



# Estrutura de Dados Avançado

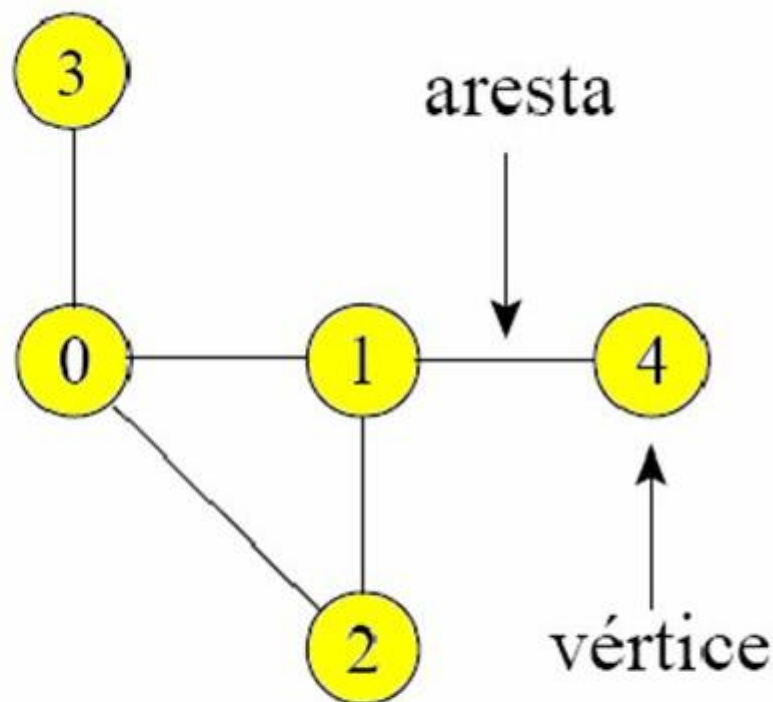
**Prof<sup>a</sup>. Angela Abreu Rosa de Sá, Dr<sup>a</sup>.**

---

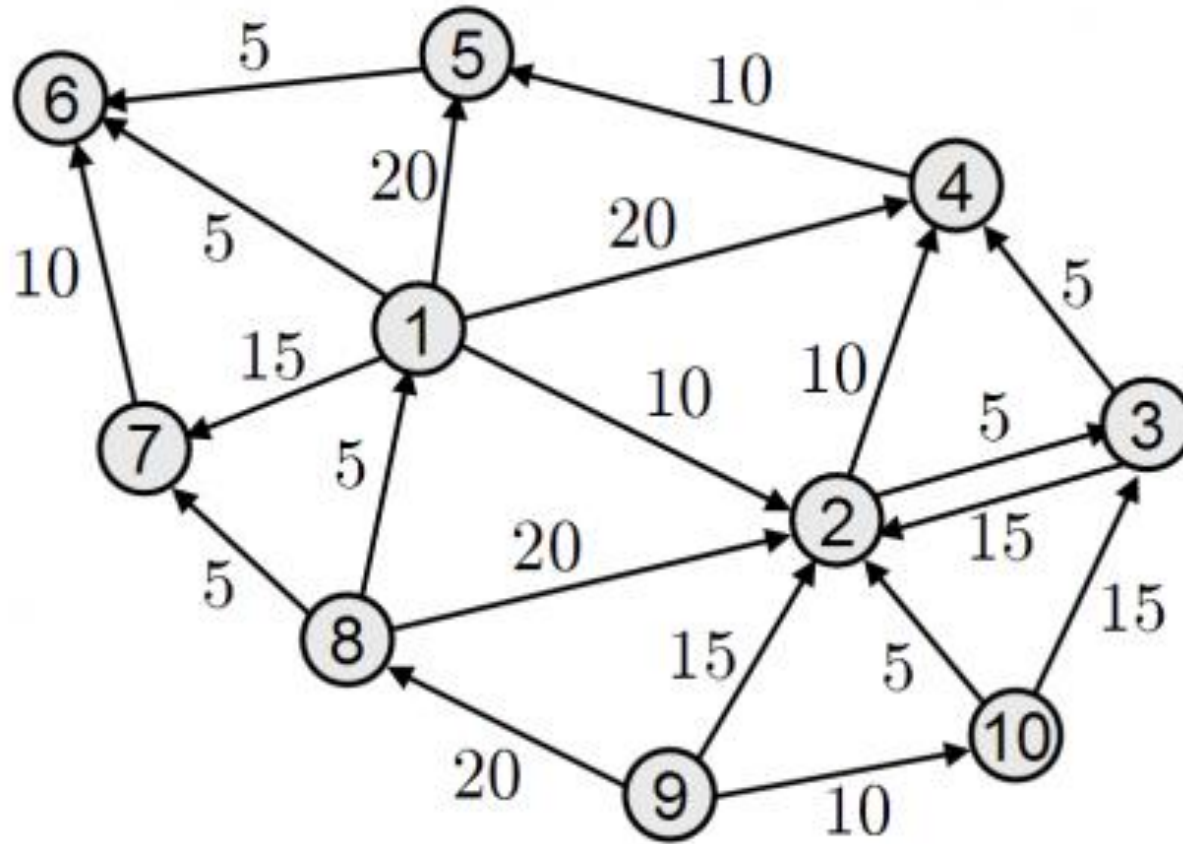
*Contato: [angelaabreu@gmail.com](mailto:angelaabreu@gmail.com)*

# Grafos

- Grafo: Conjunto de Vértices e Arestas
- Vértice: Objeto simples que pode possuir um nome e outros atributos
- Aresta: Conexão entre dois vértices
- Notação:  $G = (V, A)$ 
  - V : Conjunto de Vértices
  - A: Conjunto de Arestas



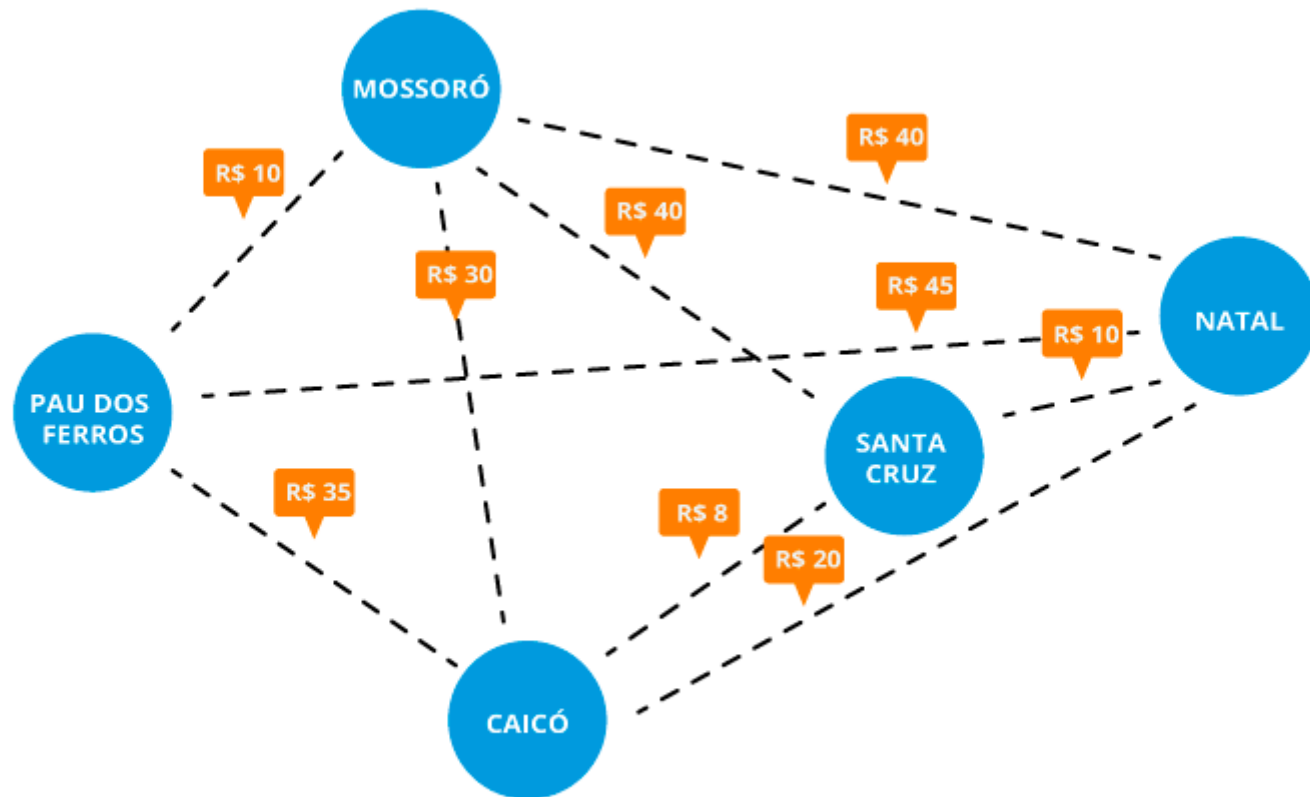
# Grafos



# Grafos



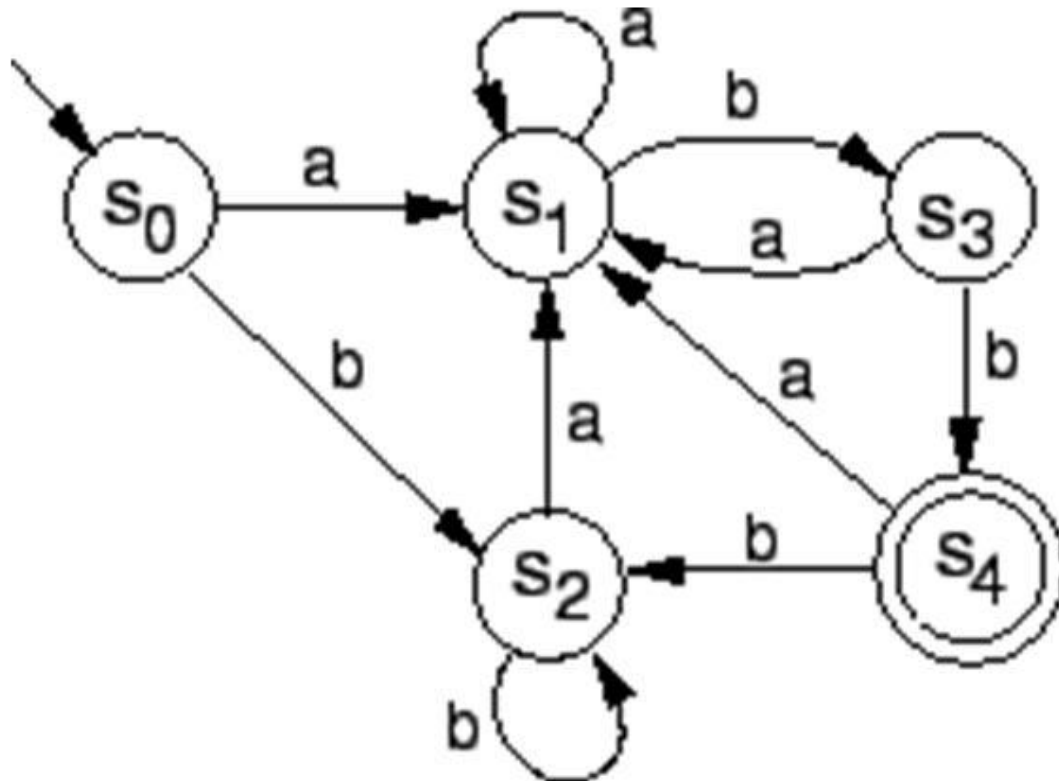
# Grafos



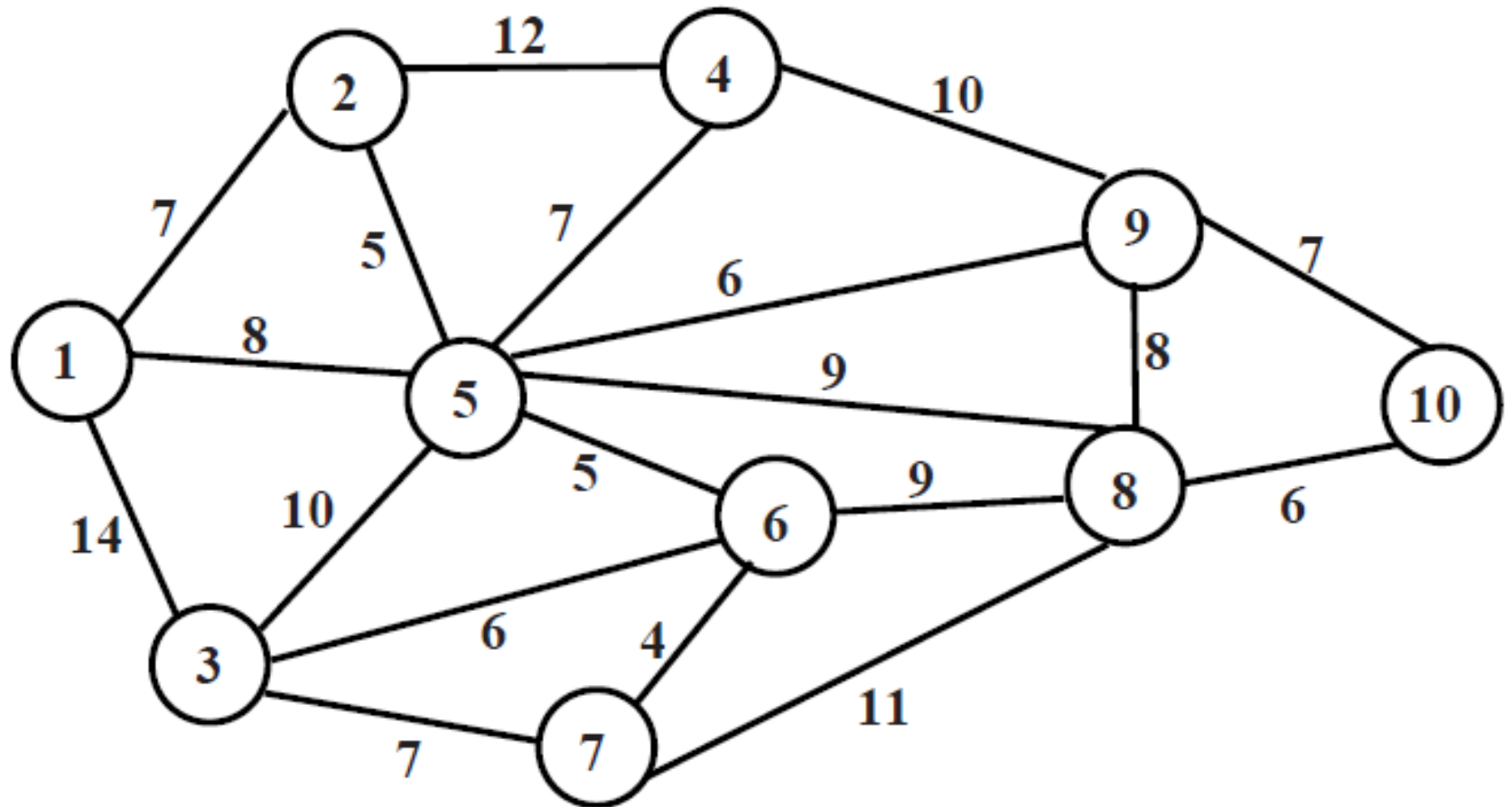


# Grafos

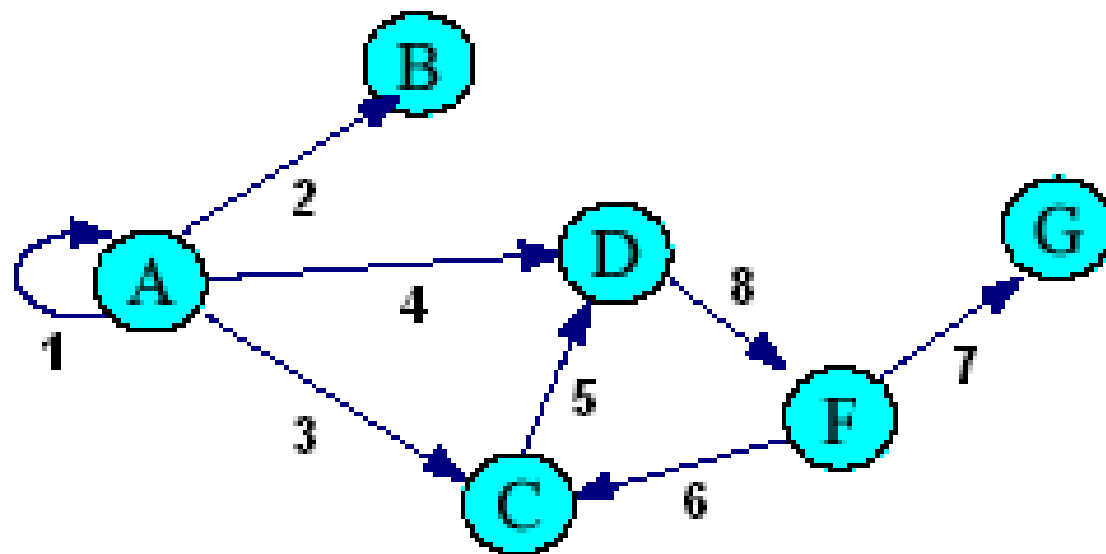
Autômato Finito determinístico para  $(a|b)^*abb$



Todo grafo é uma árvore , mas .....



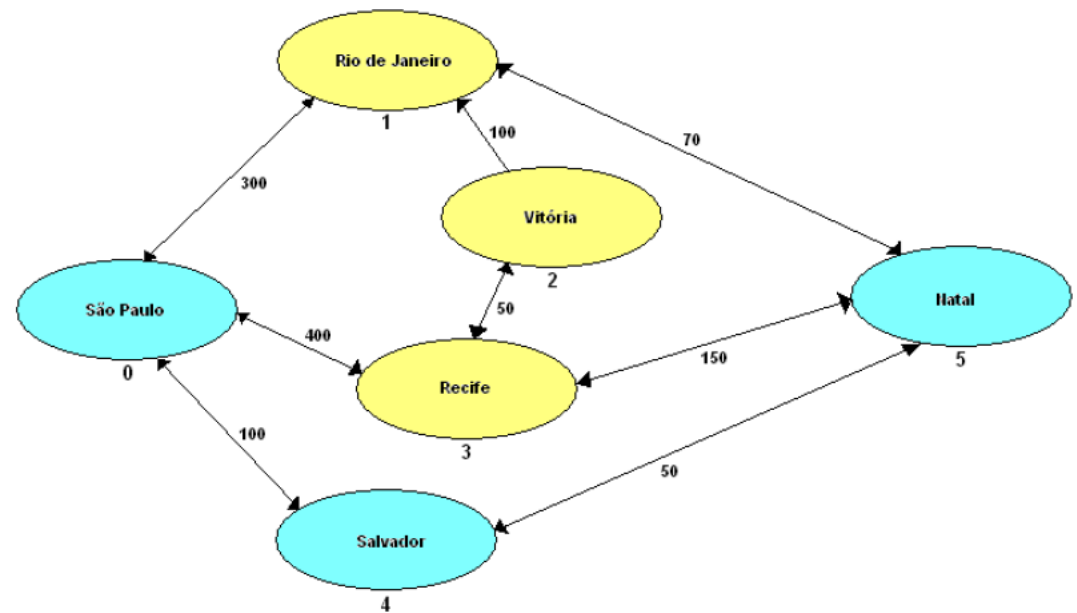
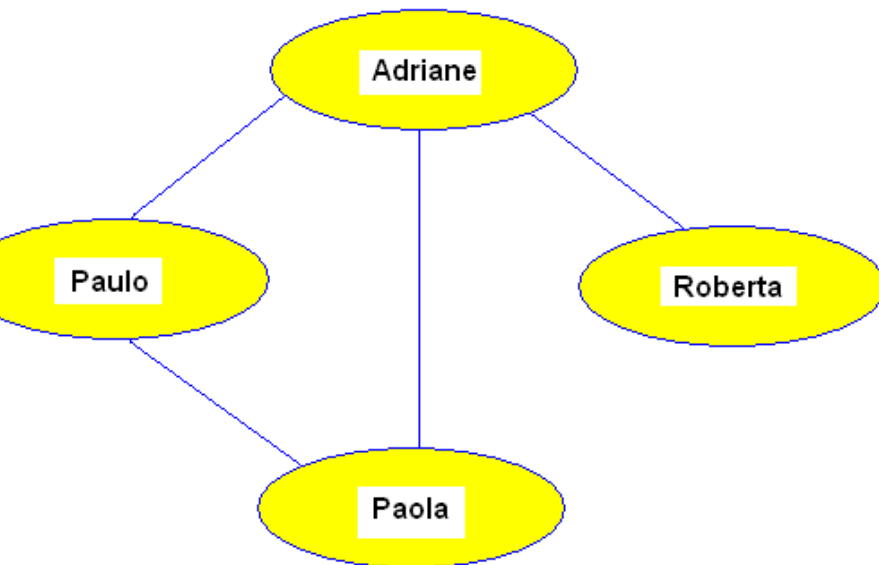
# Grafos





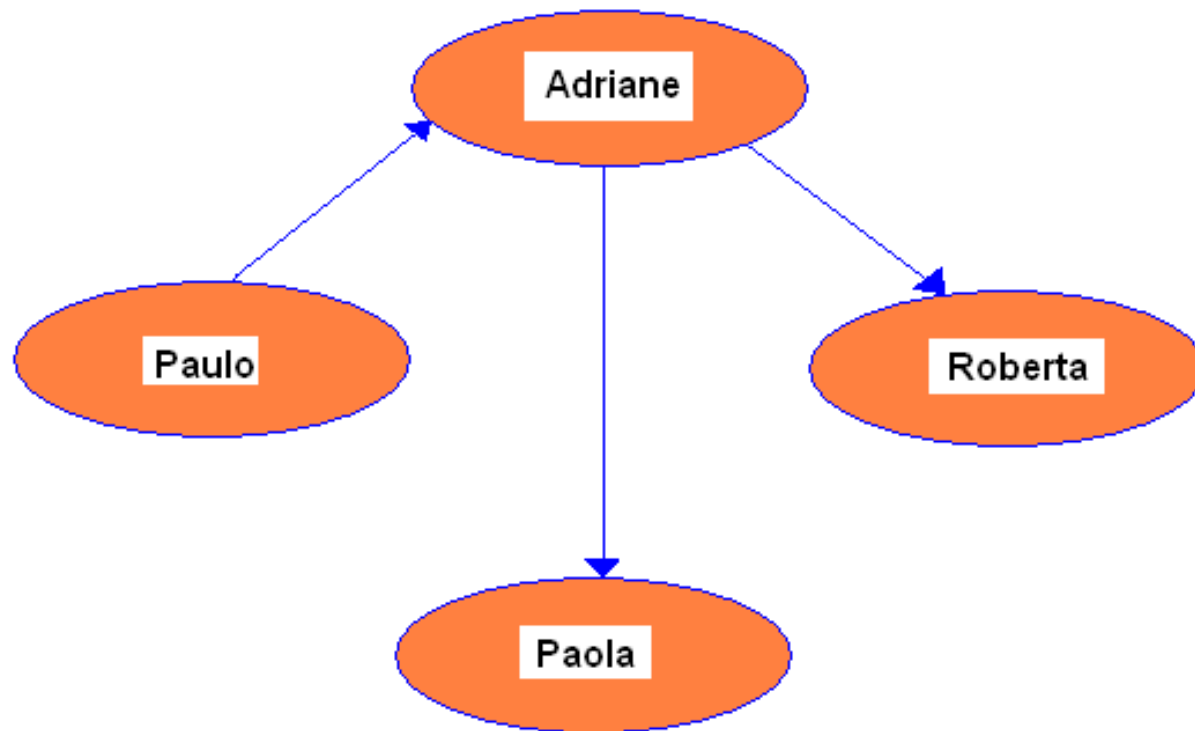
# Grafos

- Grafo não orientado

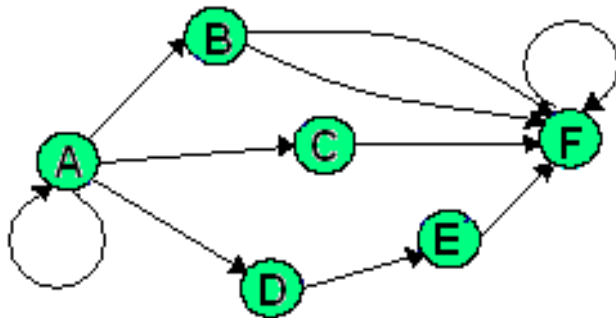


# Grafos

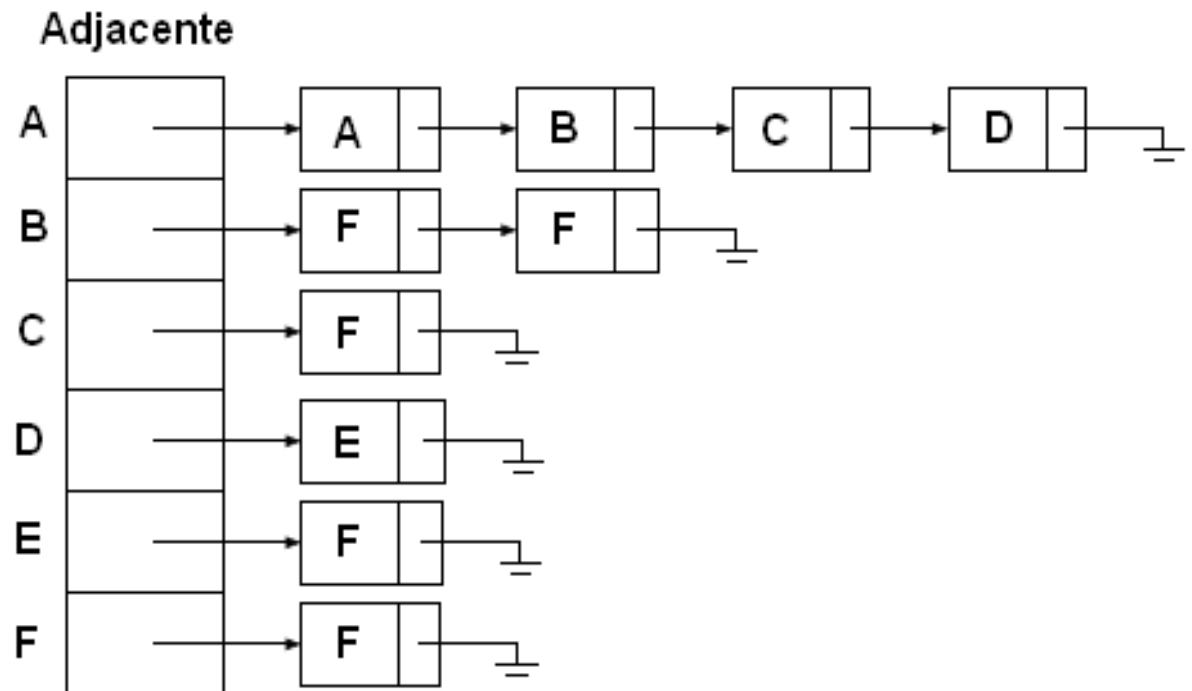
- Grafo orientado, dirigido ou dígrafo



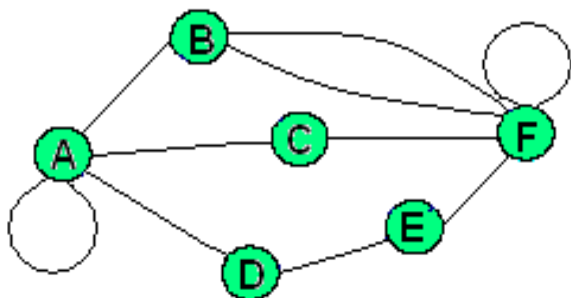
# Lista de adjacências



Adjacente[A] = [A, B, C, D]  
Adjacente[B] = [F, F]  
Adjacente[C] = [F]  
Adjacente[D] = [E]  
Adjacente[E] = [F]  
Adjacente[F] = [F]



# Lista de adjacências



Adjacente[A] = [A, B, C, D]

Adjacente[B] = [A, F, F]

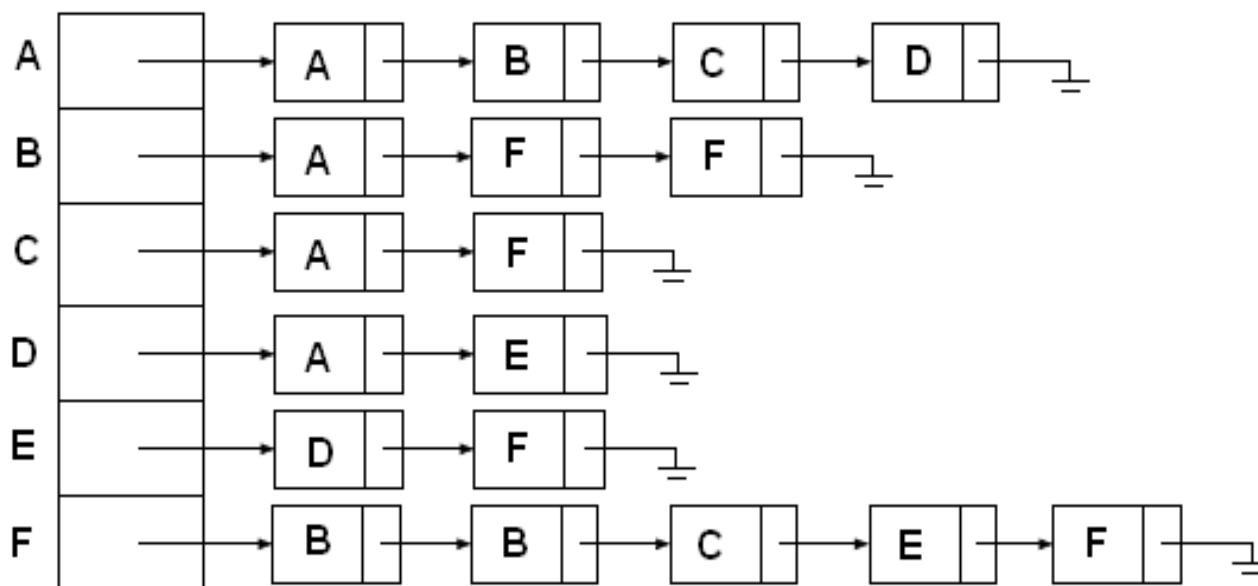
Adjacente[C] = [A, F]

Adjacente[D] = [A, E]

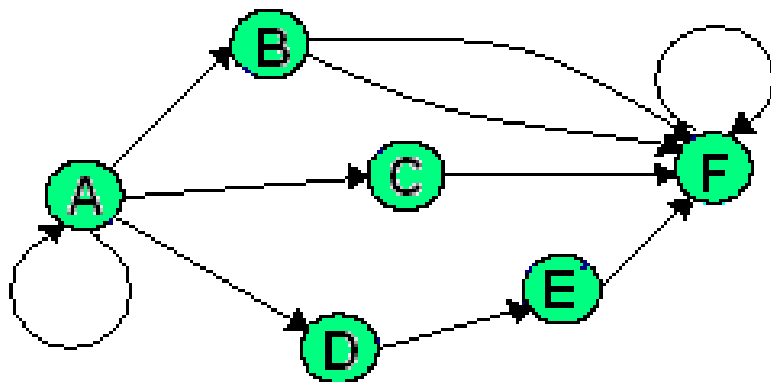
Adjacente[E] = [D, F]

Adjacente[F] = [B, B, C, E, F]

Adjacente



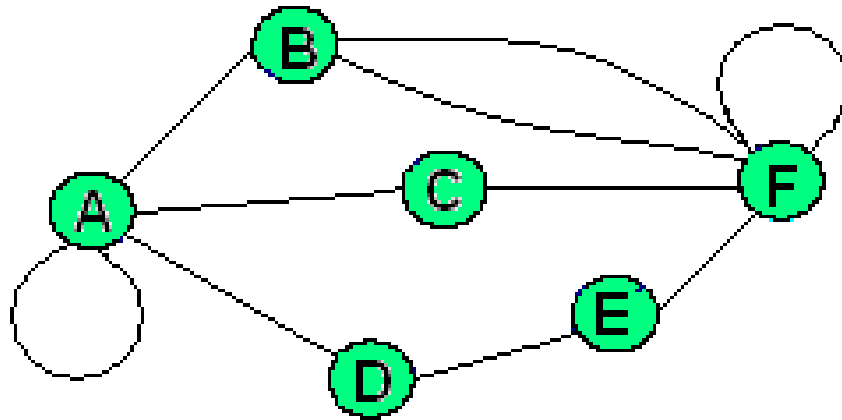
# Matriz de Adjacências



A

	A	B	C	D	E	F
A	1	1	1	1	0	0
B	0	0	0	0	0	2
C	0	0	0	0	0	1
D	0	0	0	0	1	0
E	0	0	0	0	0	1
F	0	0	0	0	0	1

## Matriz de Adjacências

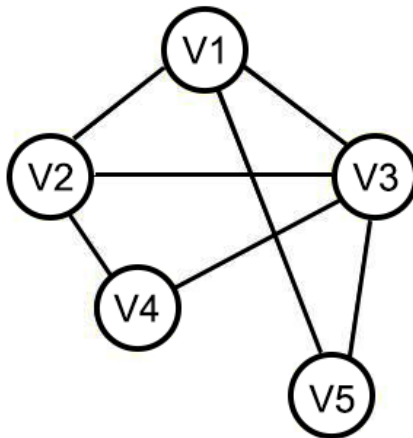


A

	A	B	C	D	E	F
A	1	1	1	1	0	0
B	1	0	0	0	0	2
C	1	0	0	0	0	1
D	1	0	0	0	1	0
E	0	0	0	1	0	1
F	0	2	1	0	1	1

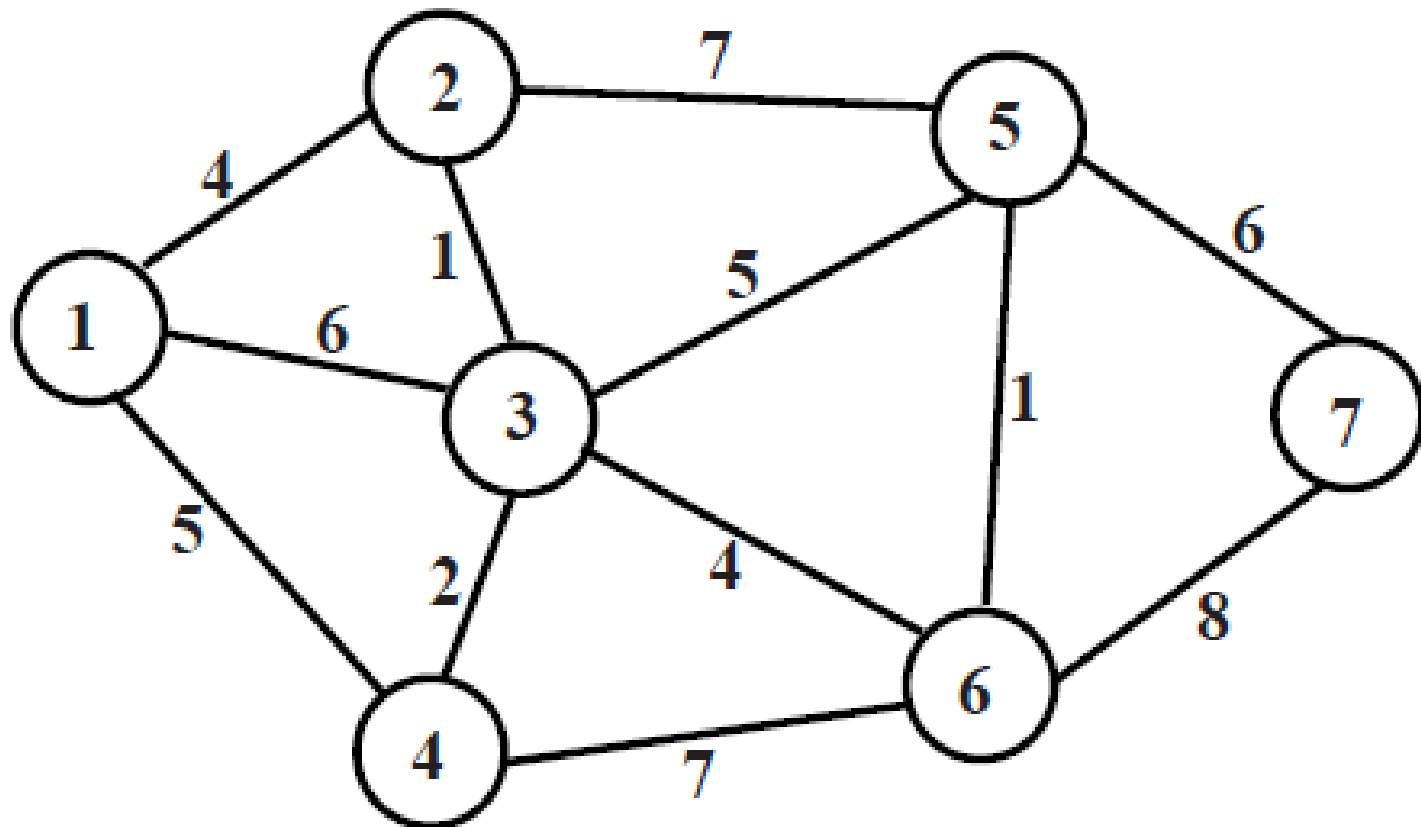


## Matriz de Adjacência

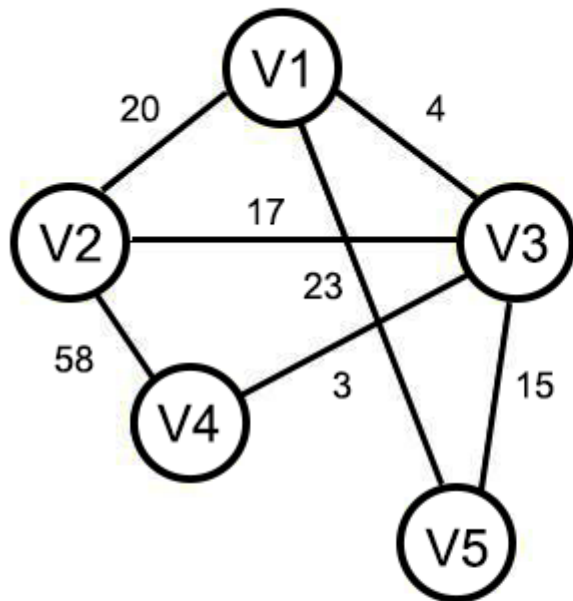


	V1	V2	V3	V4	V5
V1	0	1	1	0	1
V2	1	0	1	1	0
V3	1	1	0	1	1
V4	0	1	1	0	0
V5	1	0	1	0	0

# Grafos



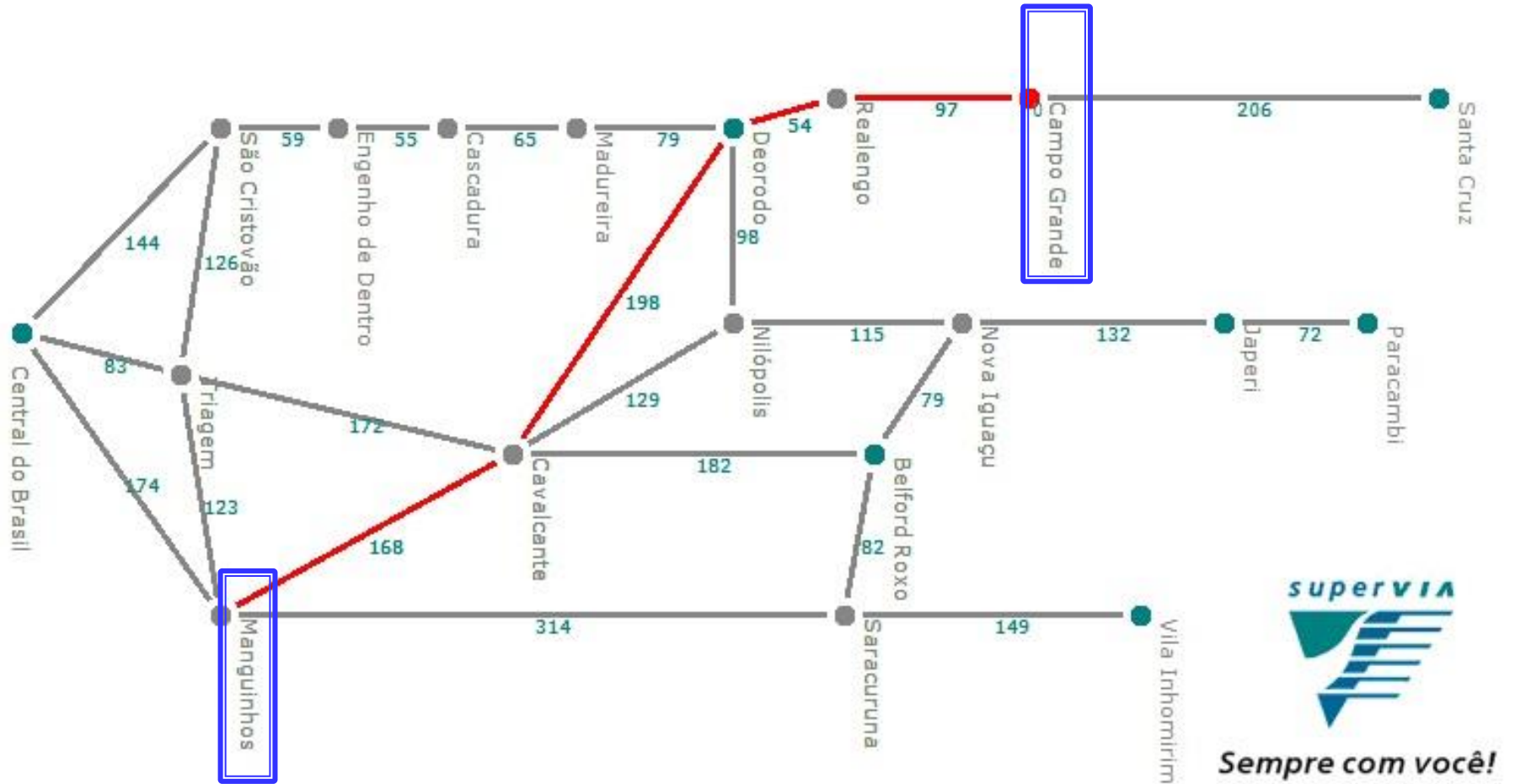
# Grafos



\*zero é um valor escolhido  
em código para considerar não ter  
nenhuma ligação entre os dois grafos,  
porém se seu grafos tiver zero como um  
valor valido deve se escolher outro valor.

	V1	V2	V3	V4	V5
V1	0*	20	4	0*	23
V2	20	0*	17	58	0*
V3	4	17	3	3	15
V4	0*	58	3	0*	0*
V5	23	0*	15	0*	0*

# Grafos



**Sempre com você!**

Origem

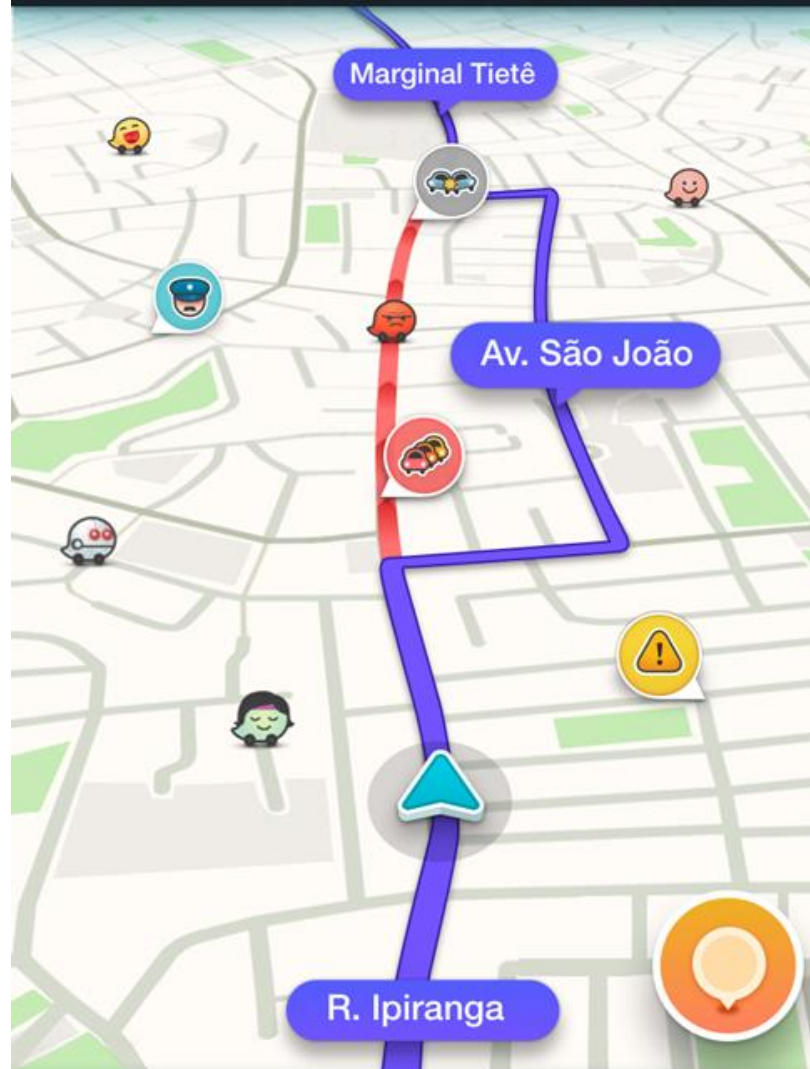
Destino





0.5 Km

Av. São João



18:22

22 min.  32 km

2

# Determinação do Menor Caminho

- o **caminho de um vértice a outro vértice é mínimo** se não existe outro caminho entre eles que tenha menos arcos.
- O problema de encontrar o caminho mais curto entre dois nós de um grafo é um dos problemas clássicos da Ciência da Computação.

Este problema consiste, genericamente, em **encontrar o caminho de menor custo entre dois nós da rede, considerando a soma dos custos associados aos arcos percorridos.**



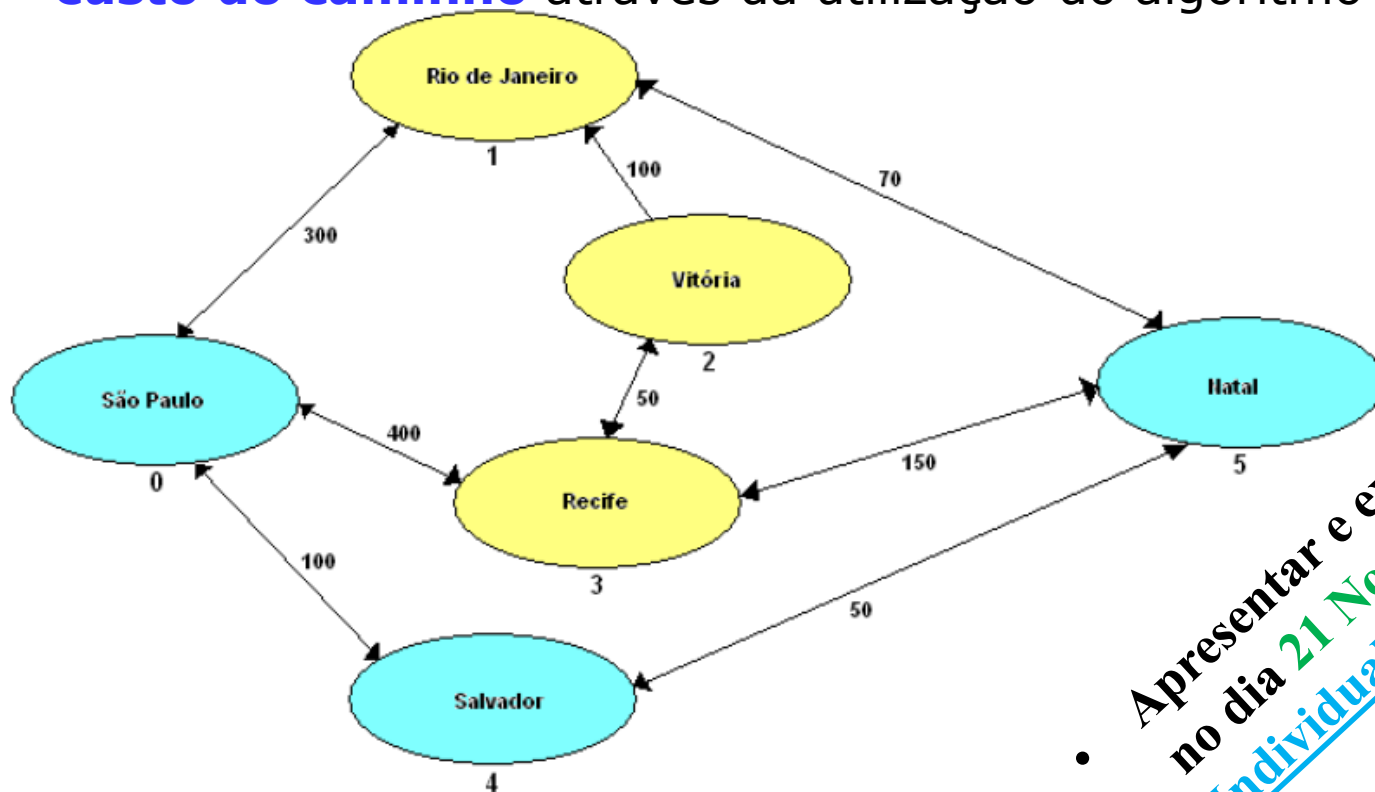
# Dijkstra

---

O mais famoso algoritmo para resolver o problema de **caminho mínimo** em grafos é o algoritmo de **Dijkstra**

# Avaliação 2 – Apresentar 21/Novembro

- Escreva um programa que cria um **grafo** representando a ligação entre seis cidades com suas respectivas distâncias (*São Paulo, Rio de Janeiro, Vitória, Recife, Salvador e Natal*).
- O programa deve perguntar ao usuário a cidade **origem** e a cidade **destino** e exibir o **caminho mínimo** entre estas duas cidades e o **custo do caminho** através da utilização do algoritmo de **Dijkstra**.



- Apresentar e explicar o algoritmo funcionando no dia **21 Novembro** Individual ou até 3 pessoas



# Muito Obrigada!

**Profª. Angela Abreu Rosa de Sá, Drª.**

---

*Contato: [angelaabreu@gmail.com](mailto:angelaabreu@gmail.com)*