



SISTEMAS DE INFORMAÇÃO

ENTERPRISE ANALYTICS AND DATA WAREHOUSING

PROFº FABIANO J. CURY MARQUES

ENTERPRISE ANALYTICS AND DATA WAREHOUSING

NOSQL GRAPH-ORIENTED



AGENDA

- ✕ Introdução
- ✕ Neo4j
- ✕ Hands On
- ✕ Referências



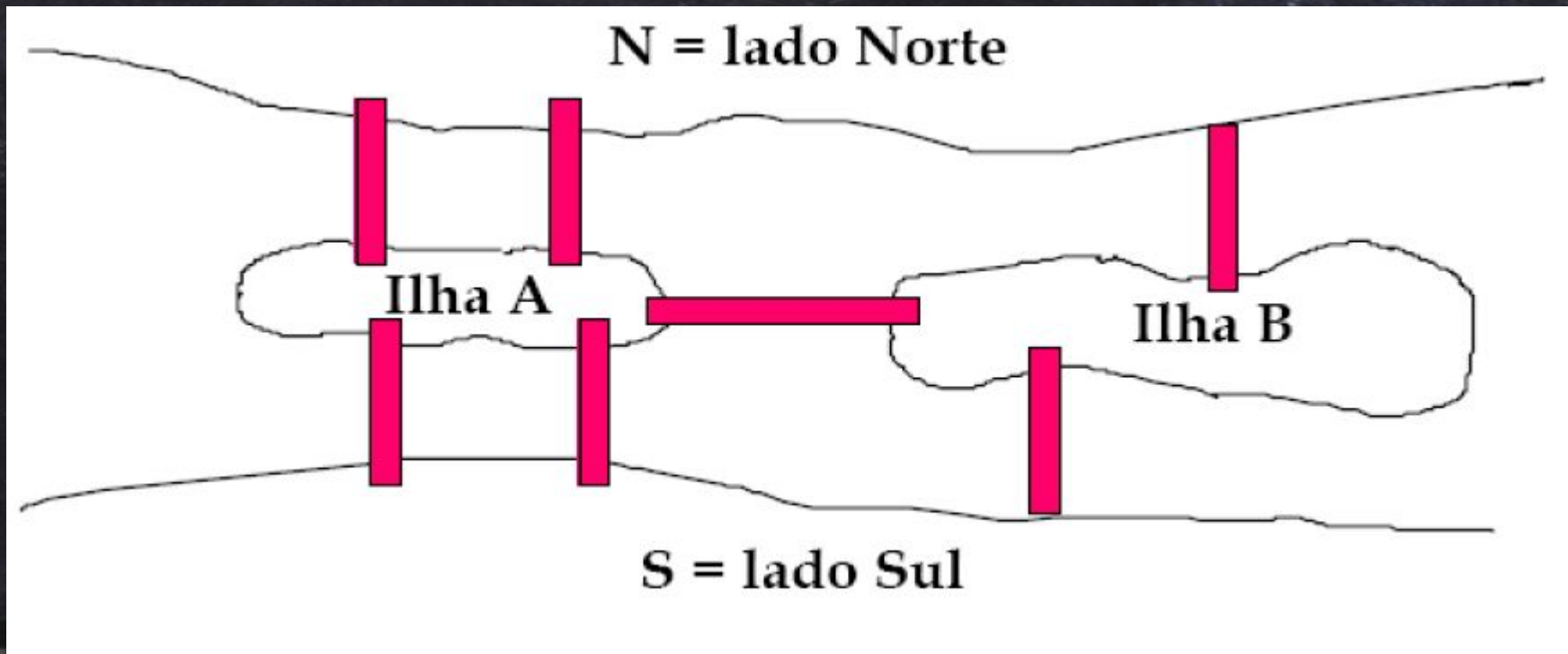
INTRODUÇÃO

Introdução

AS 7 PONTES DE KÖNIGSBERG



✕ Resolvido pelo matemático Leonhard Euler:



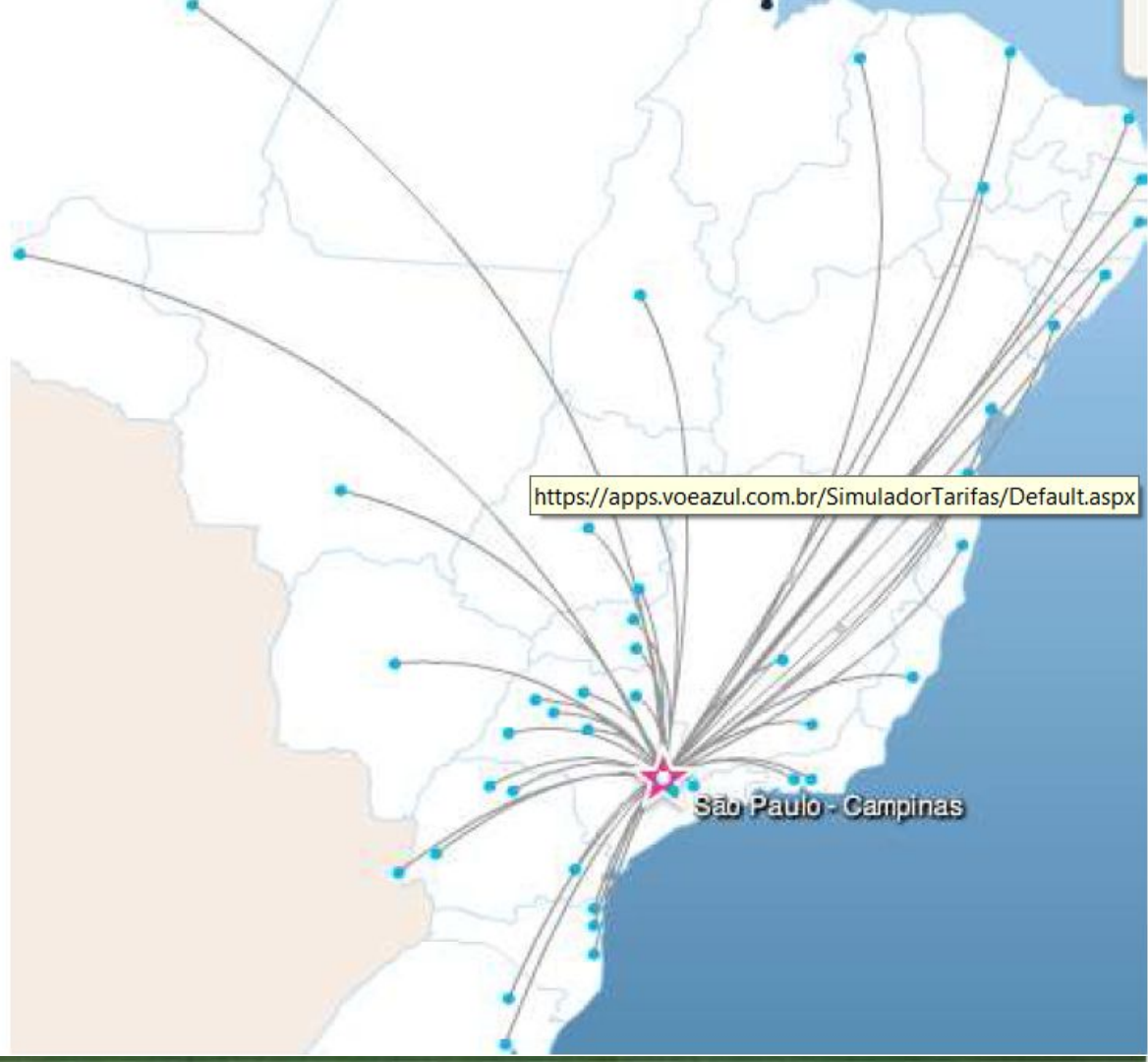


- ✗ Um grafo é um par (V, A) , onde V é um conjunto arbitrário e A é um conjunto de $V^{(2)}$;
- ✗ Os elementos de V são chamados de **Vértices**;
- ✗ Os elementos de A são chamados de **Arestas**;

VÔOS AÉREOS: AZUL LINHAS AÉREAS

Fonte:

<http://www.voeazul.com.br/>



VÔOS AÉREOS: TAM LINHAS AÉREAS

Fonte:
<http://www.tam.com.br/>



- Pontos de fiscalização
- Opções de local p/ base



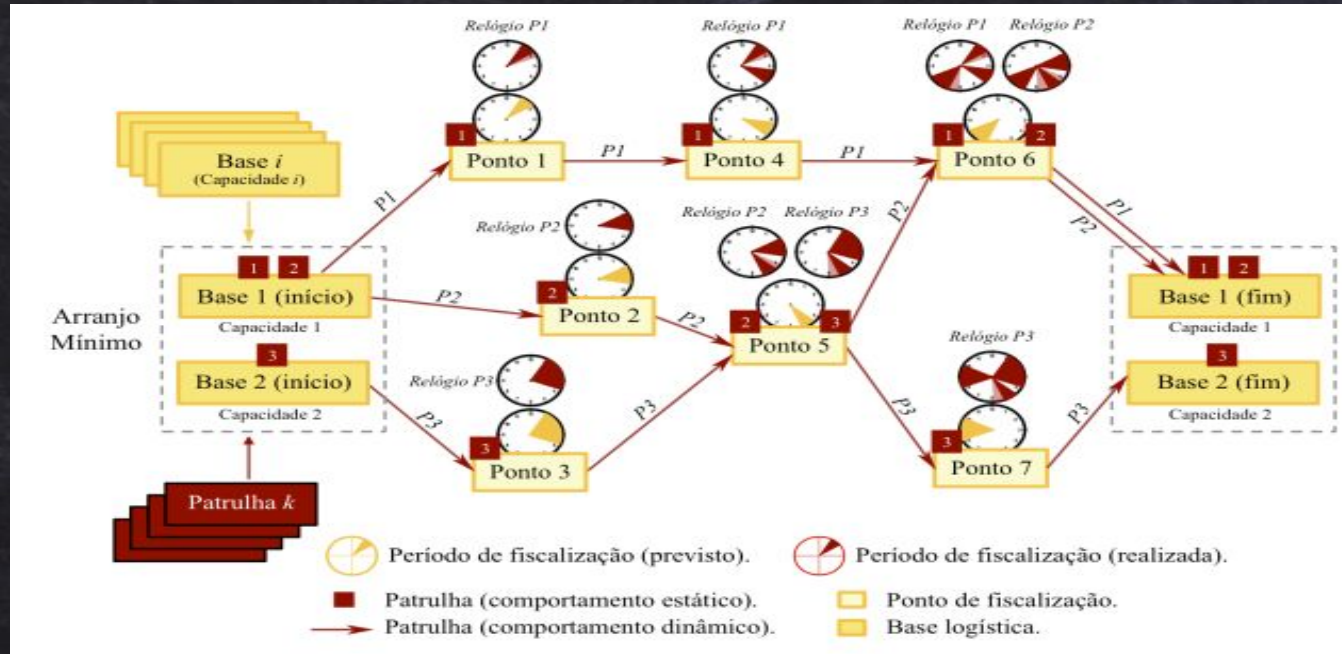
- Few and dispersed military bases; small garrisons.
- Logistical vacuum.
- High cost of military deployment from centers (rear).
- Low budget to perform military operations.



Tese de doutorado aprovada no ITA, sob orientação do Cel. Dr. Abrahão.

MODELAGEM MULTIOBJETIVO PARA DIMENSIONAMENTO DE RECURSOS EM SUPORTE A OPERAÇÕES DE FISCALIZAÇÃO EM FRONTEIRAS

PROBLEMA DE FISCALIZAÇÃO DE FRONTEIRAS DEFENDIDO COMO TESE DE DOUTORADO NO ITA/2018



GRAFOS PODE SER UTILIZADO PARA REPRESENTAR:



- ✕ Circuitos elétricos;
- ✕ Redes de distribuição de energia elétrica/água;
- ✕ Relação de parentesco entre pessoas;
- ✕ Redes sociais;
- ✕ Redes de estrada entre cidades/vãos
- ✕ Redes (físicas e lógicas) de computador;
- ✕ Navegabilidade de Páginas da web;

GRAFOS DEFINIÇÃO FORMAL



- X Um grafo G é uma tripla ordenada $G = (V, A, \varphi)$, onde
- X V é um conjunto finito e não vazio de vértices;
- X E é um conjunto finito de arestas;
- X $\varphi : E \rightarrow V \times V$ é uma **função de incidência** que associa a cada aresta e de

GRAFOS DEFINIÇÃO FORMAL



- X A aresta e é **incidente** em u e v ;
- X O par de vértices u e v são adjacentes;
- X Arestas **paralelas** ou **múltiplas** são arestas diferentes que compartilham os mesmos extremos;

GRAFO COMPLETO



✖ Um grafo é completo quando todos os seus vértices forem adjacentes.

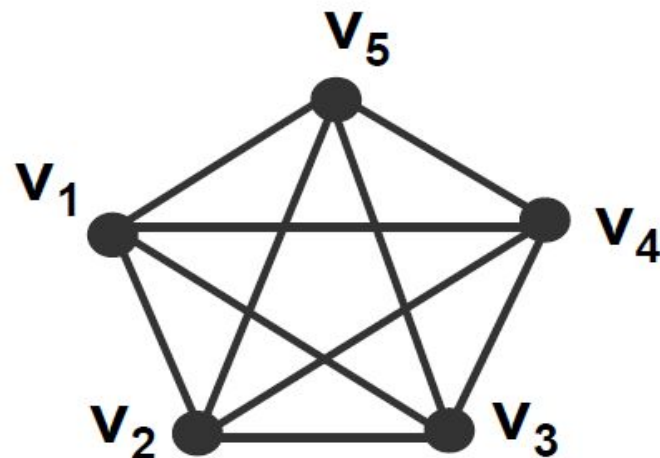
Um grafo completo possui $\frac{n(n-1)}{2}$ arestas;

Exemplo:

$$V = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5), \\ (v_2, v_3), (v_2, v_4), (v_2, v_5), \\ (v_3, v_4), (v_3, v_5), (v_4, v_5)\}$$

$$|V| = 5 \text{ e } |E| = 5(5-1)/2 = 10$$





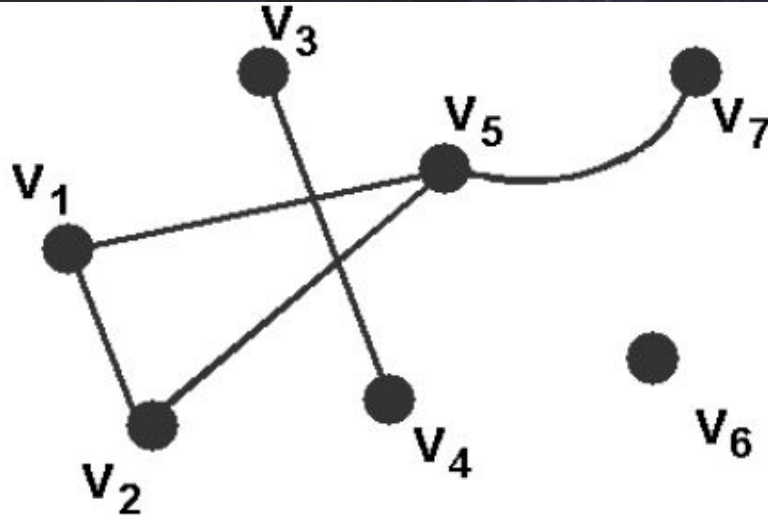
- ✗ O grau $d(v)$ de um vértice v corresponde ao números de vértices adjacentes a v (ou ao número de arestas incidentes a v):

$$d(v_6) = 0$$

$$d(v_3) = d(v_4) = d(v_7) = 1$$

$$d(v_1) = d(v_2) = 2$$

$$d(v_5) = 3$$



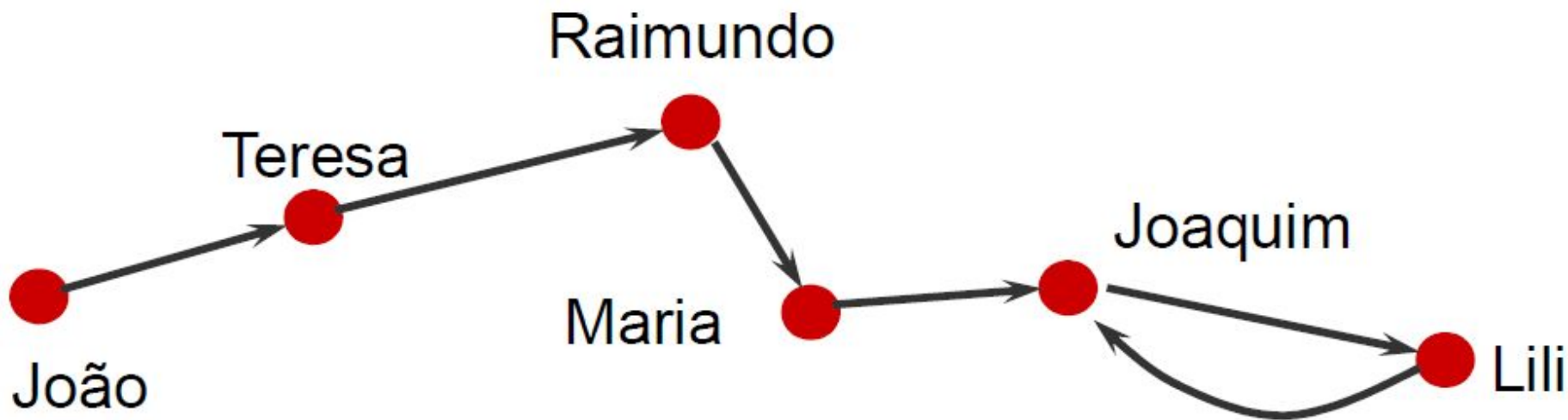
GRAFOS – CICLO EULERIANO/HAMILTONIANO



- ✗ Um **grafo é conexo** se existir caminho entre quaisquer dois vértices;
- ✗ Um grafo G conexo, possui **ciclo Euleriano** se e somente se todo vértice de G possuir grau par; (resolução para o problema de Königsberg)
- ✗ Um grafo G conexo é hamiltoniano se existir um ciclo que inclui todo vértice de G (**ciclo hamiltoniano**);



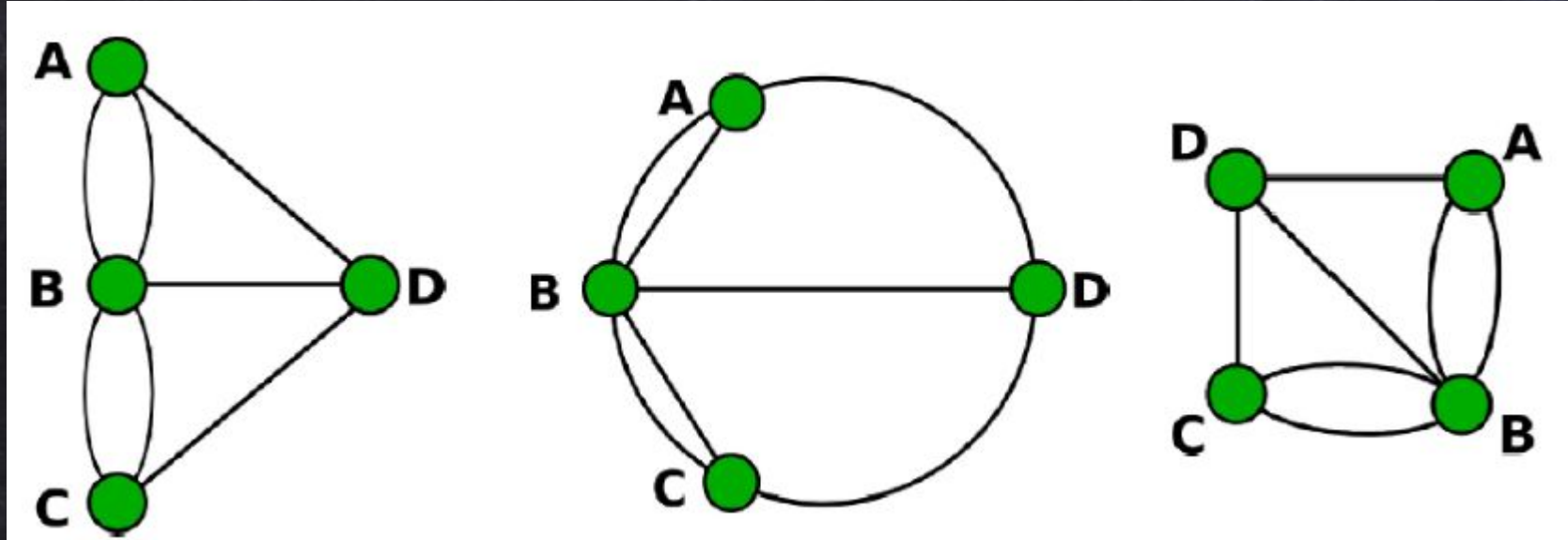
- ✕ João amava Teresa que amava Raimundo que amava Maria que amava Joaquim que amava Lili que não amava ninguém .. (Carlos Drumond de Andrade):



GRAFOS VS GRÁFICO



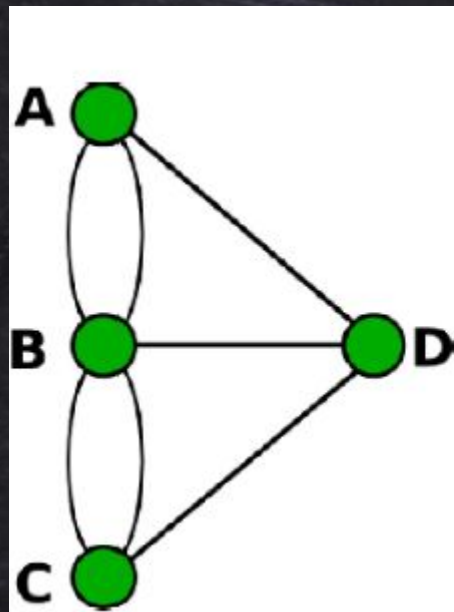
✕ Um grafo pode ser representado graficamente de diversas formas:



GRAFOS VS TABELA



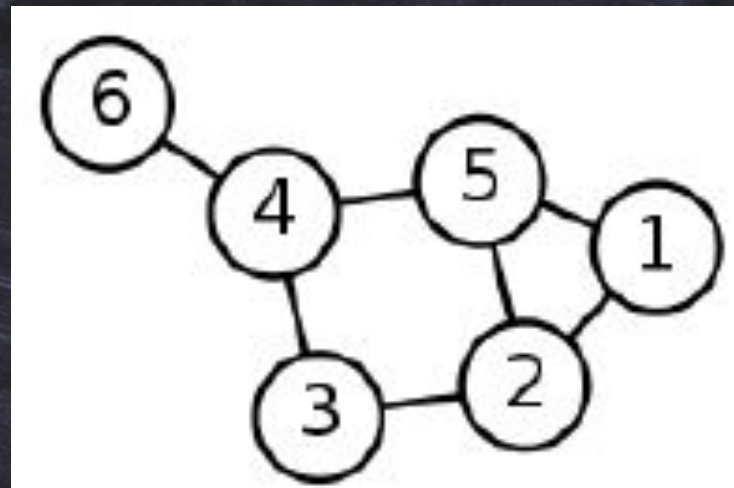
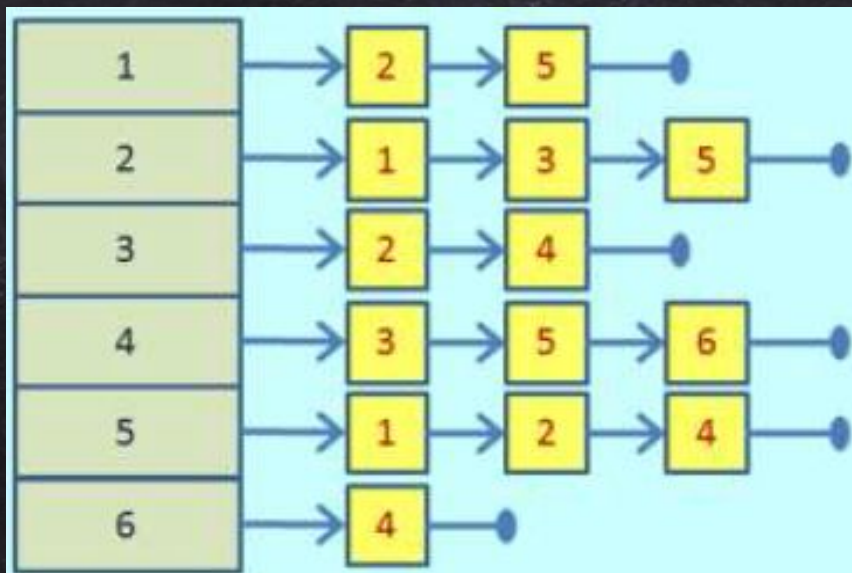
✕ Um grafo pode ser representado também no formato de tabelas:



	A	B	C	D
A	0	2	0	1
B	2	0	2	1
C	0	2	0	1
D	1	1	1	0



REPRESENTAÇÃO COMPUTACIONAL: LISTA DE ADJACÊNCIAS

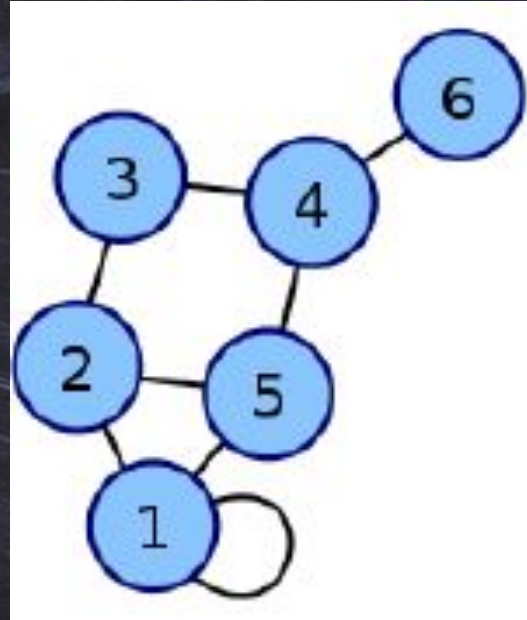


- ✗ Vantagem: espaço. Número de elementos é $|A| + |V|$;
- ✗ Desvantagem: verificar se uma aresta existe em G obriga percorrer a lista;

REPRESENTAÇÃO COMPUTACIONAL: MATRIZ DE ADJACÊNCIAS



x

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$


2.

DB NEO4J

Banco de dados de armazenamento de Grafos, Neo4j

DB PARA ARMAZENAMENTO DE GRAFOS



- ✗ Atualmente, em muitos casos a informação está associada mais ao relacionamento entre diferentes objetos do que ao objeto em si;
- ✗ Os banco de dados relacionais suportam o armazenamento de volume razoável de dados, mas não foram projetados para lidar com os relacionamentos de forma eficiente;
- ✗ Diferentemente dos banco de dados relacionais, os DBs de armazenamento de grafos foram projetados para lidar com conexões e relacionamentos;
- ✗ DBs baseados de armazenamento de grafos pode ser utilizados em aplicações OLTP;

RDBMS vs GRAPH DATABASE



RDBMS	Graph Database
Tables	Graphs
Rows	Nodes
Columns and Data	Properties and its values
Constraints	Relationships
Joins	Traversal

DB NEO4J – INTRODUÇÃO



- ✗ **Neo4j** é um Banco de dados para armazenamento de grafos;
- ✗ Escrito em linguagem de programação java, permite a manipulação de seus dados utilizando a linguagem **CQL** (Cypher Query Language);
- ✗ O modelo de dados de um banco de dados é mais simples quando comparado com modelos de dados de banco de dados relacionais
- ✗ Baseado em acessos do tipo REST API (Protocolo HTTP)

DB NEO4J – INTRODUÇÃO



- ✗ O uso do DB Neo4j envolve o uso dos conceitos: Nós, Propriedades, Relacionamentos e Labels.
- ✗ **Nó** – Unidade fundamental de um Grafo. Contém um conjunto de propriedade

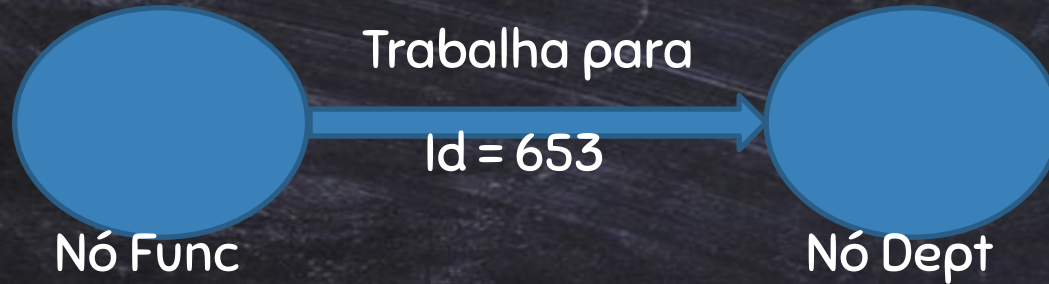
Número : 123
Nome : “José”
Salário :
2850,00

Nó Funcionário

DB NEO4J – INTRODUÇÃO



- ✗ **Propriedades** – É um par chave-valor utilizado nos nós e em seus relacionamentos;
- ✗ **Relacionamento** – Utilizados para conectar dois nós;



- ✗ **Label** – Utilizado para associar um nome comum à um nó ou relacionamento. Um mesmo nó ou relacionamento pode ter um ou mais labels;

INSTALANDO O DB NEO4J



✗ Podemos baixar a máquina virtual do vagrant abaixo:

```
Windows PowerShell

C:\>mkdir vagrant_neo4j

C:\>cd vagrant_neo4j

C:\vagrant_neo4j>vagrant init phonetworks/xenial-pho
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.

C:\vagrant_neo4j>vagrant up
```

INSTALANDO O DB NEO4J



- ✕ Podemos utilizar o console online acessando o site:

<https://console.neo4j.org/>

- ✕ Trata-se de um console online que nos permite executar alguns comandos para simular a utilização do DB Neo4j;

DB NEO4J – COMANDOS CQL



- ✗ Criação de um nó:

```
CREATE (no_empregado);
```

- ✗ Criação de muitos nós:

```
CREATE (no1), (no2), (no3)
```

- ✗ Consulta de todos os nós:

```
MATCH (n) RETURN n
```



✖ Criação de um nó funcionário:

Nó



Propriedades



```
CREATE (JOAO:funcionario{name: "João da Silva", Nasc: 1982})
```

Label



Chave-valor



Chave-valor





✕ Criação de nós com relacionamento:

```
CREATE (JOAO:funcionario{name: "João da Silva", AnoNascimento: 1982, Cidade: "Jundiai"})  
CREATE (PEDRO:funcionario{name: "Pedro da Silva", AnoNascimento: 1985, Cidade: "Jundiai"})  
CREATE (JOAO)-[r:IRMAO]->(PEDRO)
```




- ✕ Existem vasta documentação associada ao DB neo4j em:

<https://neo4j.com/developer/get-started/>

Beginning Neo4j

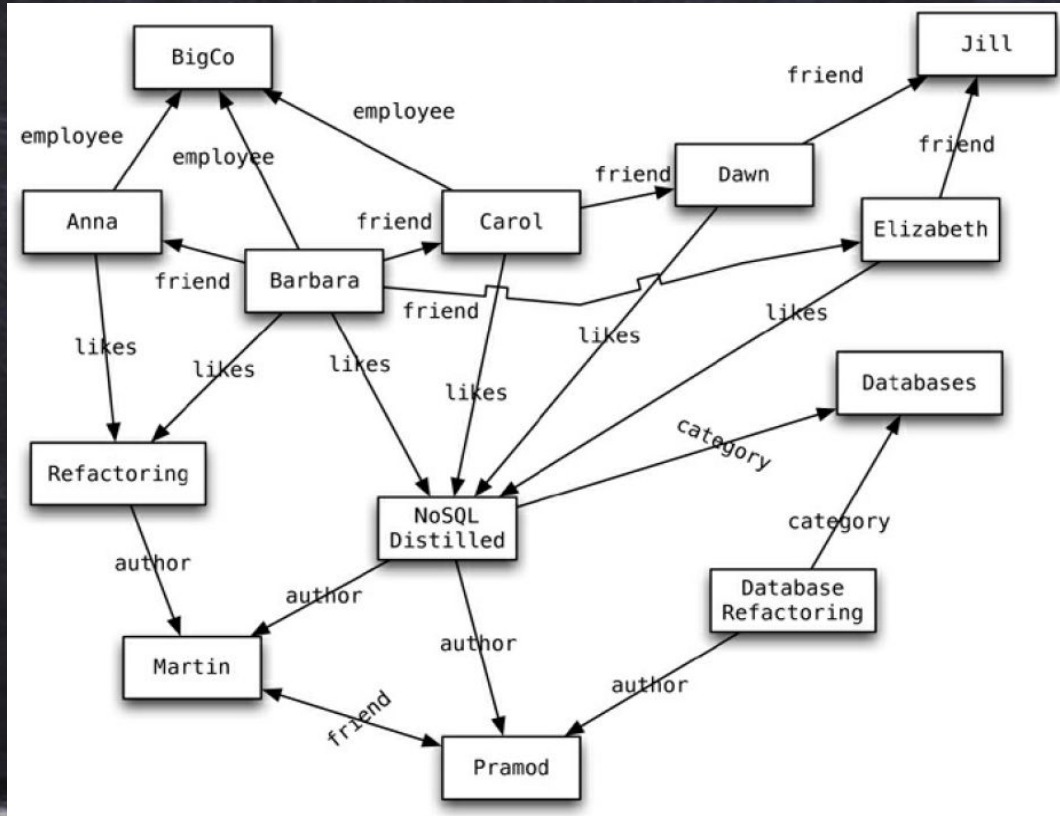
Create relationships and grow your application with Neo4j

—
Chris Kemper

apress®

ATIVIDADE

✕ Utilizando o Neo4j, construa o grafo abaixo:





REFERÊNCIAS

- ✕ P. O. Boaventura Netto. Grafos: Introdução e prática. Editora Blusher. São Paulo, 2009;
- ✕ P. O. Boaventura Netto. Grafos: Teoria, modelos algoritmos. 5° ed. Editora Blusher. São Paulo, 2012;
- ✕ M. C. Goldbarg; E. Goldbarg. Grafos: Conceitos, algoritmos e aplicações. Rio de Janeiro: Elsevier, 2012;
- ✕ Gaurav Vaish. Getting Started with NoSQL, packt publishing, 2013.



OBRIGADO!

Copyright © 2019 Prof. MSc. Eng. Wakim B. Saba

<https://br.linkedin.com/in/wakimsaba>

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).