todo o array;
- Vetores são úteis para implementarmos outros conceitos:
  - Filas
  - Pilhas
- Operações:
  - Pesquisa de um elemento específico
  - Inserção de um elemento
  - Remoção



# Pilhas

- Podemos utilizar vetores para implementar uma pilha;
- Lógica dos pratos a lavar;
- Há um indicador da posição atual do topo da pilha;
- Pode haver uma constante que nos diga quando a pilha está cheia e duas outras para codificar erros. (Pilha cheia, Pilha vazia)



# Pilhas

- Pilhas utilizam a estratégia LI-FO (*Last In, First Out*)
- O último elemento que entrou é o primeiro a sair

- Já vimos exemplos em Java e Python

## Exercício Prático

- Vamos implementar uma pilha no [Node](#) em Javascript?
  - Pilha Raiz
  - Pilha Nutella



# Filas

- Podemos utilizar vetores para implementar uma fila;
- Lógica de uma fila de banco - **abstrair prioridades**
- Enfileirar significa colocar um elemento no final da fila;
- Desenfileirar significa remover o primeiro elemento da fila.



# Filas

- Filas utilizam a estratégia FI-FO (*First In, First Out*)
- O primeiro a chegar é o primeiro a sair

## Exercício Prático

- Vamos implementar uma fila no [Node](#) em Javascript?
  - Fila Raiz
  - Fila Nutella

1ª linha →  $\begin{bmatrix} 1 & 4 & 7 \end{bmatrix}$   
 2ª linha →  $\begin{bmatrix} 2 & \sqrt{3} & -3 \end{bmatrix}$   
 3ª linha →  $\begin{bmatrix} 0 & 0 & 5 \end{bmatrix}$

↑  
 ↑  
 ↑  
 3ª coluna  
 2ª coluna  
 1ª coluna

- X = Critical to role

```
int[][] matriz = new int[3][3];
```

ou

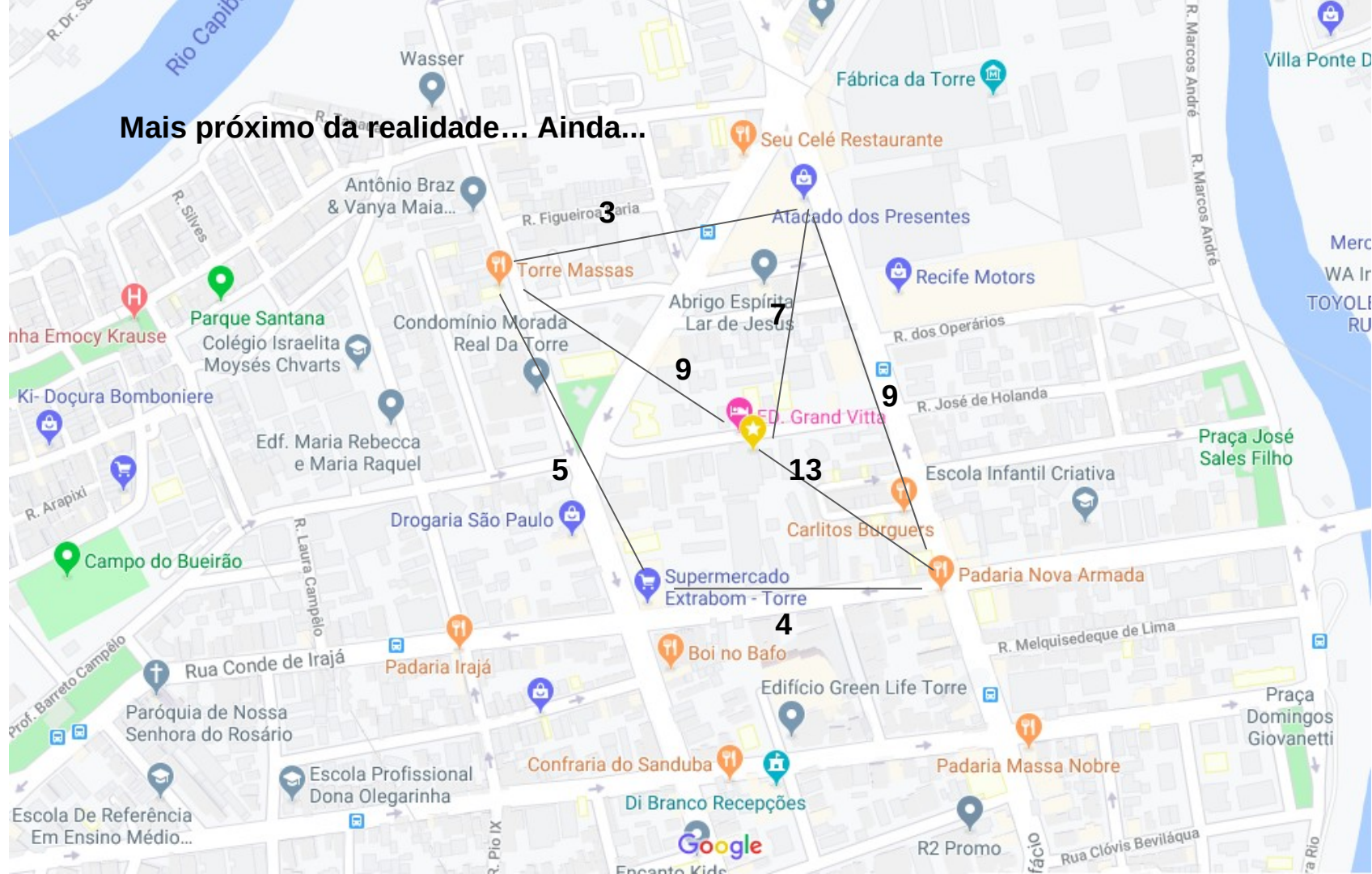
		Project Management	Leadership	Financial Acumen	Six Sigma	Help Desk Software	PC Hardware	PC Software	Customer Service	Written Communication	Presentation
	Role										
Joe Smith	Supervisor		X	X				X	X	X	
Sally Johnson	Technician				X	X	X	X			
Susan Jones	Technician				X	X	X	X			
Alex Taylor	Technician				X	X	X	X			
Sandra Day	Technician				X	X	X	X			
Frank Wright	Analyst	X		X	X	X	X	X	X	X	X
Jane Doe	Quality Engineer			X				X	X	X	X
Barbara Steel	Project Manager	X	X	X				X	X	X	X

```
matriz = [[1,2,3],[4,5,6],[7,8,9]]
```

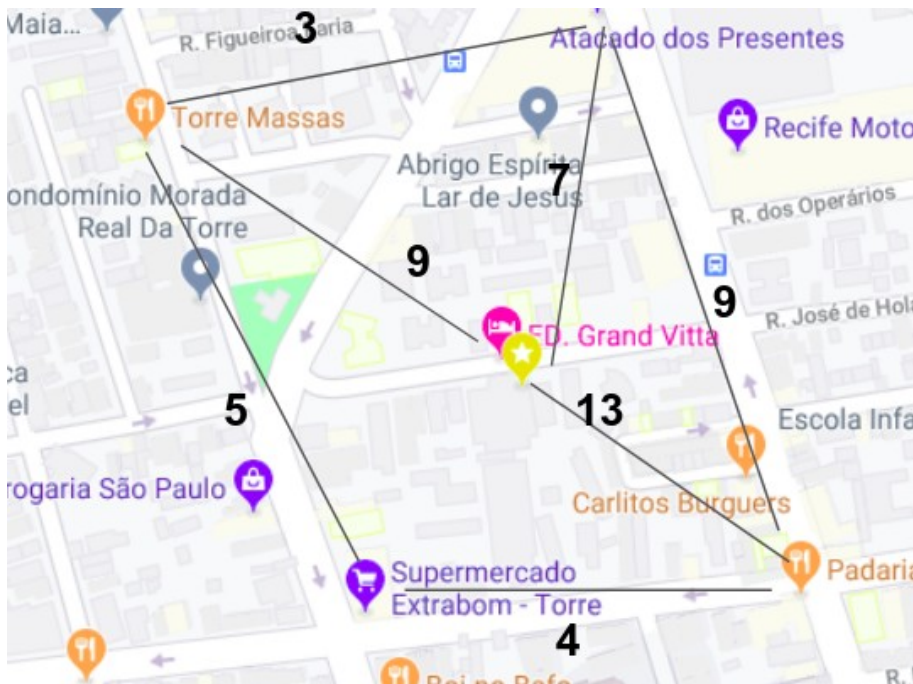
## Mais próximo da realidade...



Mais próximo da realidade... Ainda...



# Uma Matriz pode representar um mapa



	Casa	AtP	ToM	ExB	NoA
Casa	0	7	9	X	13
AtP	7	0	3	X	9
ToM	9	3	0	5	X
ExB	X	X	5	0	4
NoA	13	9	X	4	0

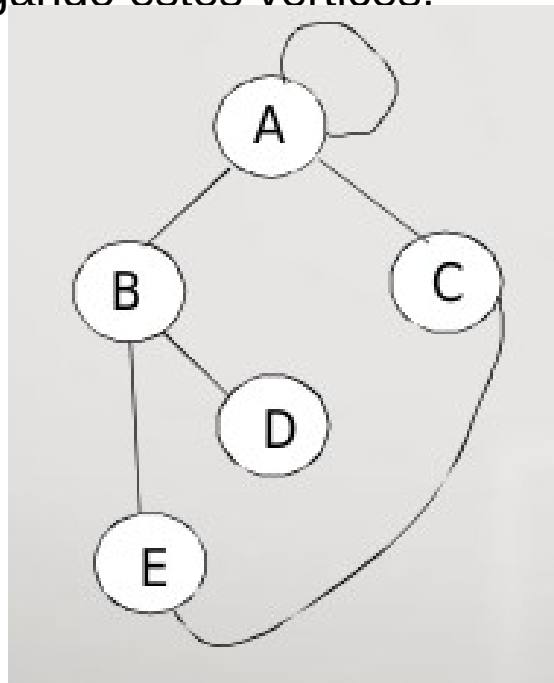
# Grafos

- São estruturas de dados formadas por um conjunto de não vazio de **vértices** (ou nós) e por um conjunto de **arestas** (ou arcos), ligando estes vértices.

Vértices = {A, B, C, D, E}

Arestas = {(A,A), (A,B), (A, B), (A,C), (B,D), (B,E), (C,E)}

- As aplicações com grafos são inúmeras...



# Grafos

- Dos grafos derivamos ainda as árvores;
- **Toda árvore é um grafo, mas nem todo grafo é uma árvore.**

## Exercício Prático

Vamos construir uma matriz 4 x 5 e depois vamos multiplicar todos os seus valores por 2?

