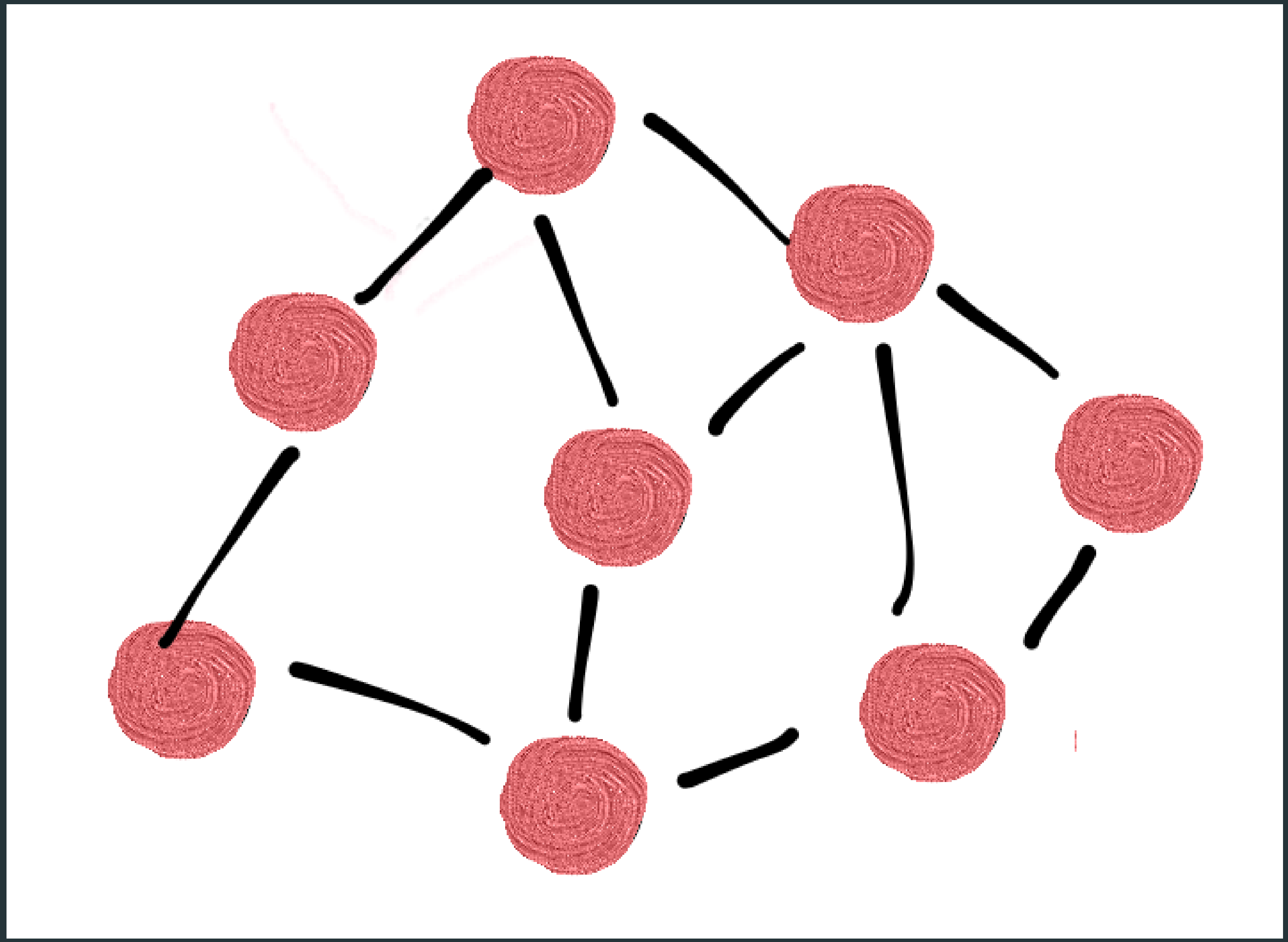
grafos

Faculdade de Computação - UFU
Estruturas de Dados 2
Maria Adriana Vidigal de Lima



Tópicos importantes

Sumário

Estruturas Lineares X Não Lineares

Motivação

Aplicações

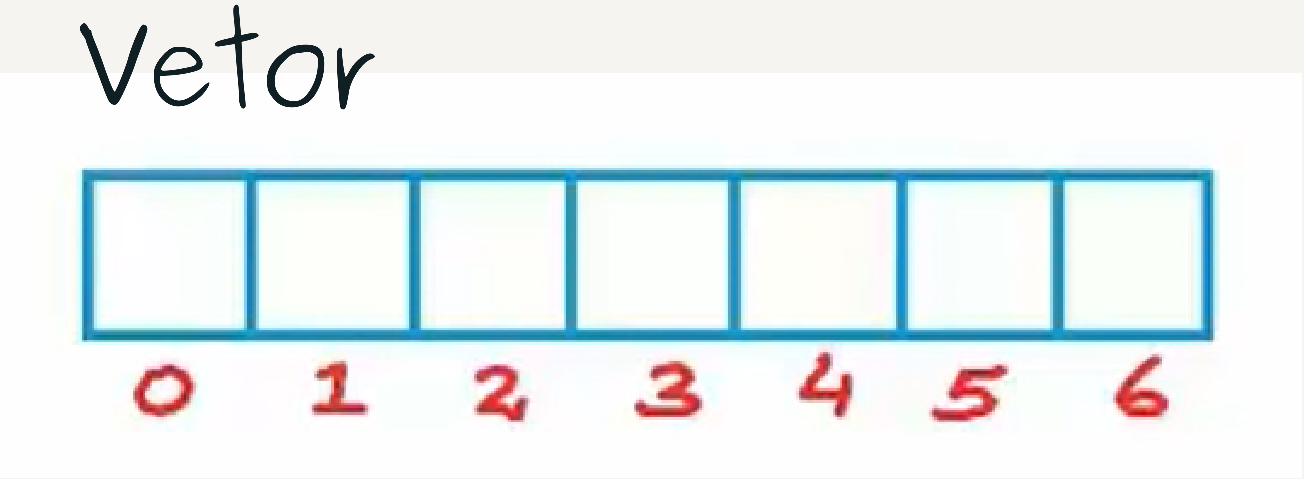
Conceitos Básicos

- Grafos Direcionados e Não Direcionados
- Grau de um Vértice e Caminhos
- Ciclos em grafos

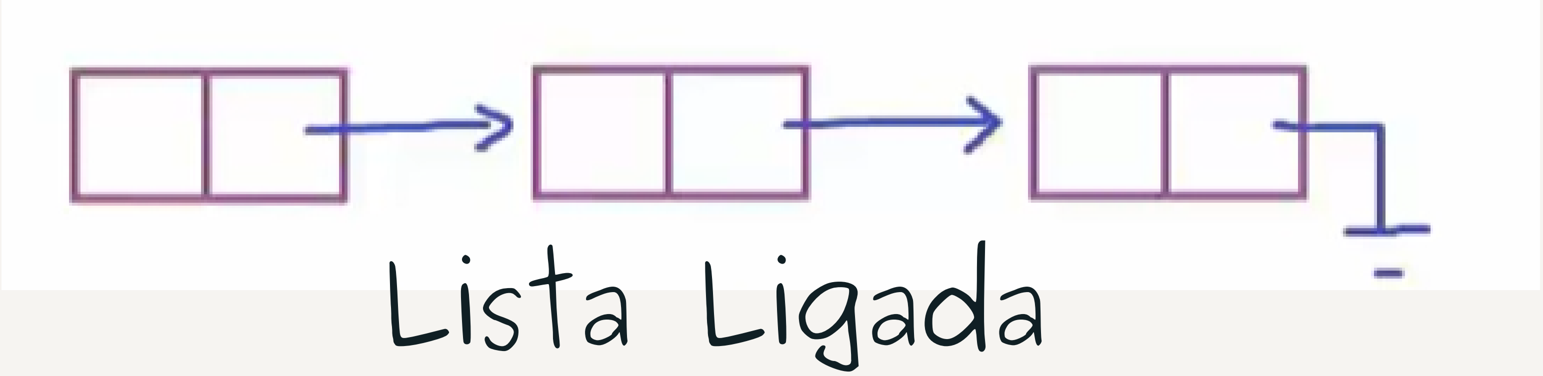
Grafo ponderado

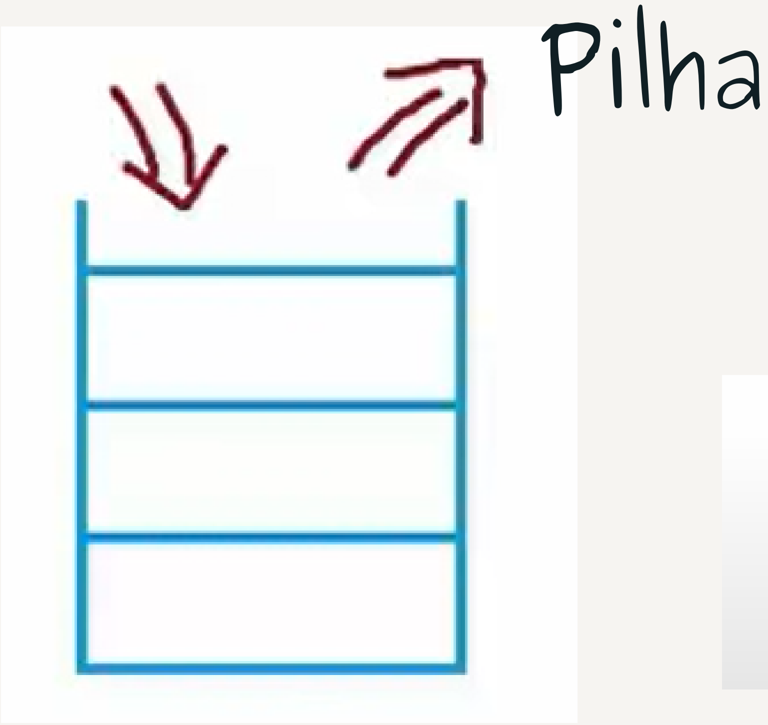
Grafos Variantes no Tempo

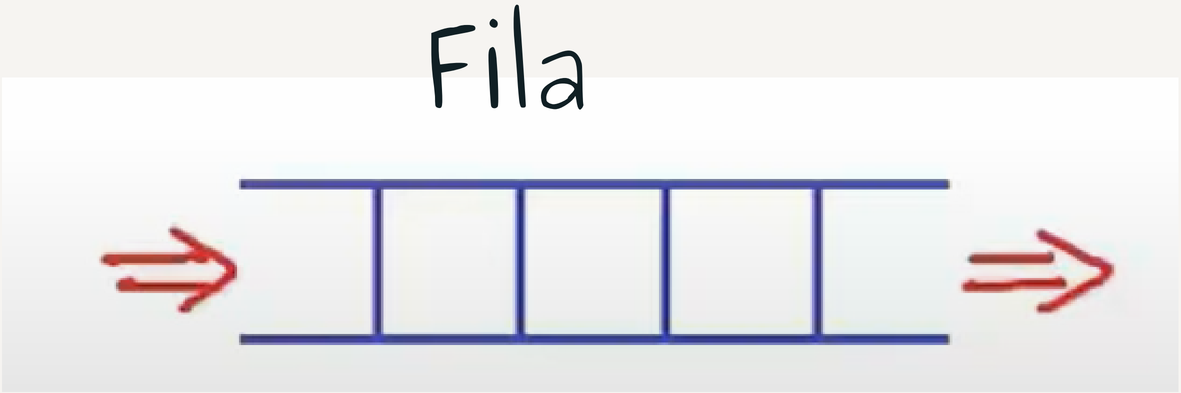
Estruturas de Dados Lineares



Estruturas de dados representam maneiras de armazenar e organizar dados.

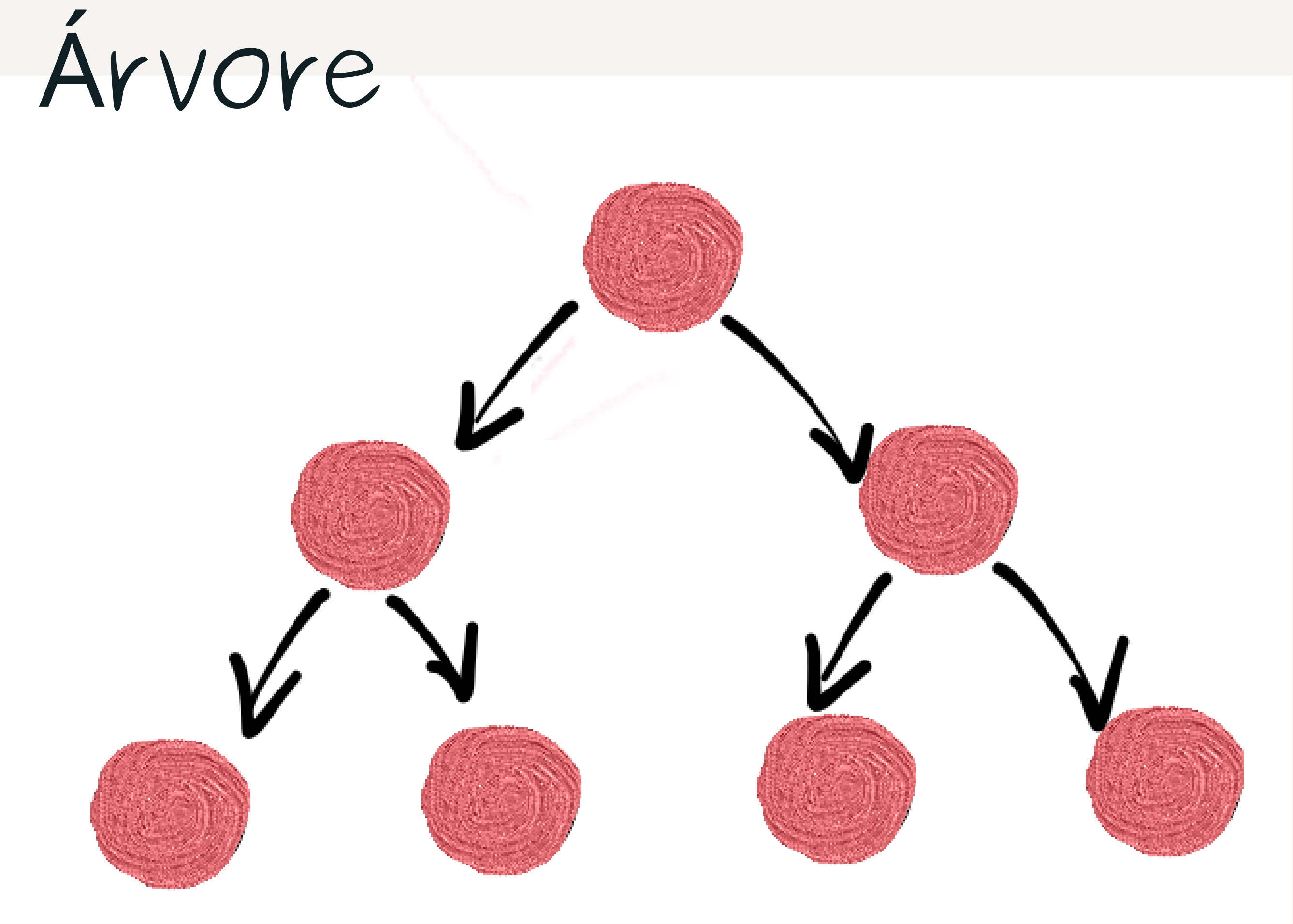


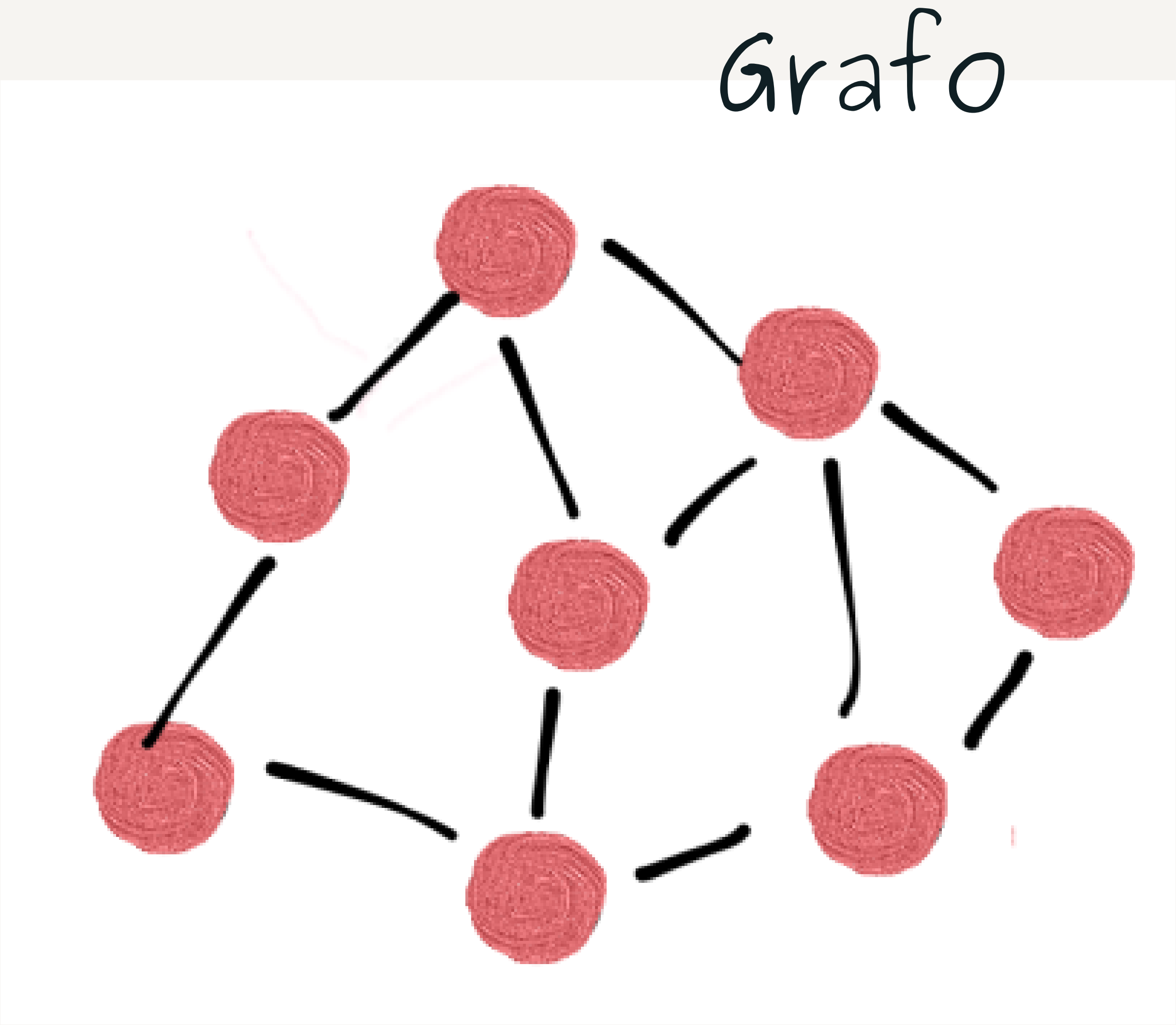




Em todas estas estruturas, os dados são organizados **linearmente**, de forma sequencial. São, portanto, estruturas de dados lineares.

Estruturas de Dados Não Lineares





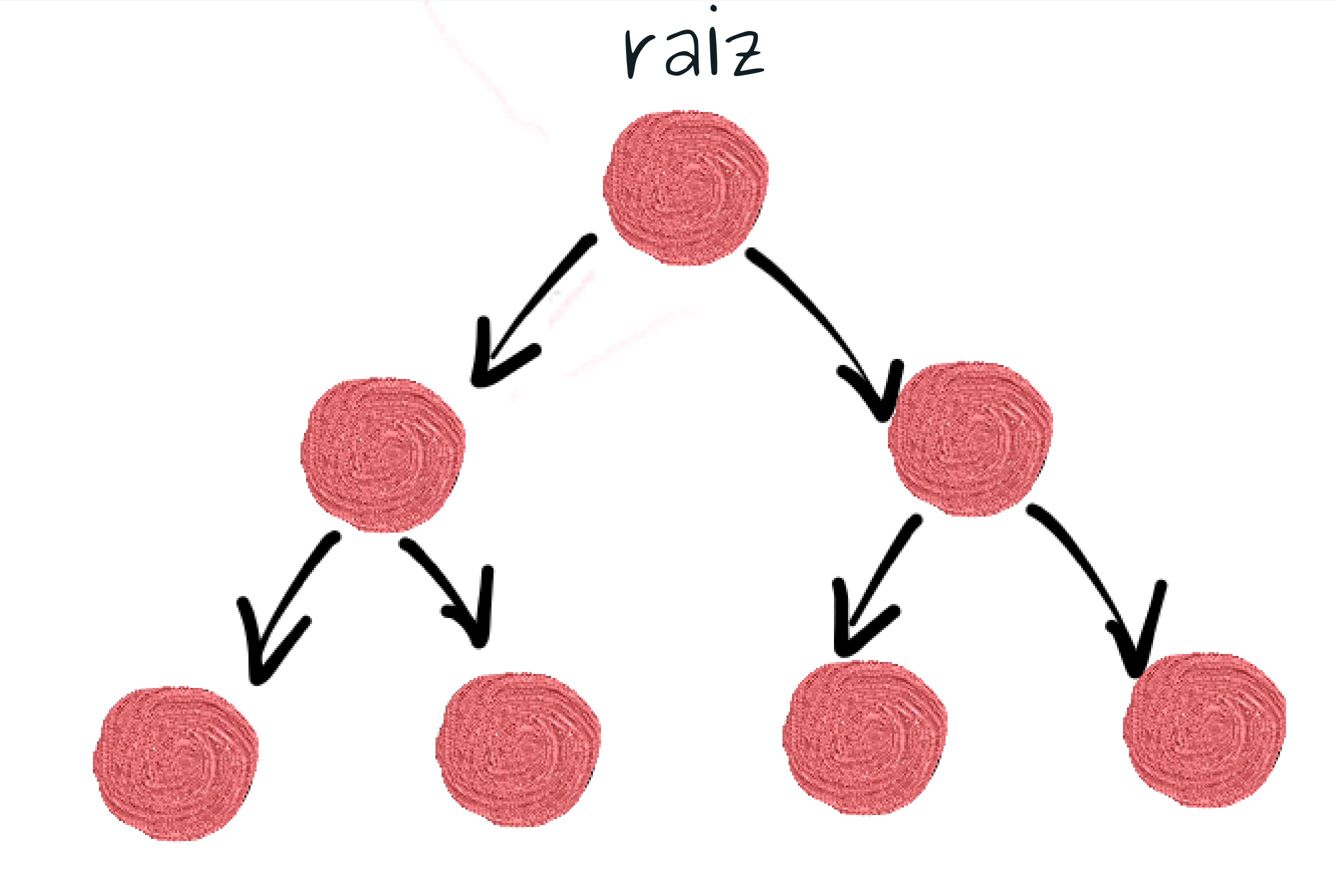
Quando existe mais de um caminho possível para navegar pela estrutura, esta é dita não linear. Exemplos clássicos deestruturas deste tipo são as **árvores** e **grafos**.

Estruturas de Dados Lineares X Não Lineares

	LINEAR	NÃO LINEAR
Definição	Os dados estão organizados de forma sequencial e são acomodados um após o outro.	Os dados não organizados em sequência, um elemento pode se conectar à qualquer outro para indicar um relacionamento específico.
Percurso	É possível passar por todos os elementos em uma única travessia.	Não é possível passar por todos os elementos em uma única travessia.
	implementação simples	*implementação complexa*

Estruturas de Dados Não Lineares

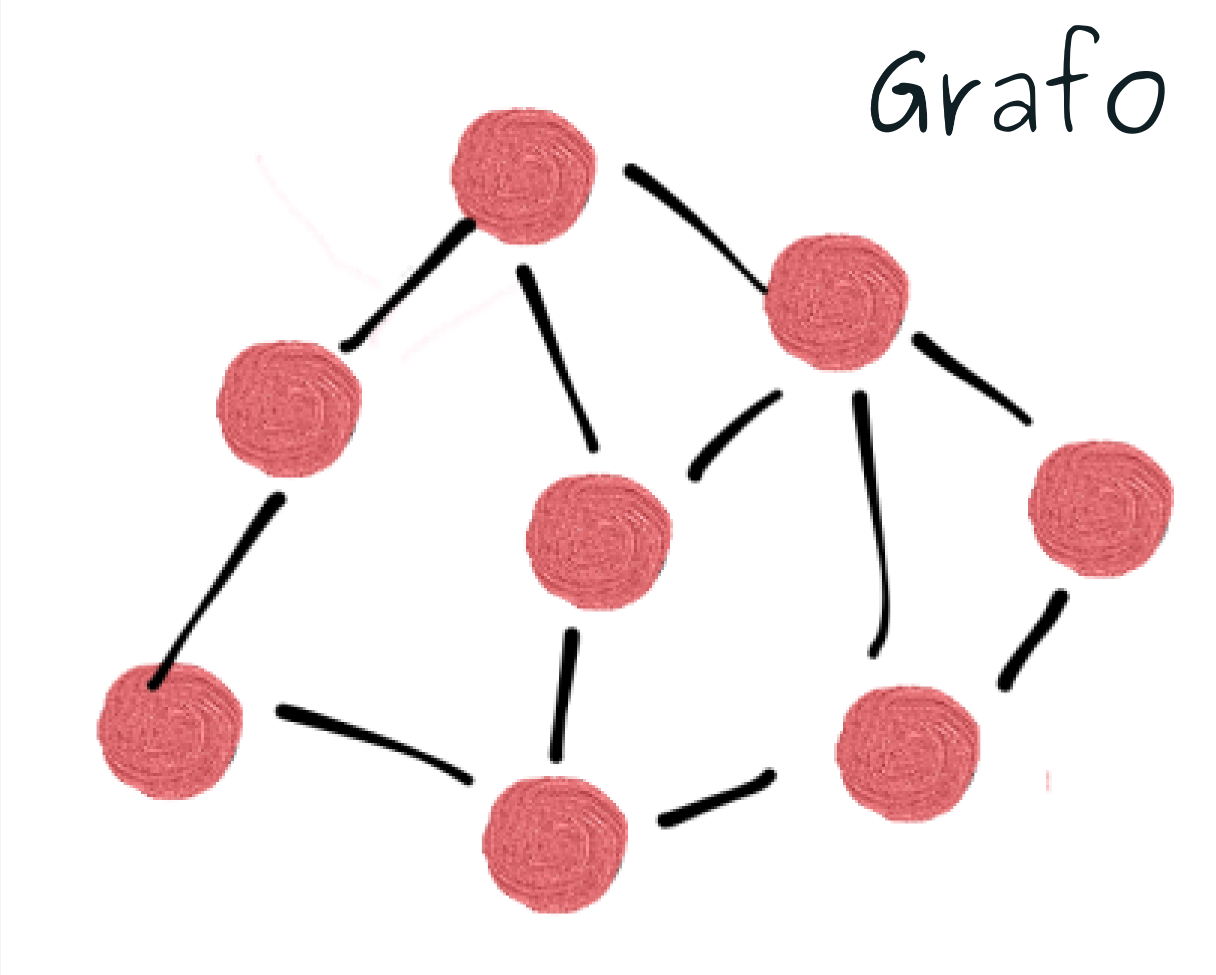
Árvore



Árvores começam com um nó **raiz** que pode ou não se conectar a outros nós. Assim, todos os caminhos partem de um ponto em específico.

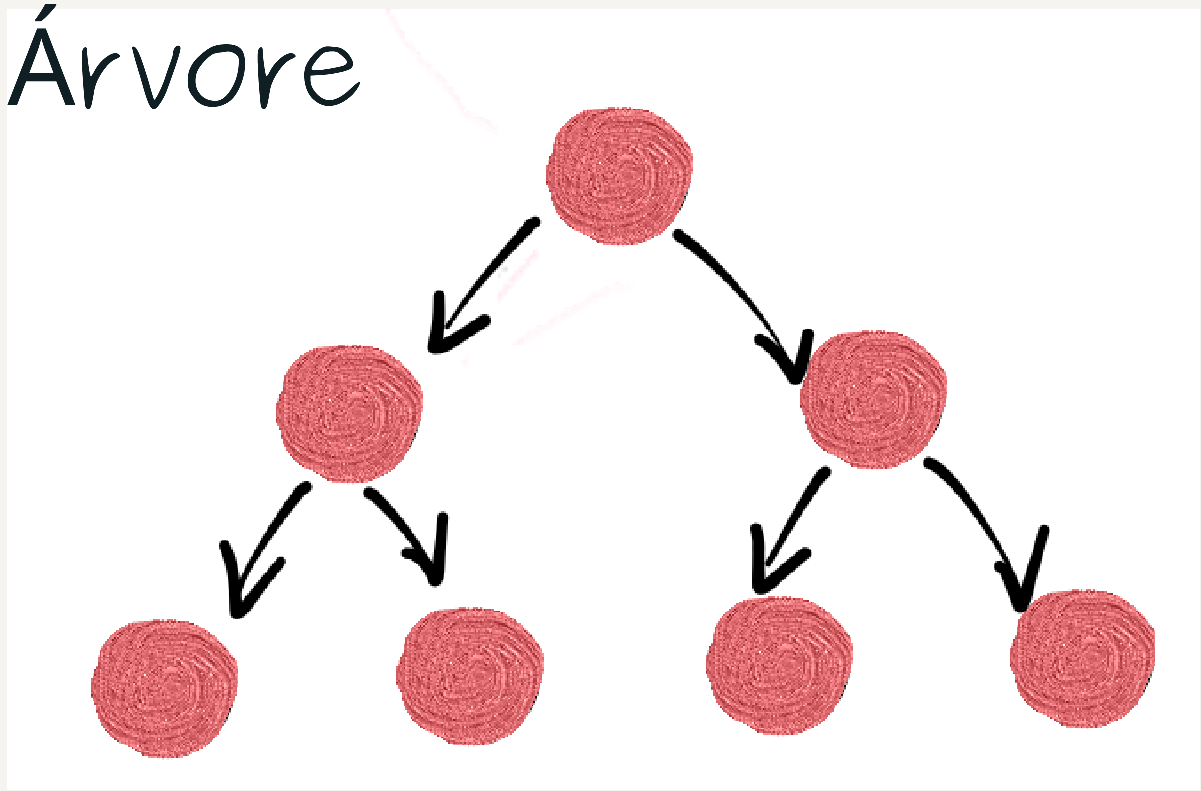
Estruturas de Dados Não Lineares

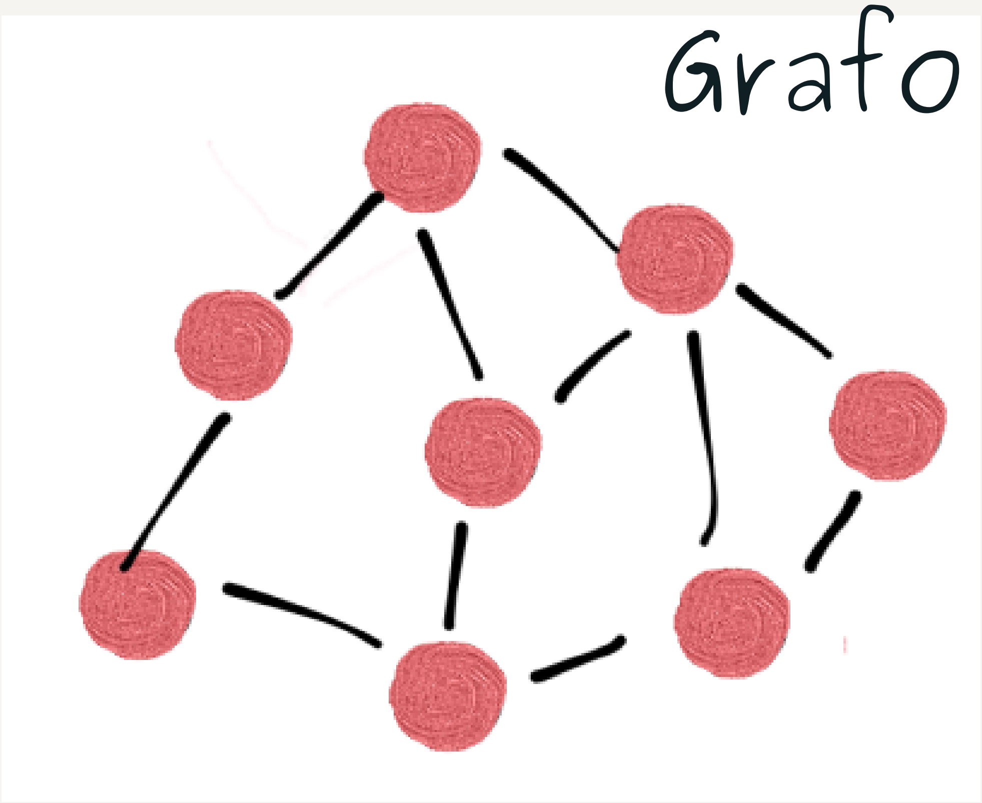
Grafos são estruturas que consistem em **vértices** e **arestas** que ligam estes vértices.



Podem representar espécies diferentes em um nicho ecológico e suas ligações, identificar quem influencia quem em uma organização, modelar resultados em torneios esportivos, mapear colaboração entre pesquisadores, identificar links entre websites, retratar mapas rodoviários, e serem úteis em diversos outros problemas.

Estruturas de Dados Não Lineares





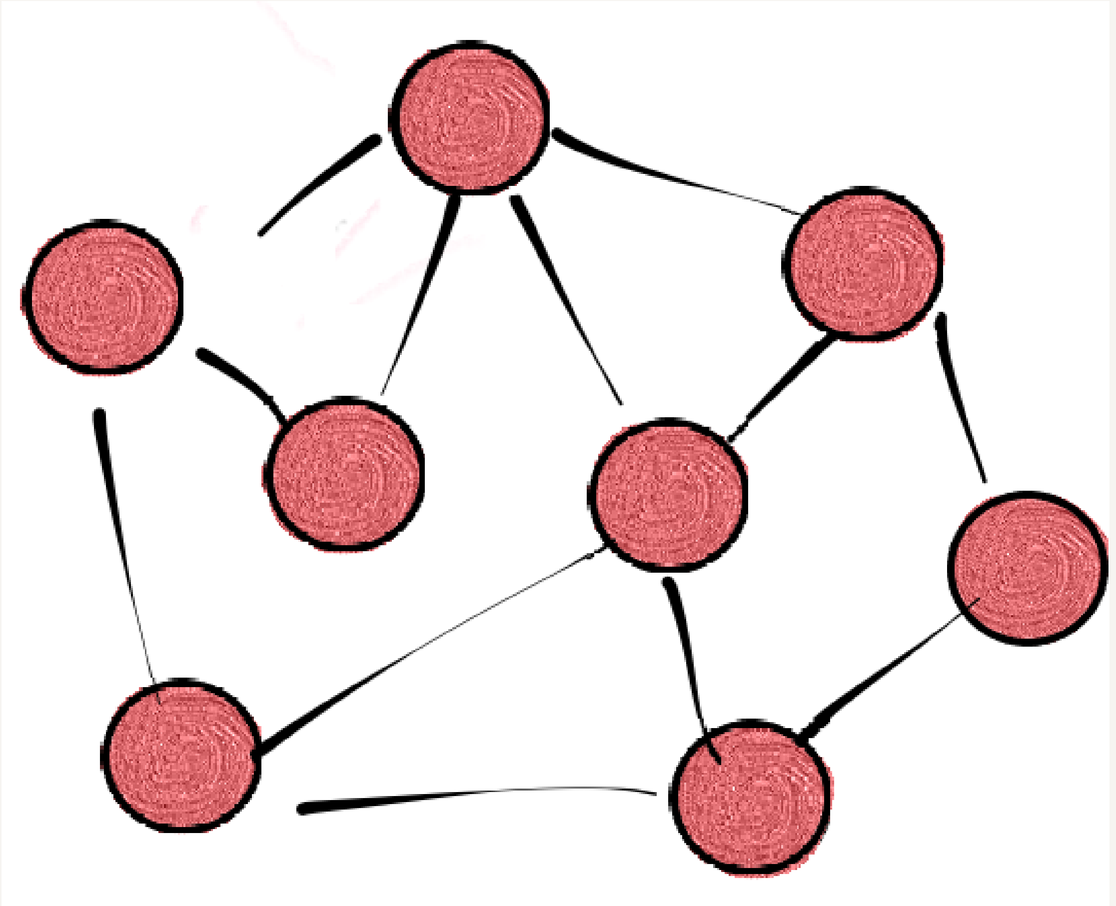
- N nós => (N-1) ligações
- Uma única ligação para cada relacionamento "pai - filho"
- Todos os elementos são **alcançáveis** a partir da raiz
- Árvore é um tipo especial de grafo

- Não há regras definindo a ligação entre dois elementos

Motivação para o uso de Grafos

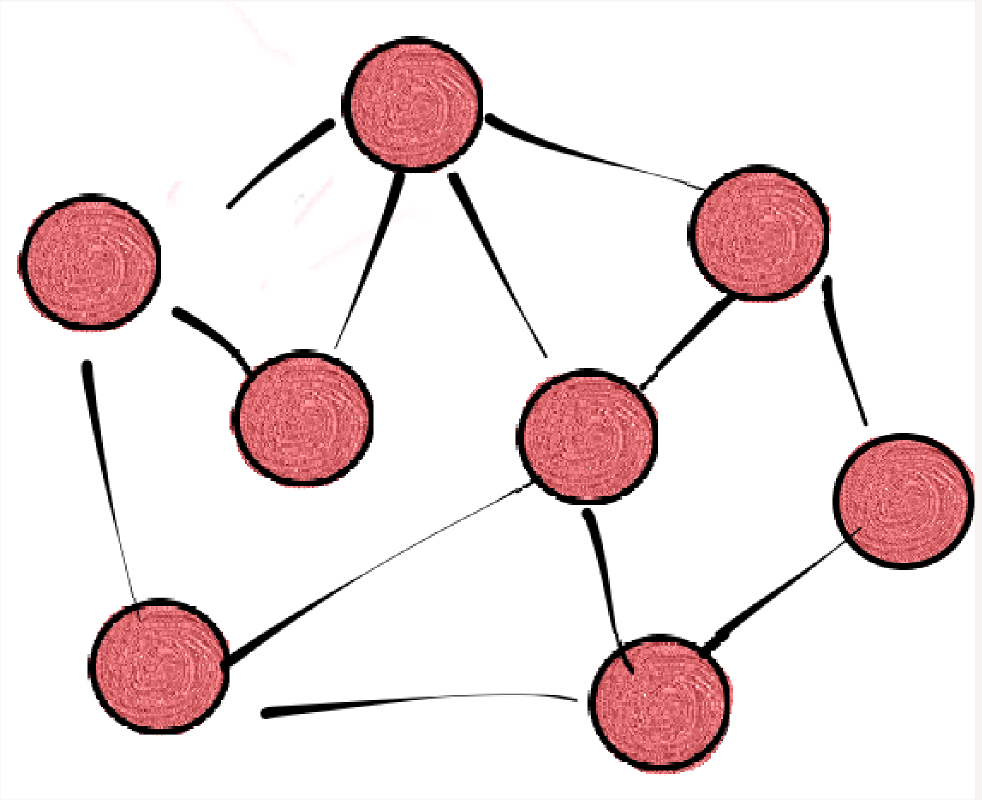
Muitas aplicações em computação necessitam considerar conjuntos de ligações entre pares de elementos para analisar situações como:

1. Existência de um **caminho** para ir de um elemento até outro, seguindo as ligações.
2. Escolha do melhor (ou menor) caminho entre dois elementos.
3. Definição da **abrangência** de um elemento, ou seja, os lugares que se pode chegar a partir dele.

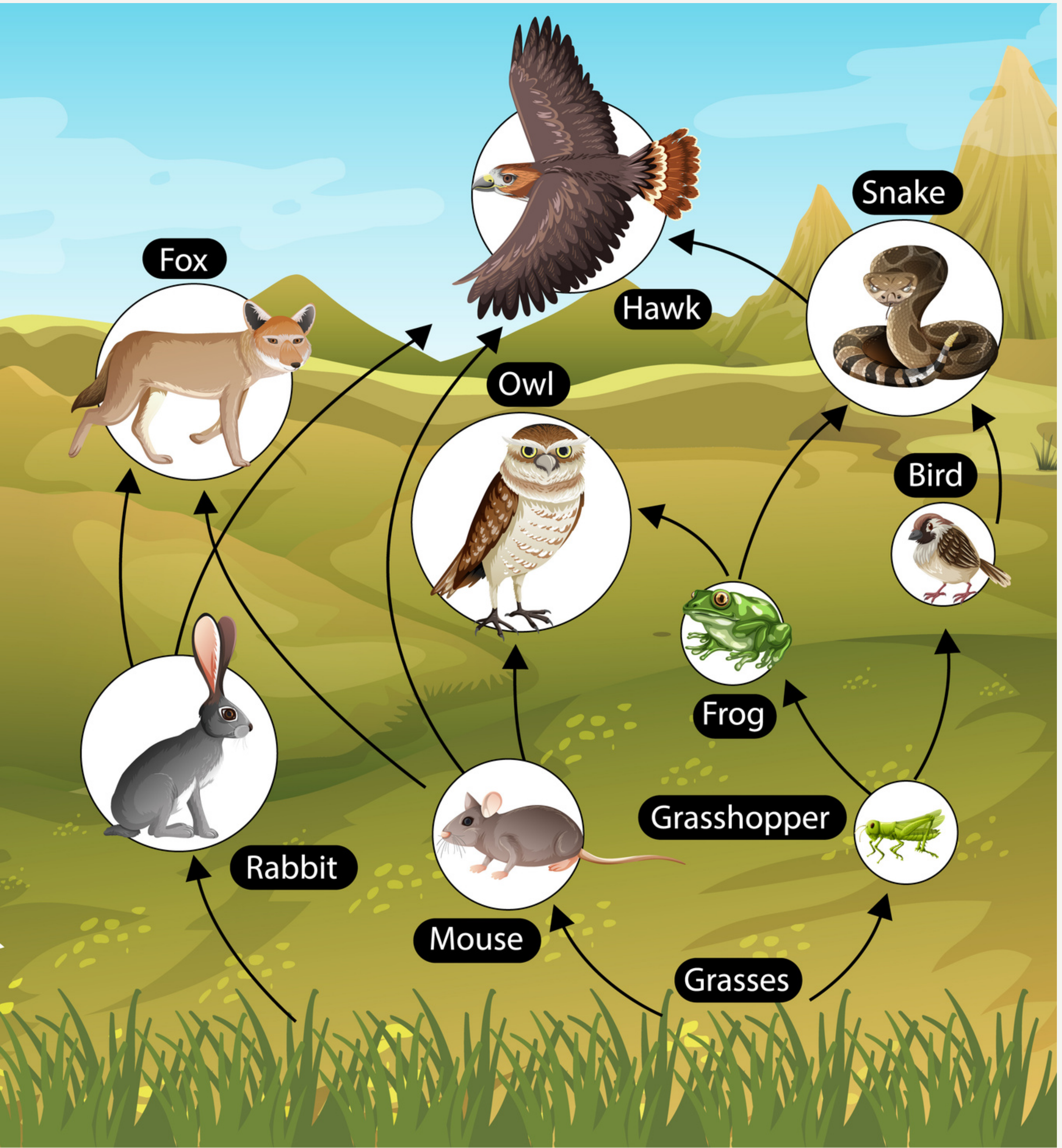


Aplicações em Grafos

- Interdependência de espécies em um ecossistema
- Influência entre pessoas de uma organização
- Combinação entre oferta e procura (vagas de trabalho e pessoas interessadas)
- Resultados de torneios esportivos
- Colaboração entre pesquisadores
- Mapas geográficos / rodoviários / aéreos
- Ligações (links) entre websites



Aplicações



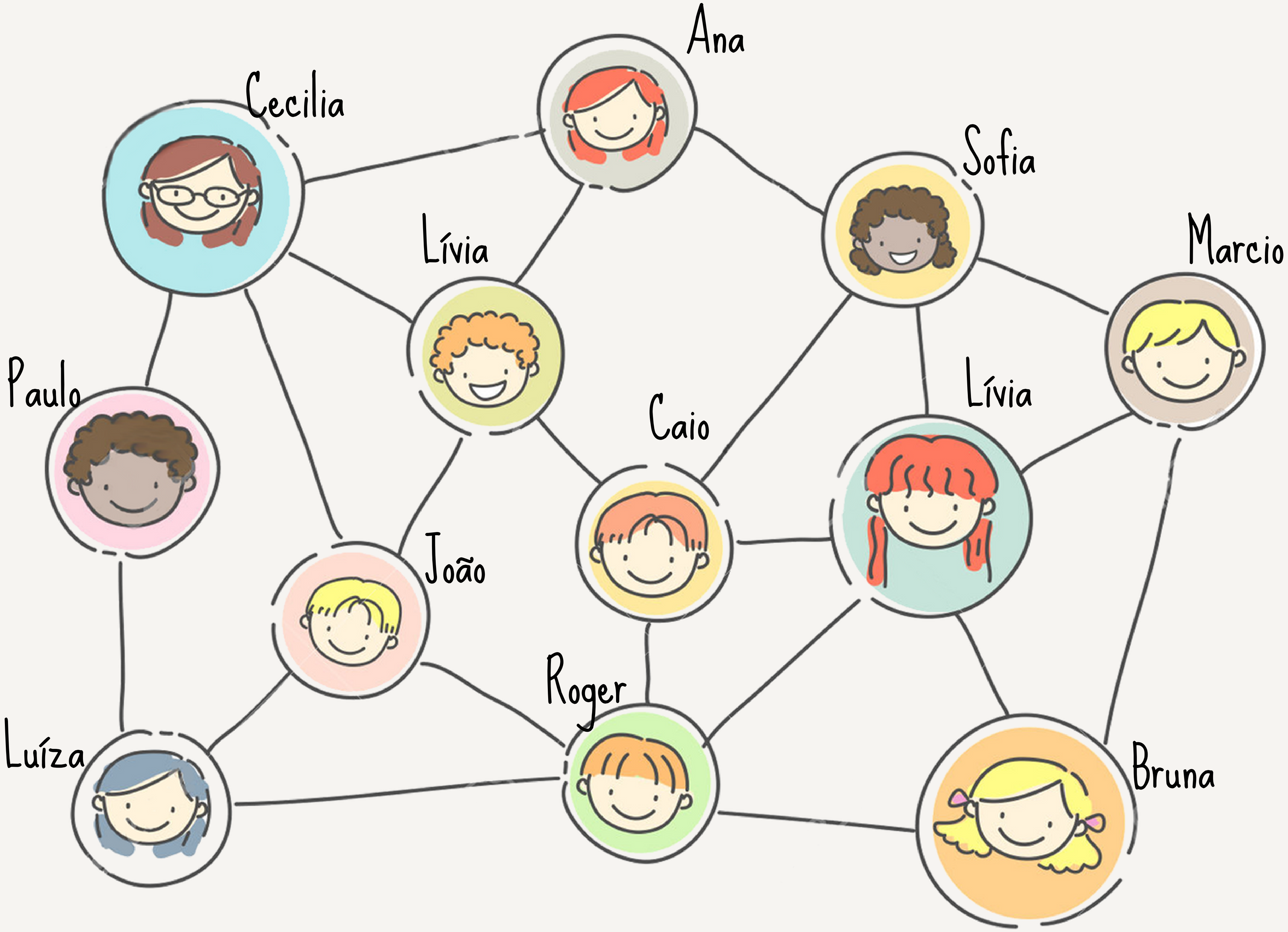
Ecossistema na Natureza

Grafo de representação da interdependência entre algumas espécies: cadeia alimentar

Em Ecologia, as cadeias/redes alimentares quantificam a interação entre espécies diversas. Os vértices simbolizam as espécies e a relação predador-presa é definida por arestas.

Vecteezy.com

Aplicações

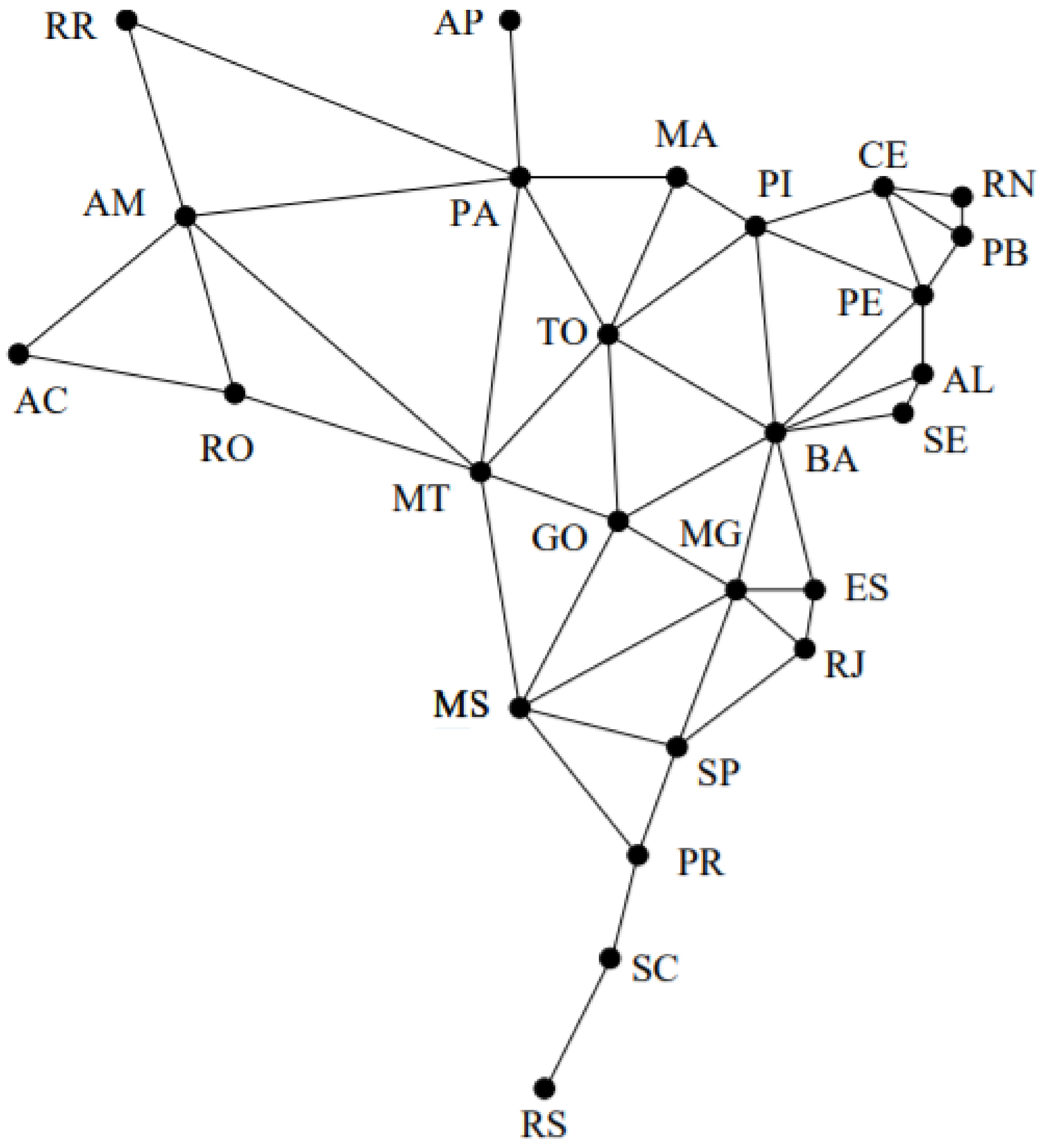


Rede Social (Facebook)

Que novos amigos podem ser indicados para Bruna?

pikpng.c

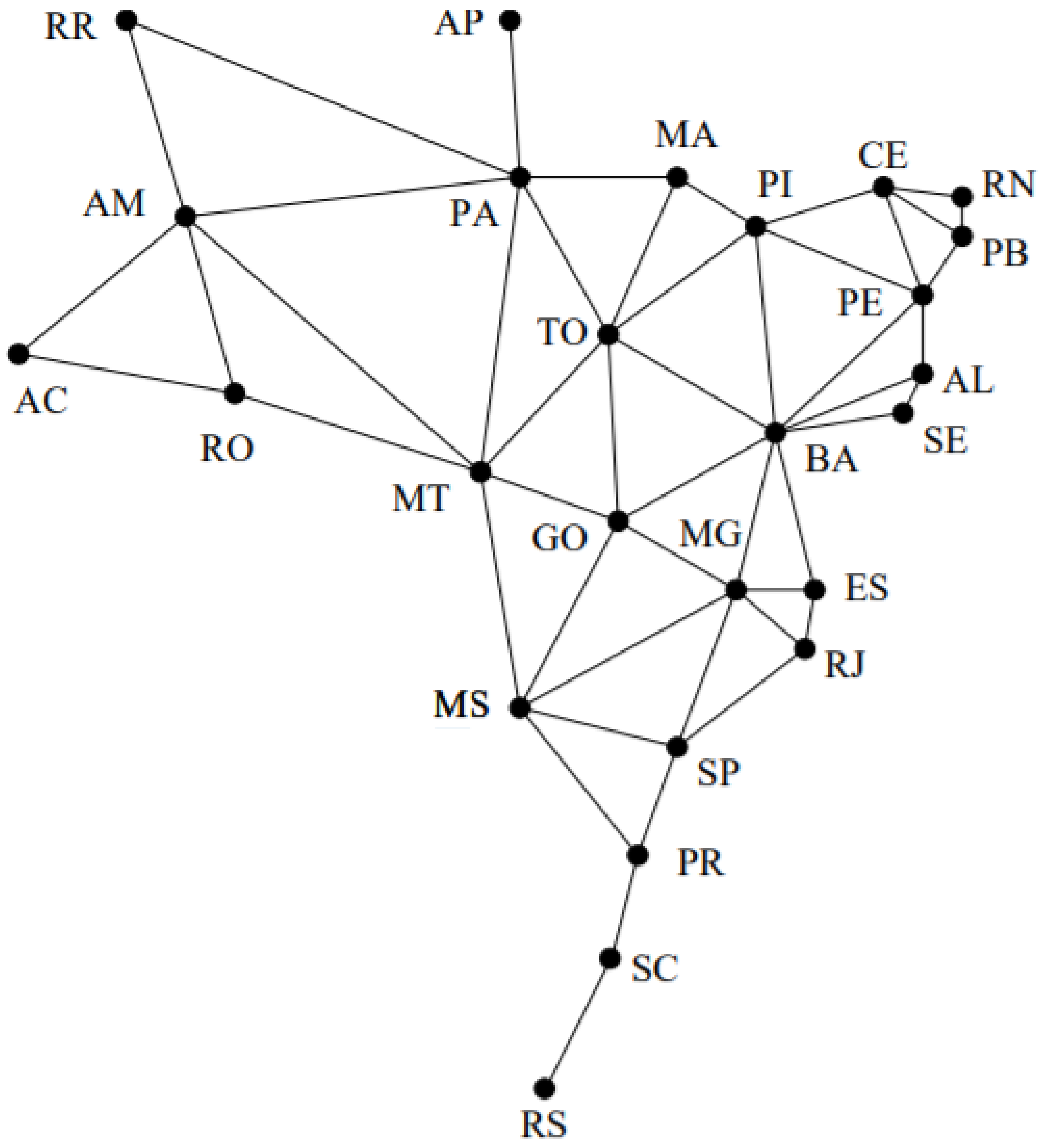
Adjacência entre estados do Brasil



Grafo dos **estados do Brasil**: cada vértice é um dos estados da República Federativa do Brasil e dois estados são adjacentes se têm uma fronteira comum.

1. Quantos elementos (estados) tem o grafo?
2. Quantas ligações?

Adjacência entre estados do Brasil

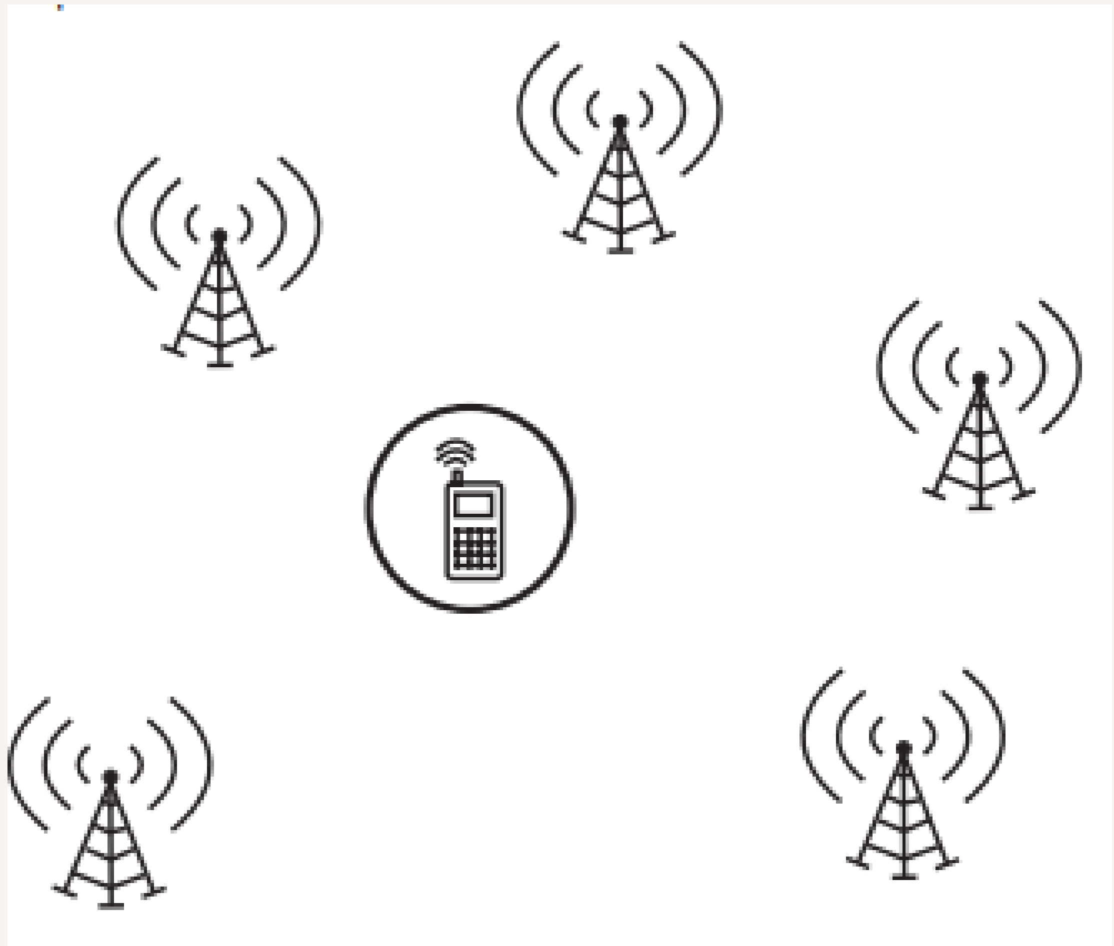


Se desejarmos criar um **mapa colorido** com a seguinte restrição: regiões vizinhas devem ter cores diferentes. Neste caso, dois problemas podem ser estudados:

1. Como colorir o mapa de forma a atender à restrição?
2. Qual o menor número de cores necessárias?

Alocação de Frequências

Seja uma **rede de telefonia celular** com estações base (torres) e a seguinte restrição: células vizinhas não podem usar mesma frequência (por problemas de interferência).



Problema 1: Como alocar frequências às células?
Problema 2: Qual o menor número de frequências necessárias?

Conceitos Básicos

Grafo: um par contendo um conjunto de vértices e um conjunto de arestas.

Vértice: objeto simples que pode ter nome e outros atributos

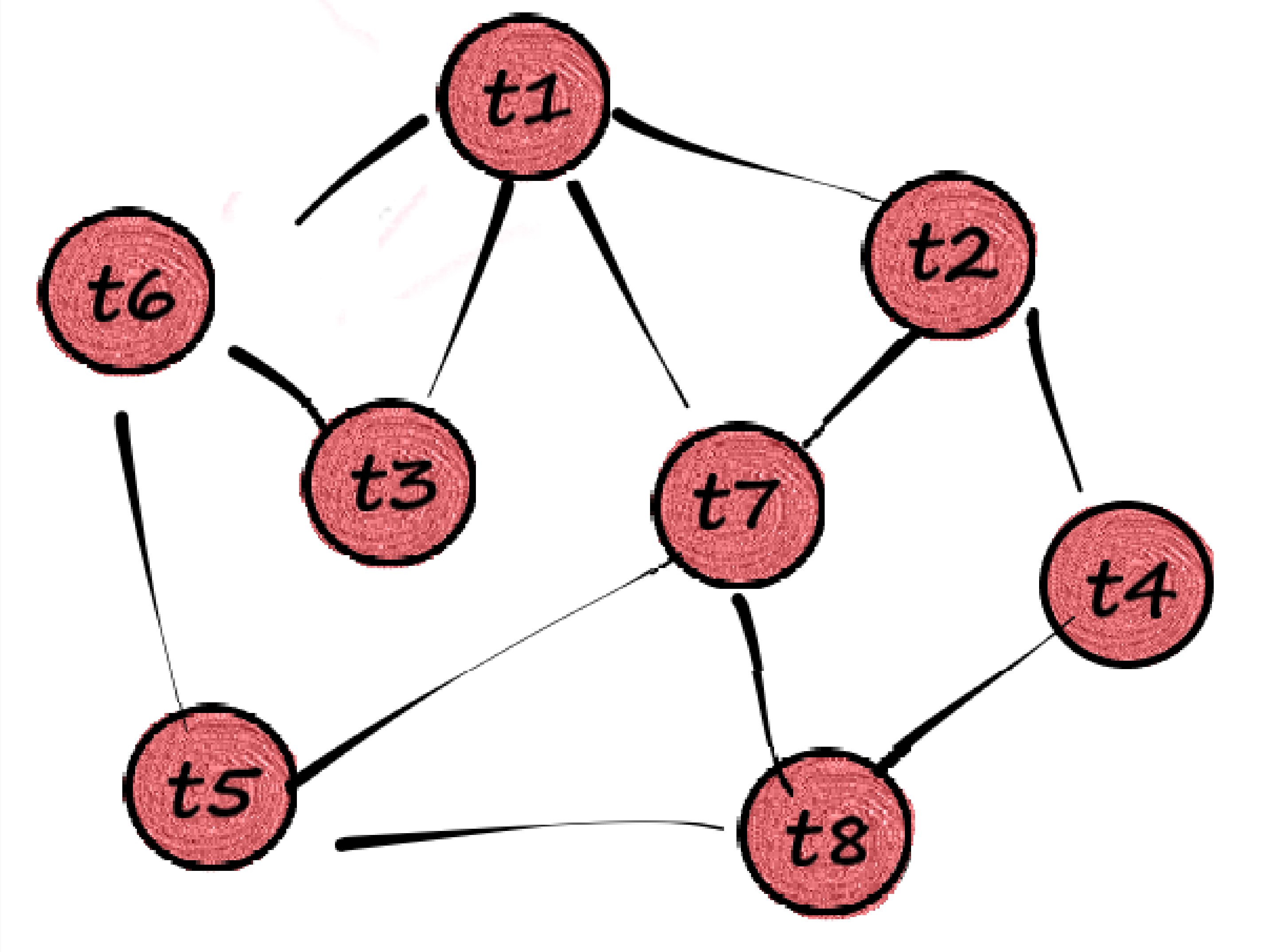
Aresta: conexão entre dois vértices

Notação: $G = (V, A)$

- G: grafo
- V: conjunto de vértices
- A: conjunto de arestas

Conceitos Básicos

Exemplo de um grafo ***G***:



$G = (V, A)$

$V = \{t1, t2, t3, t4, t5, t6, t7, t8\}$

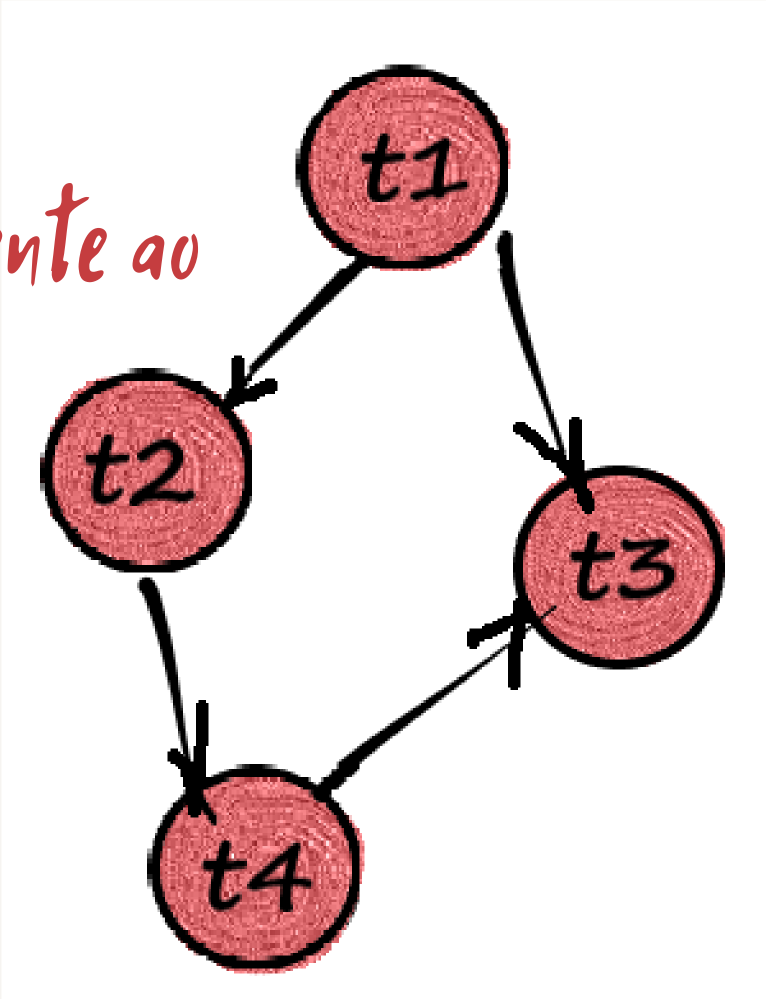
$A = \{\{t1,t6\}, \{t1,t3\}, \{t1,t7\}, \{t1,t2\}, \{t2,t7\}, \{t2,t4\}, \{t3,t6\}, \{t4,t8\}, \{t5,t6\}, \{t5,t7\}, \{t5,t8\}, \{t7,t8\}\}$

Cada elemento em *A* é um par não ordenado.

Conceitos Básicos

Um **grafo direcionado** G = (V, A) consiste de V, um conjunto não vazio de vértices (ou nós) e de A, um conjunto de arestas em que cada uma é definida por um ***par ordenado***.

O vértice t1 é adjacente ao vértice t2

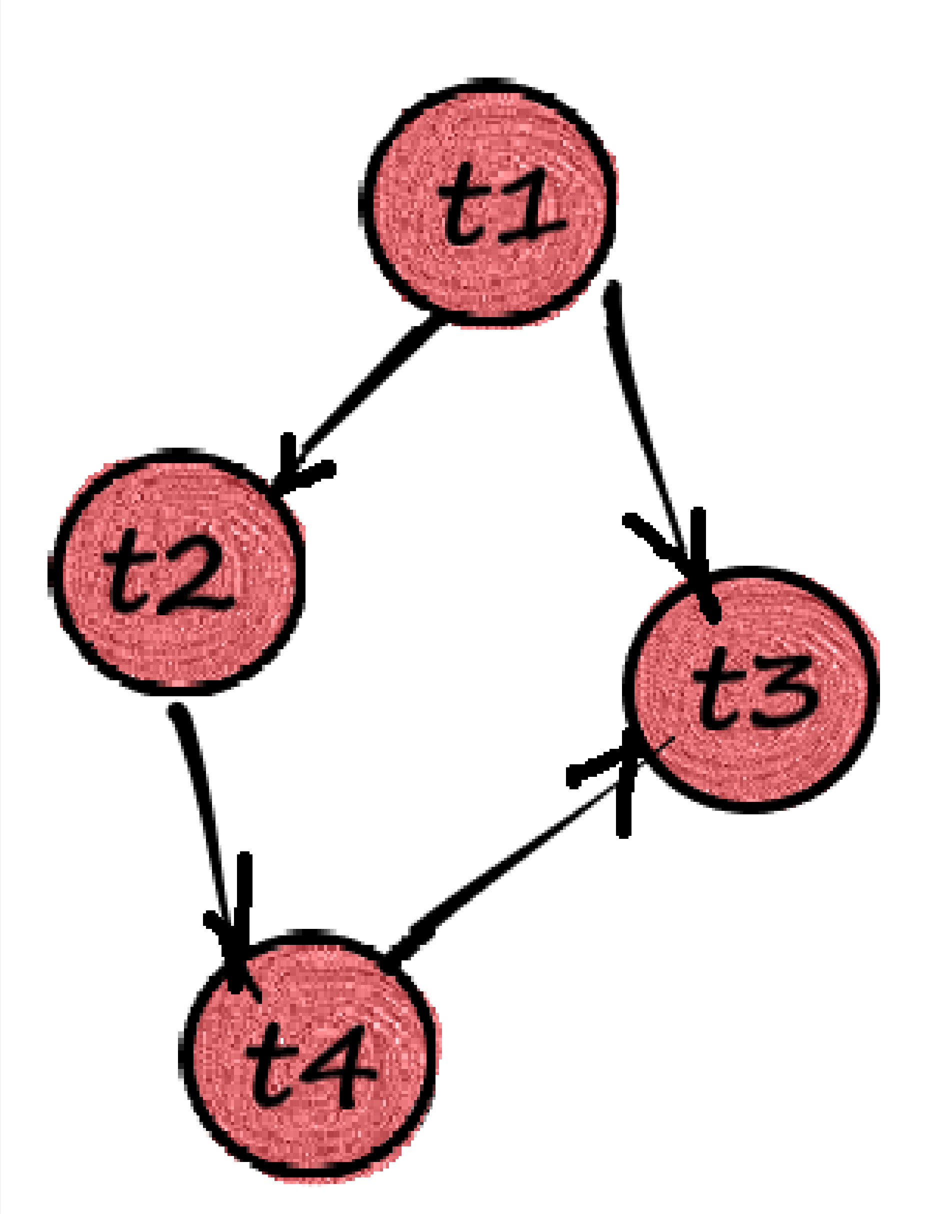


G = (V, A)

V = {t1, t2, t3, t4}

A = {(t1,t2), (t1,t3), (t2,t4), (t4,t3)}

Conceitos Básicos



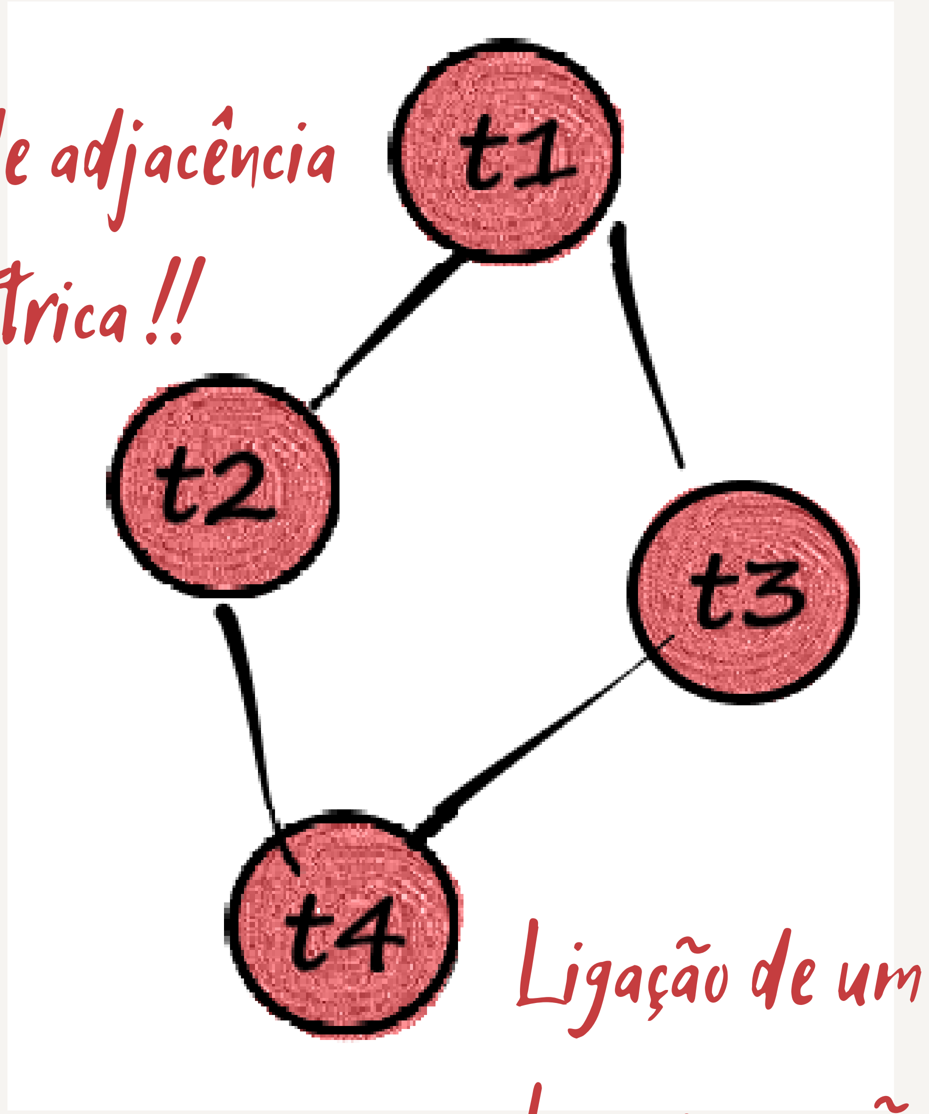
Em um **grafo direcionado**, se existe uma aresta (*t1*,*t2*) então:

- o vértice *t1* é **adjacente** ao vértice *t2*
- a aresta sai do vértice *t1* (origem) a aresta chega no vértice *t2* (destino)
- a existência de (*t1*,*t2*) não implica na existência de (*t2*,*t1*), ou seja o vértice *t2* não é adjacente ao vértice *t1*
- os vértices *t1* e *t2* são **vizinhos**

Conceitos Básicos

Um **grafo não direcionado** G = (V, A) consiste de V, um conjunto não vazio de vértices (ou nós) e de A, um conjunto de arestas em que cada uma é definida por um ***par não ordenado***.

A relação de adjacência é simétrica !!



Ligação de um vértice para ele mesmo: não permitida !!

G = (V, A)
V = {t1, t2, t3, t4}
A = {{t1,t2}, {t1,t3}, {t2,t4}, {t3,t4}}

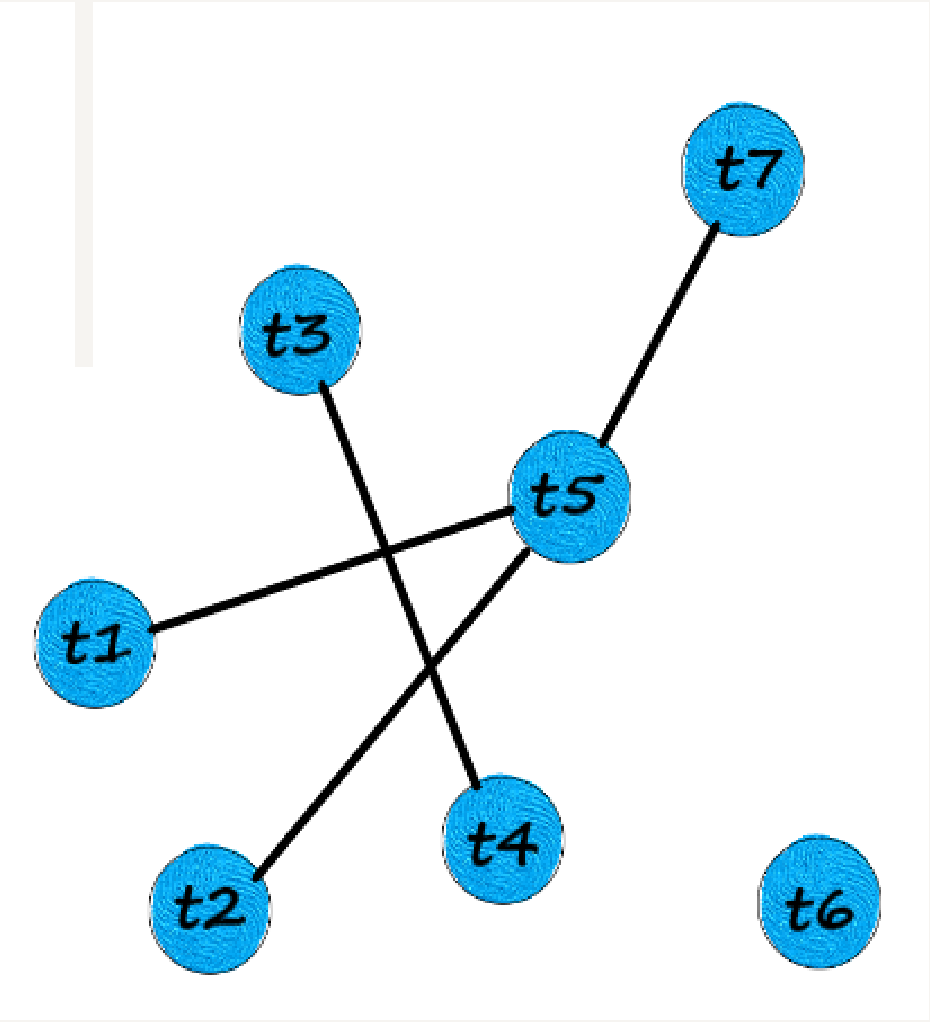
Em *A*, cada par indica a ligação nas duas direções, assim:

{t1,t2} indica que t1 -> t2 e t2 -> t1

Conceitos Básicos

Dois vértices *u* e *v* de um grafo não direcionado são **adjacentes** (ou vizinhos) quando eles forem os extremos de uma mesma aresta *(u,v)*.

t3 é adjacente a *t4* ?
t5 é adjacente a *t4* ?
t4 é adjacente a *t3* ?



A aresta {*t3*,*t4*} é dita **incidente** a *t3* e a *t4*

Grau de um vértice

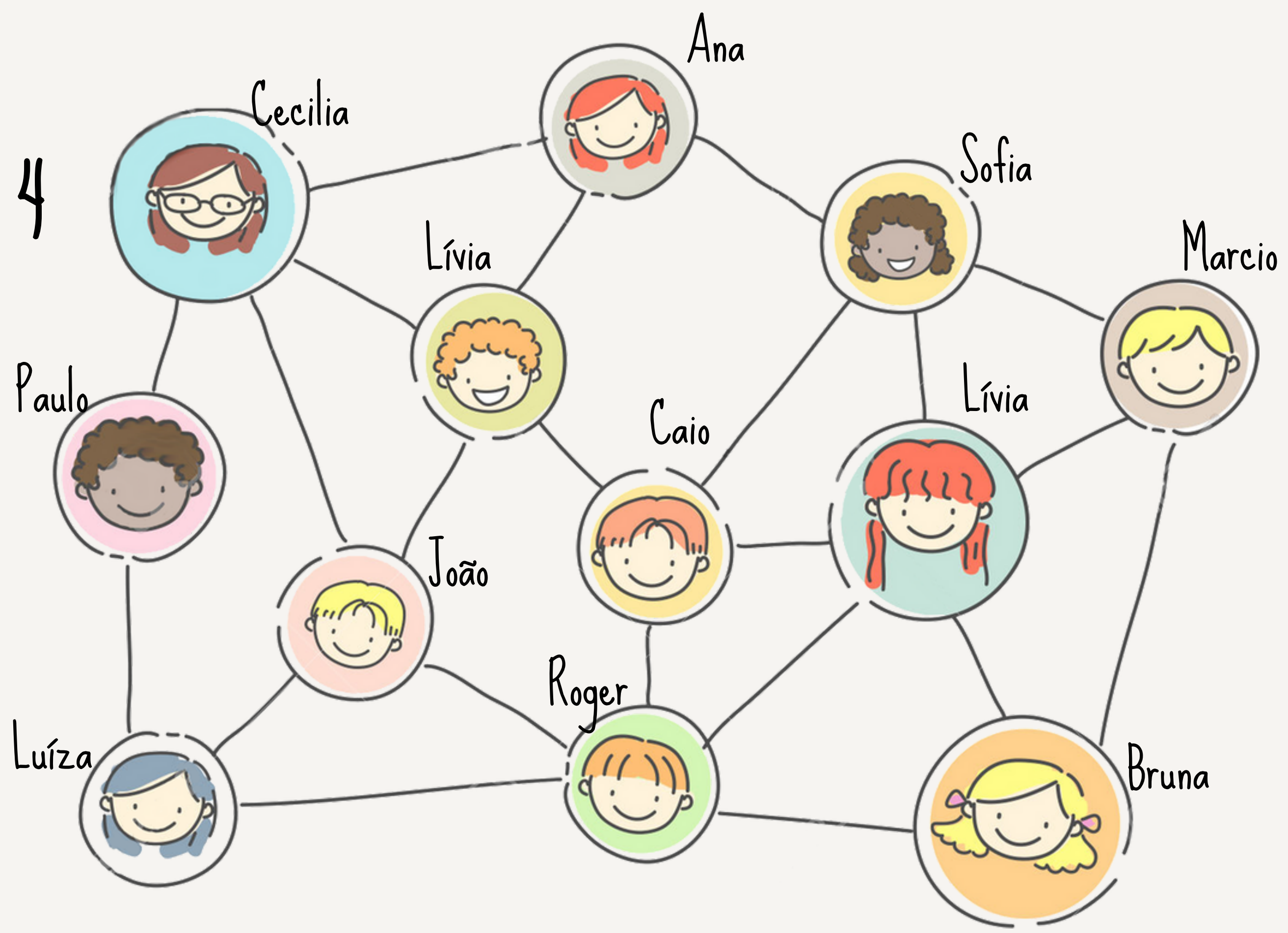
Em grafos não direcionados:

- O grau de um vértice é o número de arestas que incidem nele.
- Um vérice de grau zero é dito **isolado** ou **não conectado**.

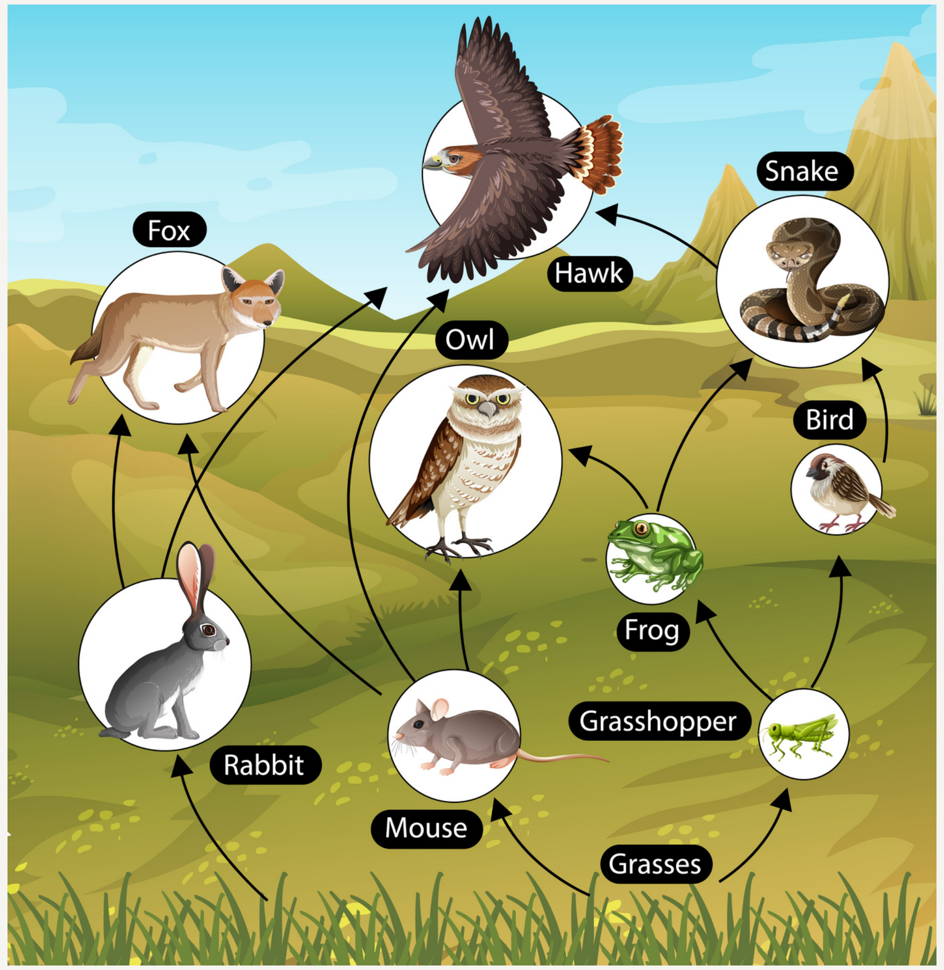
Em grafos direcionados

- O grau de um vértice é o número de arestas que saem dele (*out-degree*) mais o número de arestas que chegam nele (*in-degree*).

Cecília: grau 4



Coelho: grau 3



Caminho em um grafo

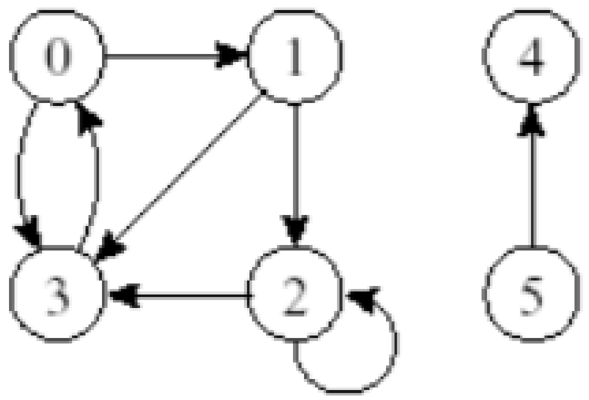
Um caminho de **comprimento** k de um vértice x a um vértice y em um grafo $G = (V, A)$ é uma seqüência de vértices $(v_0, v_1, v_2, \ldots, v_k)$ tal que $x = v_0$ e $y = v_k$, e $(v_{i-1}, v_i) \in A$ para $i = 1, 2, \ldots, k$.

O comprimento de um caminho é o número de arestas nele, isto é, o caminho contém os vértices $v_0, v_1, v_2, \ldots, v_k$ e as arestas $(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$.

Se existir um caminho c de x a y então y é **alcançável** a partir de x via c.

Um caminho é **simples** se todos os vértices do caminho são distintos.

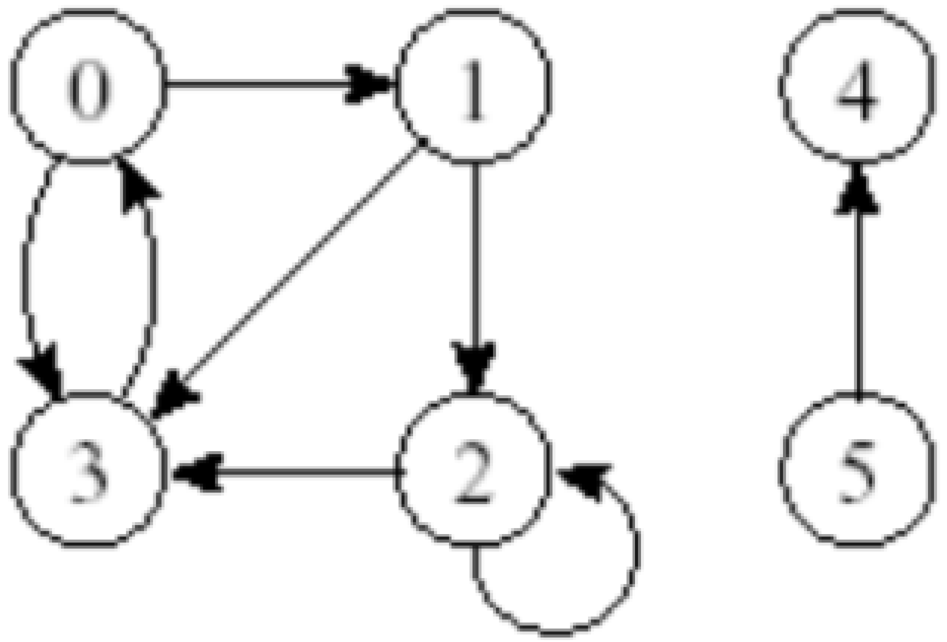
Ex.: O caminho $(0, 1, 2, 3)$ é simples e tem comprimento 3. O caminho $(1, 3, 0, 3)$ não é simples.



Caminho em um grafo: ciclo

Em um grafo direcionado:

- Um caminho $(v_0, v_1, \ldots, v_k)$ forma um ciclo se $v_0 = v_k$ e o caminho contém pelo menos uma aresta.
- O ciclo é simples se os vértices $v_1, v_2, \ldots, v_k$ são distintos.
- O *self-loop* é um ciclo de tamanho 1.

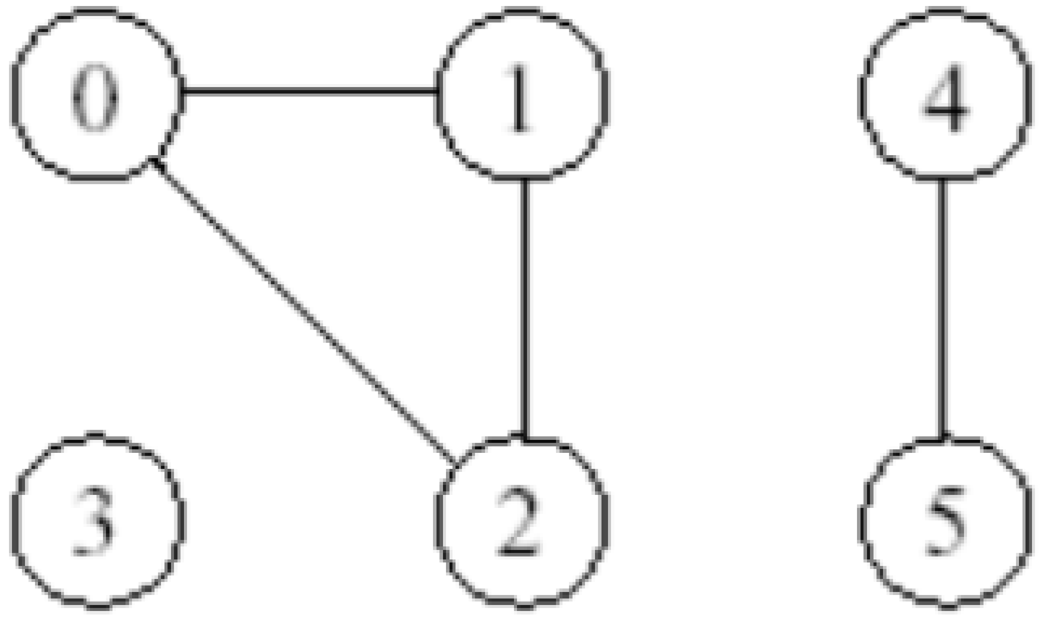


O caminho (0,1,2,3,0) forma um ciclo simples.

Caminho em um grafo: ciclo

Em um grafo não direcionado:

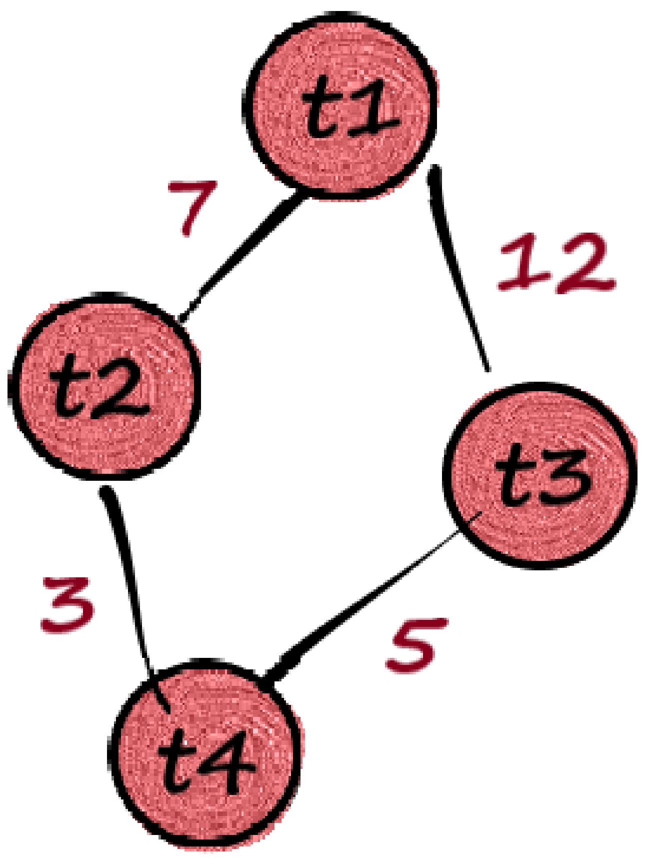
- Um caminho $(v_0, v_1, \ldots, v_k)$ forma um ciclo se $v_0 = v_k$ e o caminho contém pelo menos três arestas.
- O ciclo é simples se os vértices $v_1, v_2, \ldots, v_k$ são distintos.



O caminho (0,1,2,0) forma um ciclo simples.

Grafo Ponderado

Em muitas aplicações de grafos existem dados quantitativos associados a vértices ou ligações (arestas) relacionados ao problema. Estes dados quantitativos são chamados de **pesos** e o grafo é denominado **ponderado**.

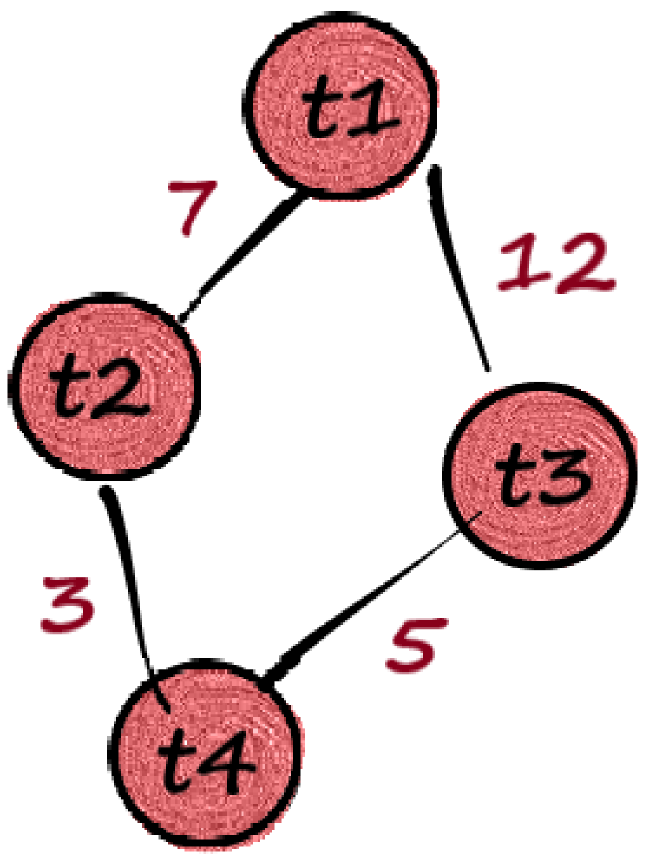


Arestas são triplas: *(u, v, valor)*

A = {{t1,t2,7}, {t1,t3,12}, {t2,t4,3}, {t3,t4,5}}

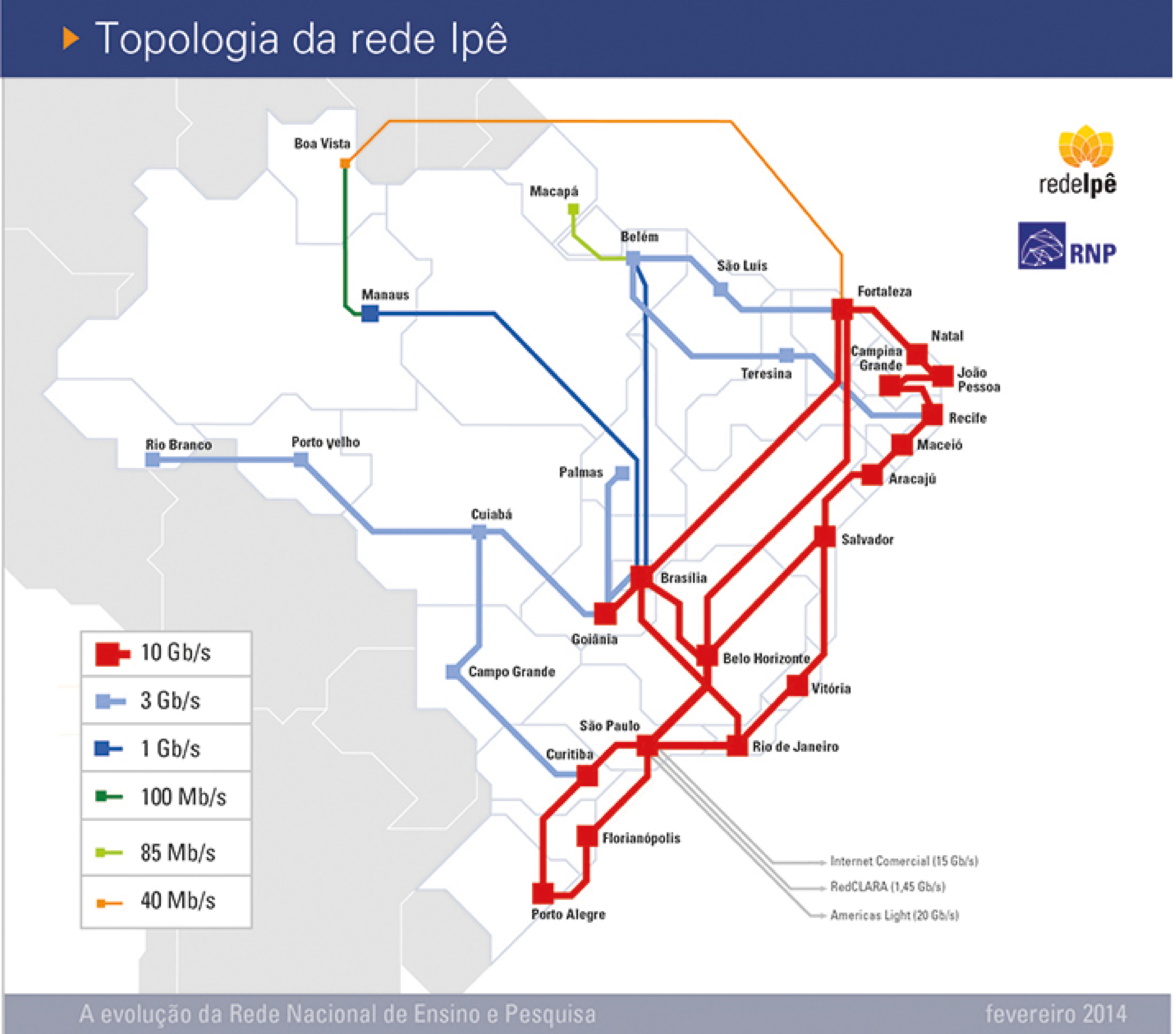
Grafo Ponderado

Em muitas aplicações de grafos existem dados quantitativos associados a vértices ou ligações (arestas) relacionados ao problema. Estes dados quantitativos são chamados de **pesos** e o grafo é denominado **ponderado**.



Arestas são triplas: *(u, v, valor)*

A = {{t1,t2,7}, {t1,t3,12}, {t2,t4,3}, {t3,t4,5}}



Exemplo de Grafo Ponderado

Rede Ipê

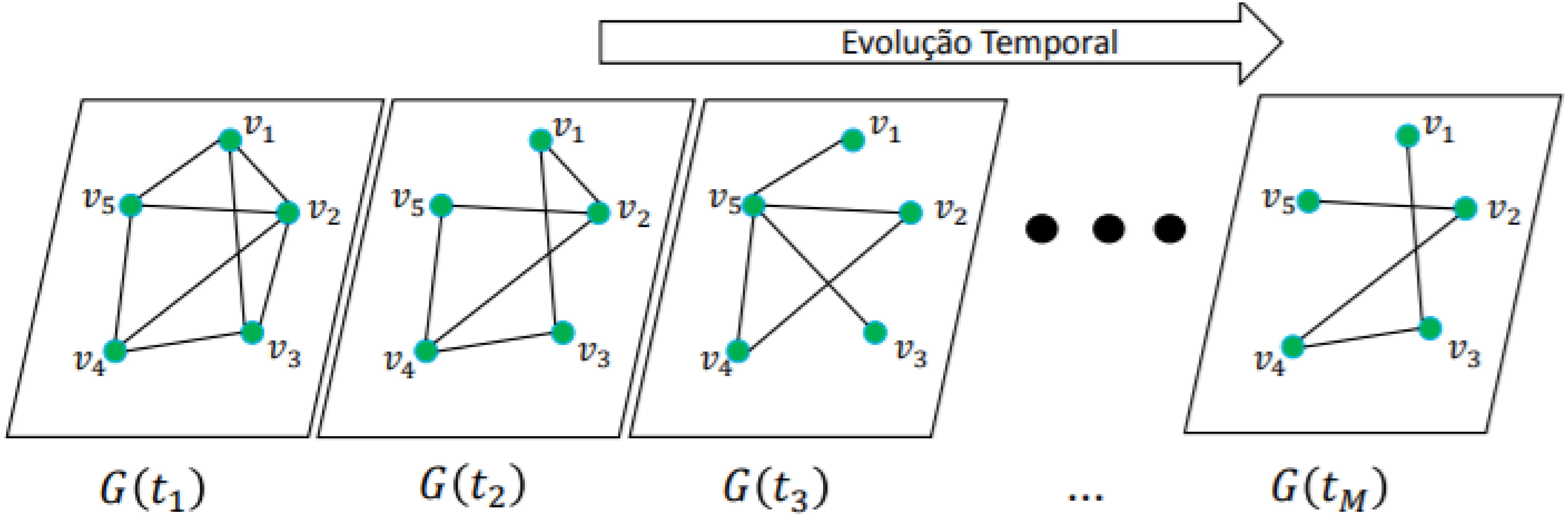
Transmissão de dados para projetos científicos e desenvolvimento de novas tecnologias.

As conexões estão associadas à diferentes faixas de velocidade de tráfego, de acordo com a cor/legenda.

Grafos Variantes no Tempo

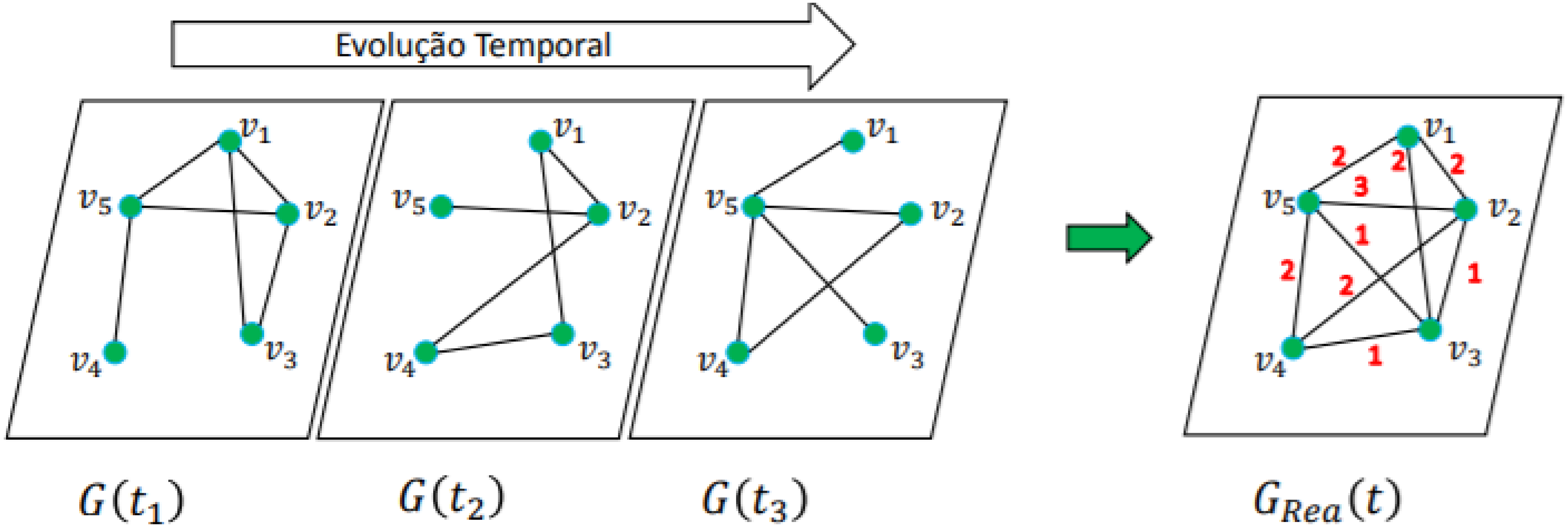
Devido às características dinâmicas presentes em muitas redes, surgiu o conceito de Grafos Variantes no Tempo.

São representados como uma sequência de grafos com número fixo de vértices e ordenados no tempo:



Grafos Variantes no Tempo

Um grafo ponderado pode ser construído a partir da superposição das conexões de todos os grafos. Para este fim, consideram-se todas as conexões existentes em todos os grafos de um intervalo de tempo:



Rea = Rede Estática Agregada

Grafos Variantes no Tempo: Aplicações

O formalismo dos GVTs é utilizado em análises de redes de diversas áreas da pesquisa científica, como:

- análise de redes sociais
- análise da conectividade funcional do cérebro humano
- dinâmica de epidemias
- mobilidade urbana
- encaminhamento de mensagens (IoT) em redes tolerantes à atrasos e desconexões

Problema: Alto consumo de espaço e de tempo para consultas e processamento

Bibliografia

N. Ziviani. **Projeto de Algoritmos**, Thomson, 2a. Edição, 2004.

Y. Langsam, M. J. Augenstein and A. M. Tenenbaum. **Data Structures Using C and C++**, Prentice Hall, 2a Edição, 1996.

R. A. Souza. ***Redes de Interação Preferencial: um modelo de redes complexas com dinâmica de arestas ponderadas***. Dissertação de Mestrado. UFBA, 2016.