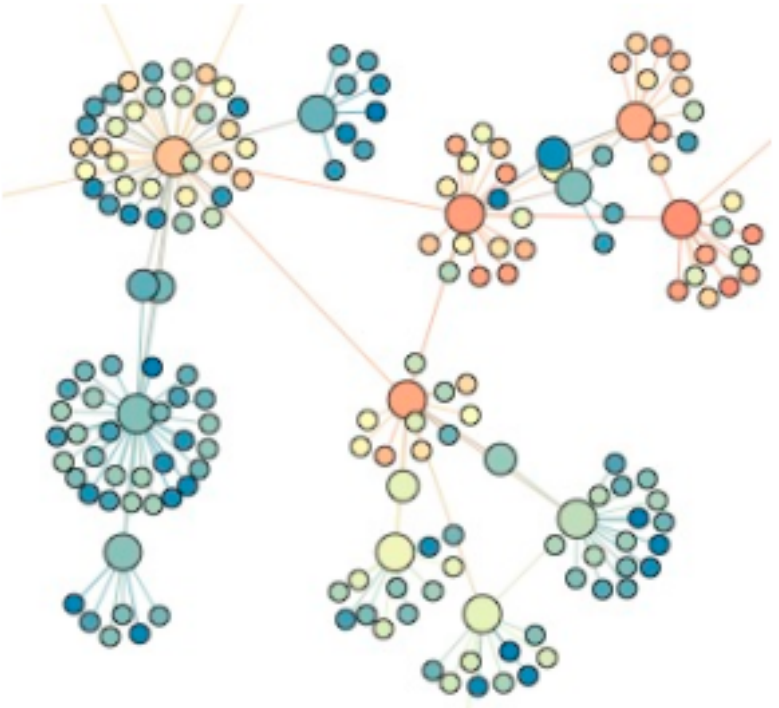


Teoria dos Grafos em Java

Thiago B. Procaci

Apresentação

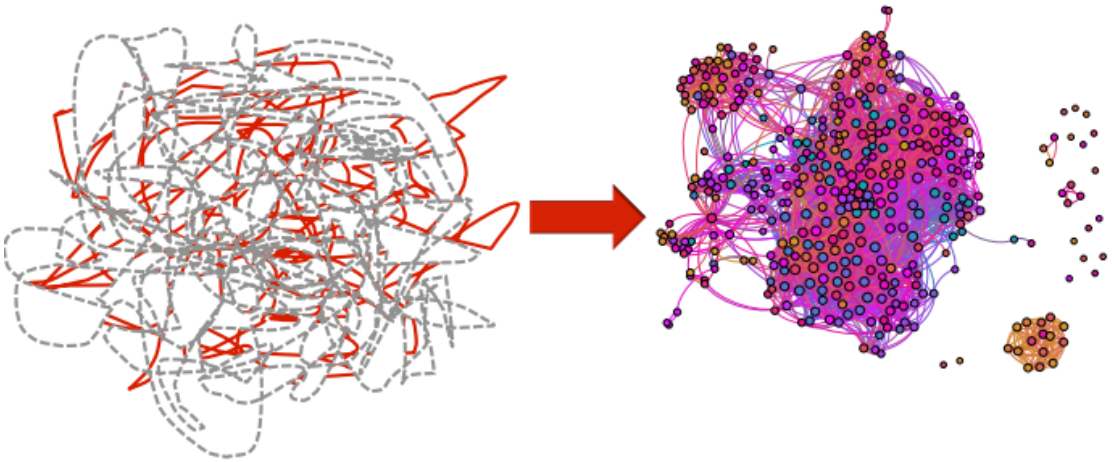
- Formação
  - UFJF → Graduação em Ciência da Computação
  - UNIRIO → Mestrado e Doutorado (em andamento)
- Semantics and Learning → Grupo de Pesquisa
  - Dados Não Estruturados → Extração de Conhecimento
  - Análise de Redes Sociais
  - Machine Learning
- Mercado
  - Desenvolvimento Java
  - Arquitetura de Sistemas
  - Cientista de Dados (Python, R)
- Docência
  - Palestras e Seminários
  - Estágio

Objetivos da Aula

- Compreender o conceito de grafos
- Para que servem grafos?
- Construir um grafo em Java
- Ser capaz de criar aplicação
  - Rede de amigos
  - Recomendação de amizades

Introdução

- O que podemos entender modelando o mundo como grafos?
- Entidades → Relações → Complexidade
  - Pessoas interagindo em uma rede social
  - Caminhos entre lugares (estradas entre cidades)
  - Redes de distribuição de energia
- Complexidade → Abstração → Grafos
- Grafos
  - Relações entre entidades
  - Abstração nem sempre trivial
  - Ver de outra perspectiva
  - Grafos = Redes = Network
  - Pontos e Linhas



Blogs de Política

Eleições EUA 2004

- Liberais → Ponto Azul
- Conservadores → Ponto Vermelho
- Conexões (quem fala com quem)
  - Azul para Azul (liberal para liberal)
  - Vermelho para Vermelho (conservador para conservador)
  - Cor intermediária (liberal para conservador ou vice-versa)
- Homofilia
- Grafo → outra perspectiva

Comments:

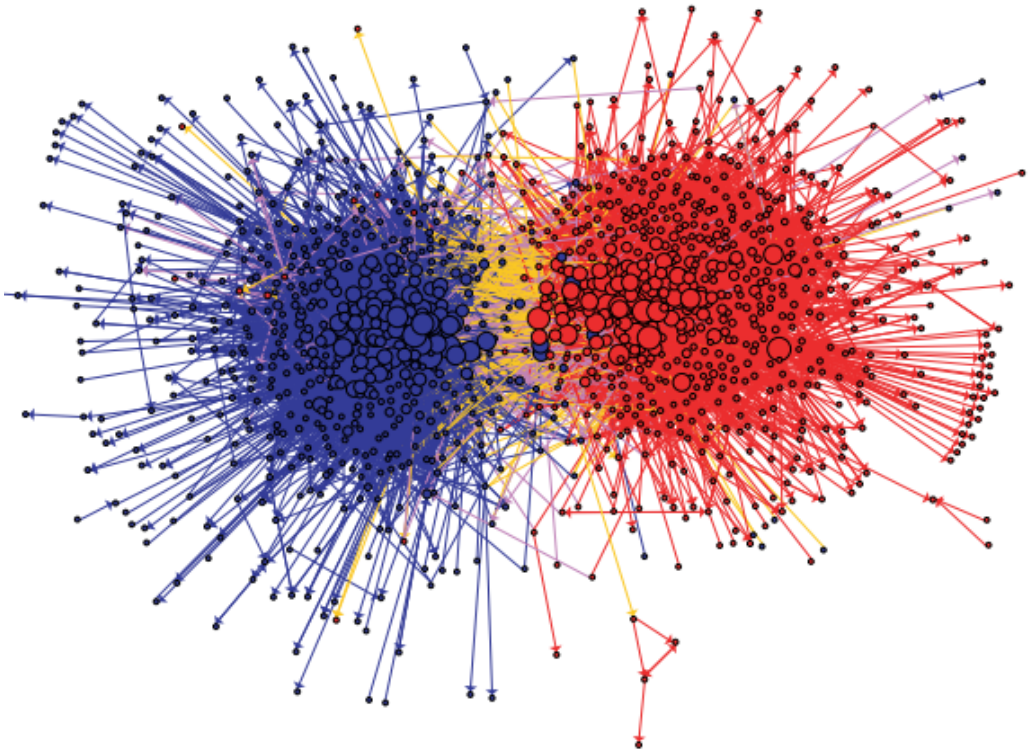
Can't the same be said of genocides? Who remembers the Armenians? Never again? But, it goes on and on. Samantha Power reminds us all the time, bless her, but then there are the tax cuts, social security, and the other important things in life that neocons are pushing. Who has time to be compassionate, other than our exalted compassionate conservative President, unless you're against him and are thus evil.

posted by Shag from Brookline : 4:41 PM

This issue was not really mentioned during the presidential campaign, suggesting again how easy the administration truly had it in certain ways: somehow, perhaps rightly, the competition did not think the public would truly want their leader to be seriously challenged, since maybe they too share his sentiments a bit too much.

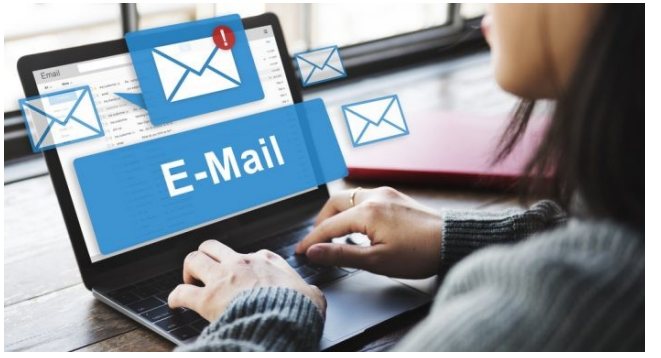
posted by Joe : 7:01 PM

As a conservative, I am anti-torture. I think that we, as Americans, hold ourselves to a higher standard than those in other countries, and we try to live up to principles of morality rather than expediency. While there are no doubt going to be situations like this in every war, they should be dealt with in a way to ensure that those responsible are punished, and that everyone else understands that this behavior is unacceptable.

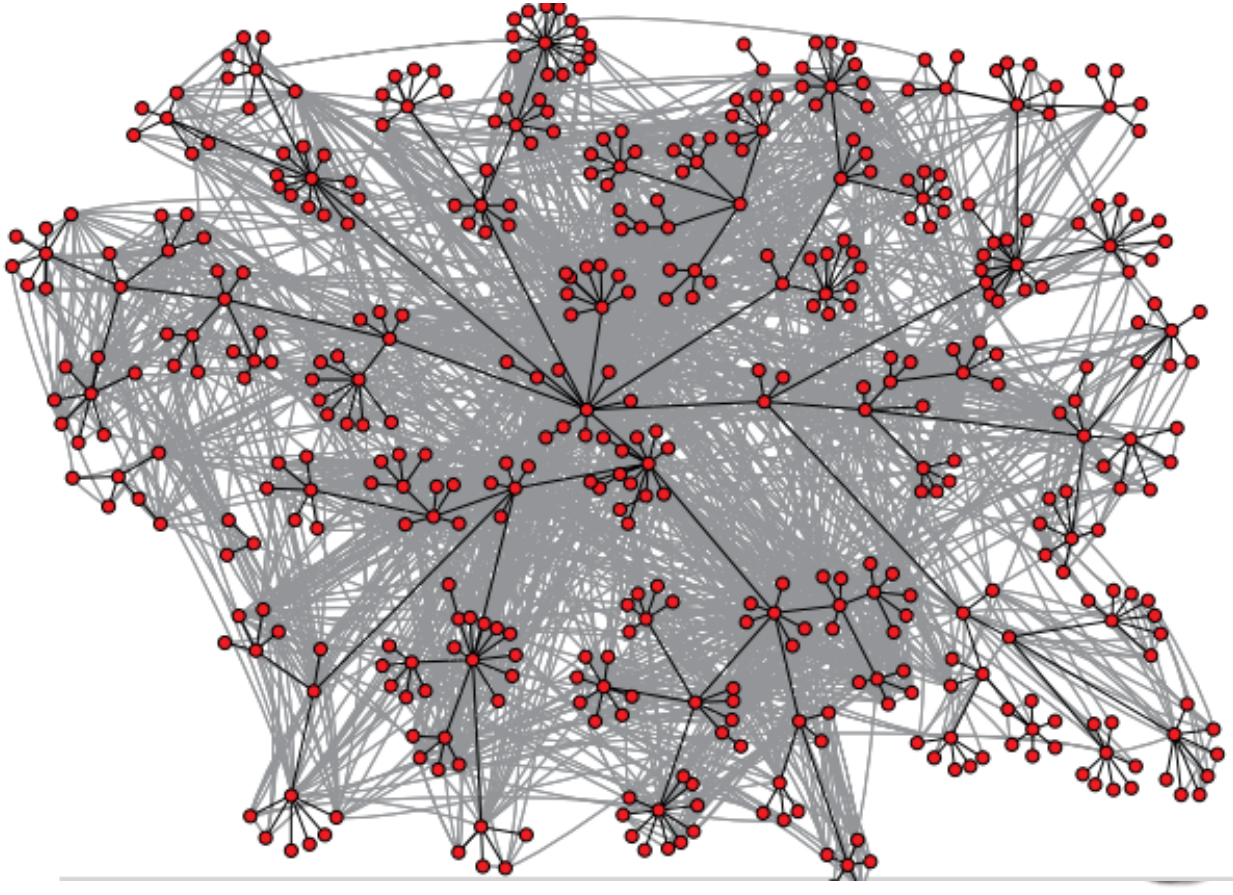


Fonte: The Political Blogosphere and the 2004 U.S. Election: Divided They Blog

Comunicação via e-mail

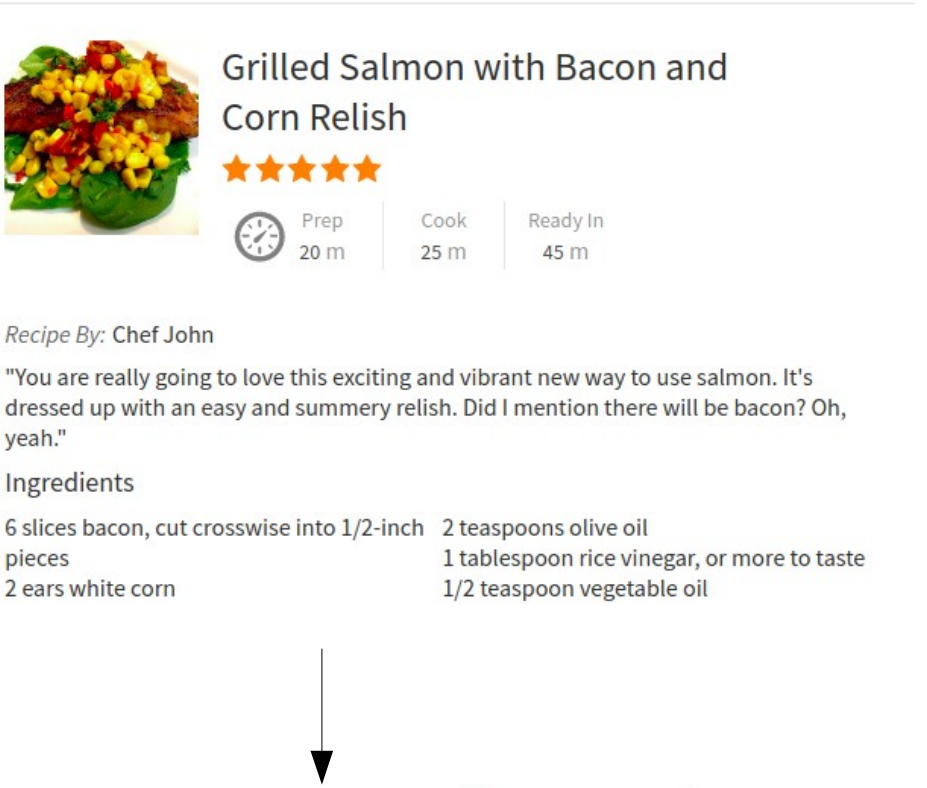


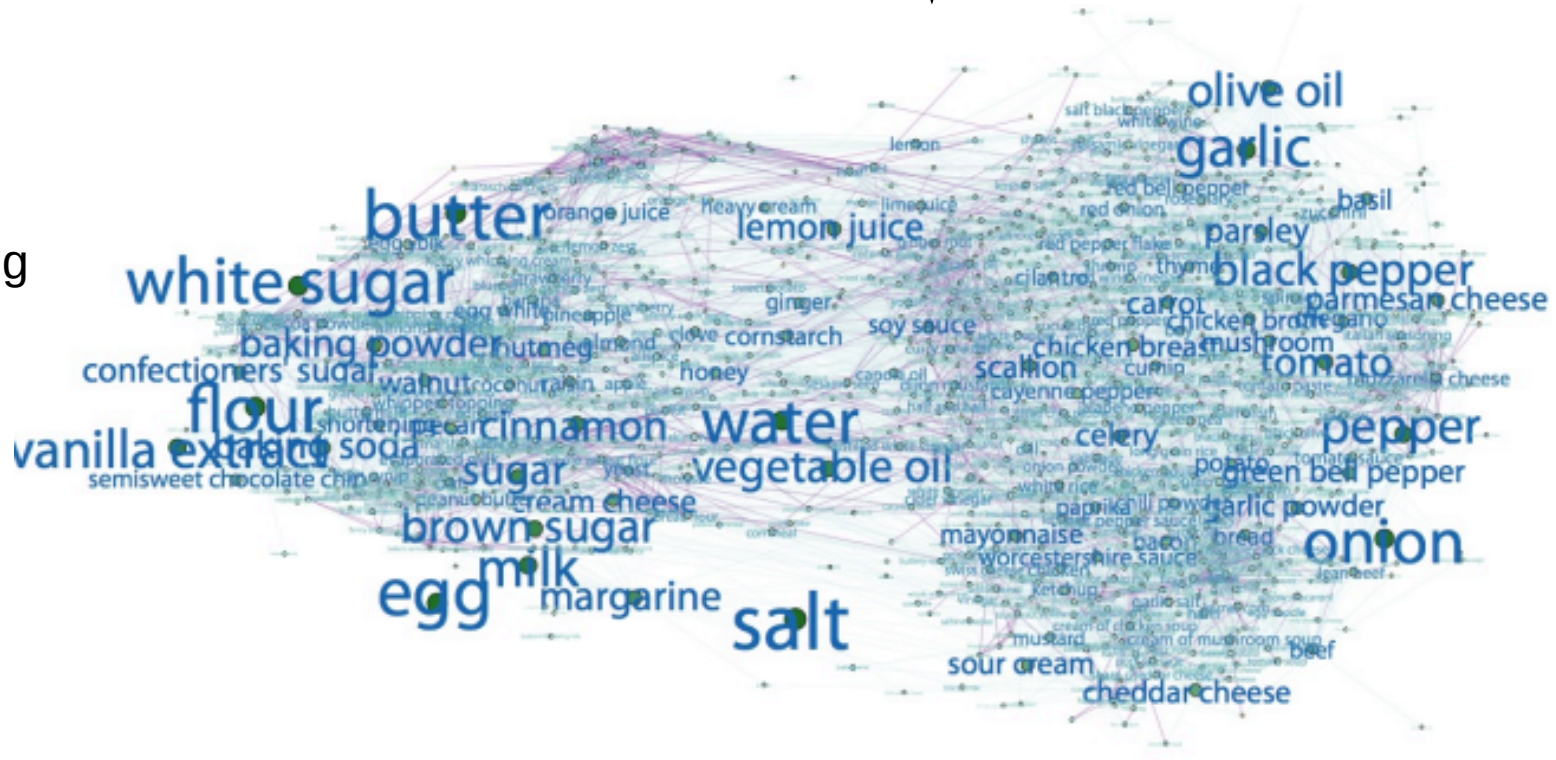
- Hewlett-Packard Labs
  - Pessoas → pontos vermelhos
  - Conexões em preto
    - Hierarquia
    - Canal formal
  - Conexões em cinza
    - Canais informais
    - Mais utilizados
    - Atalhos para resolver problemas



Rede de Ingredientes

- Como ingredientes estão conectados?
- Quais são os mais importantes?
- Novas receitas?



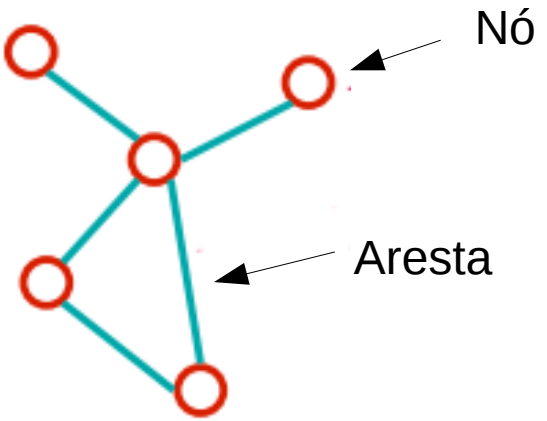


Fonte: Recipe recommendation using ingredient networks

Definição de Grafos

- Já sabemos o que podemos fazer com grafos
- O que são Grafos?
  - Conjunto de nós conectados por arestas

Pontos	Linhas	Campo/Ambiente
Vértices	Arcos	Matemática
Nós	Arestas	Computação
Atores	Relações	Sociologia
Usuários	Laços	Redes Sociais



Grafo em Java

- Definição: Conjunto de nós conectados por arestas
  - Um grafo tem várias arestas
  - Uma aresta conecta dois nós

Definindo o Nó

```
public class No {
    int id;
    String nome;
}
```

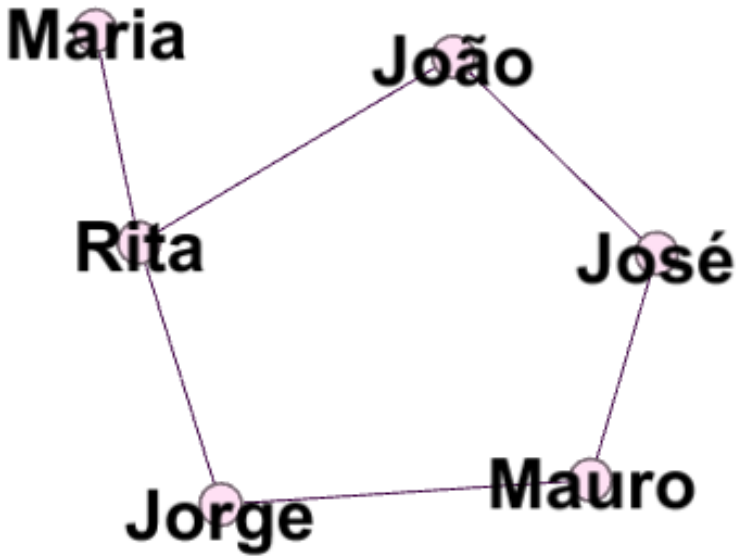
Definindo a Aresta

```
public class Aresta {
    No origem;
    No destino;
}
```

Definindo o Grafo

```
public class Grafo {
    List<Aresta> listaAresta;
}
```

Rede de Amigos



Definindo a classe No

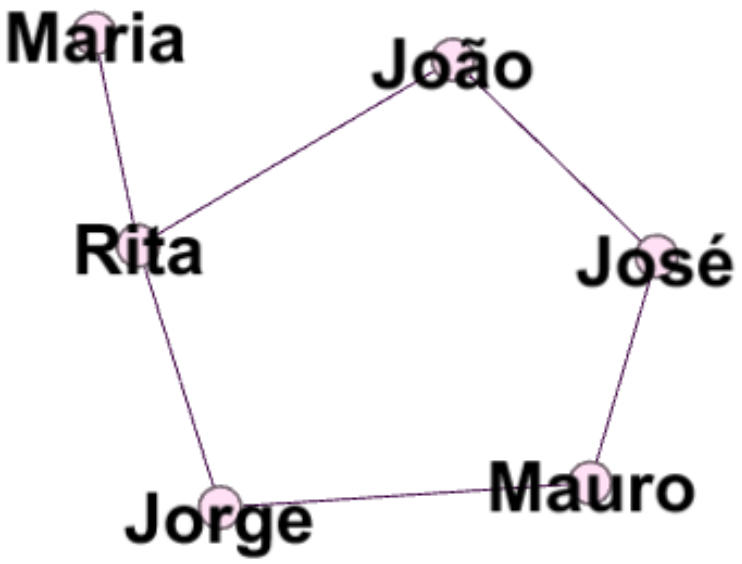
```
public class No {
    int id;
    String nome;

    public No(int id, String nome) {
        this.id = id;
        this.nome = nome;
    }
}
```

```
No maria = new No(1, "Maria");
No rita = new No(2, "Rita");
No joao = new No(3, "João");
No jose = new No(4, "José");
No mauro = new No(5, "Mauro");
No jorge = new No(6, "Jorge");
```

Criando os nós (instâncias)

Rede de Amigos




```java
public class Aresta {
    No origem;
    No destino;

    public Aresta(No origem, No destino) {
        this.origem = origem;
        this.destino = destino;
    }
}
```

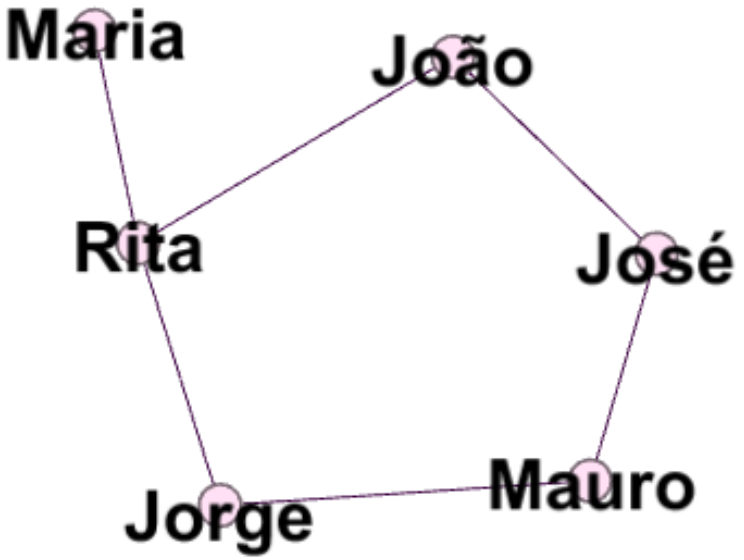
```java
No maria = new No(1, "Maria");
No rita = new No(2, "Rita");
No joao = new No(3, "João");
No jose = new No(4, "José");
No mauro = new No(5, "Mauro");
No jorge = new No(6, "Jorge");
```

Criando nós

```java
Aresta mariaRita = new Aresta(maria, rita);
Aresta ritaJoao = new Aresta(rita, joao);
Aresta ritaJorge = new Aresta(rita, jorge);
Aresta joaoJose = new Aresta(joao, jose);
Aresta joseMauro = new Aresta(jose, mauro);
Aresta jorgeMauro = new Aresta(jorge, mauro);
```

Criando arestas

Rede de Amigos




```java
public class Grafo {
    List<Aresta> listaAresta;

    public Grafo() {
        listaAresta = new ArrayList<Aresta>();
    }

    public void adicionarAresta(Aresta aresta) {
        listaAresta.add(aresta);
    }
}
```

```java
Aresta mariaRita = new Aresta(maria, rita);
Aresta ritaJoao = new Aresta(rita, joao);
Aresta ritaJorge = new Aresta(rita, jorge);
Aresta joaoJose = new Aresta(joao, jose);
Aresta joseMauro = new Aresta(jose, mauro);
Aresta jorgeMauro = new Aresta(jorge, mauro);
```

Arestas

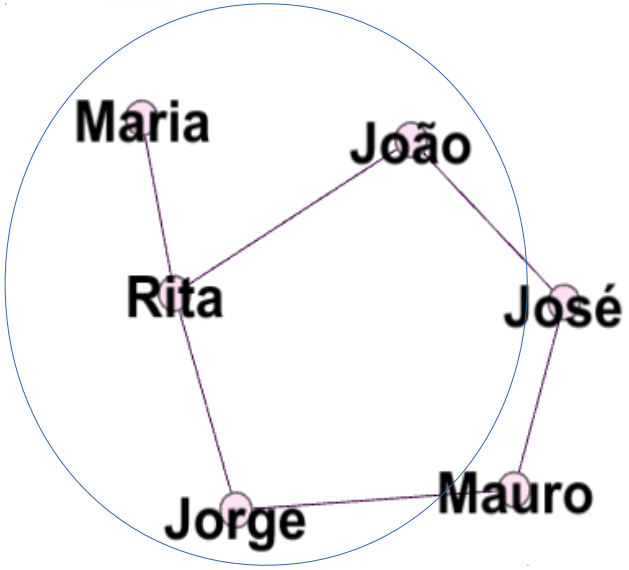
```java
Grafo grafo = new Grafo();
grafo.adicionarAresta(mariaRita);
grafo.adicionarAresta(ritaJoao);
grafo.adicionarAresta(ritaJorge);
grafo.adicionarAresta(joaoJose);
grafo.adicionarAresta(joseMauro);
grafo.adicionarAresta(jorgeMauro);
```

Criando o Grafo

Resumindo ...

- Para construir um Grafo em Java
  - Cria-se as instâncias dos nós
  - Cria-se as instâncias das arestas
  - Cria-se o Grafo
    - O Grafo possui arestas
    - E as arestas contêm os nós
- O que podemos fazer inicialmente com o Grafo?
  - Caminhar na rede → Visitar nós

Vizinhos



Algoritmo: retorna vizinhos

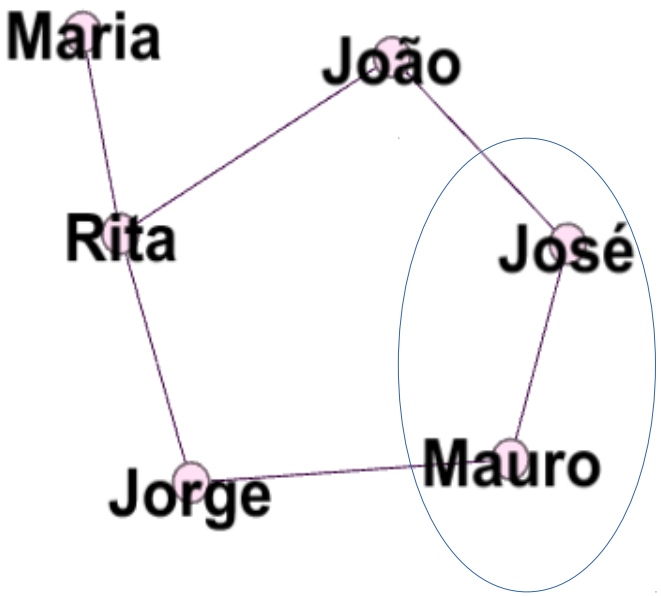
Entrada: qualquer no do grafo
Ideia:
- Percorrer a lista de arestas
- Se algum nó de uma aresta for igual a entrada, retornar o outro nó da aresta

```
public class Grafo {
    List<Aresta> listaAresta;

    public List<No> vizinhos(No no) {
        List<No> nosVizinhos = new ArrayList<No>();
        for(Aresta aresta: listaAresta) {
            if(aresta.origem.equals(no)) {
                nosVizinhos.add(aresta.destino);
            } else if(aresta.destino.equals(no)) {
                nosVizinhos.add(aresta.origem);
            }
        }
        return nosVizinhos;
    }
```

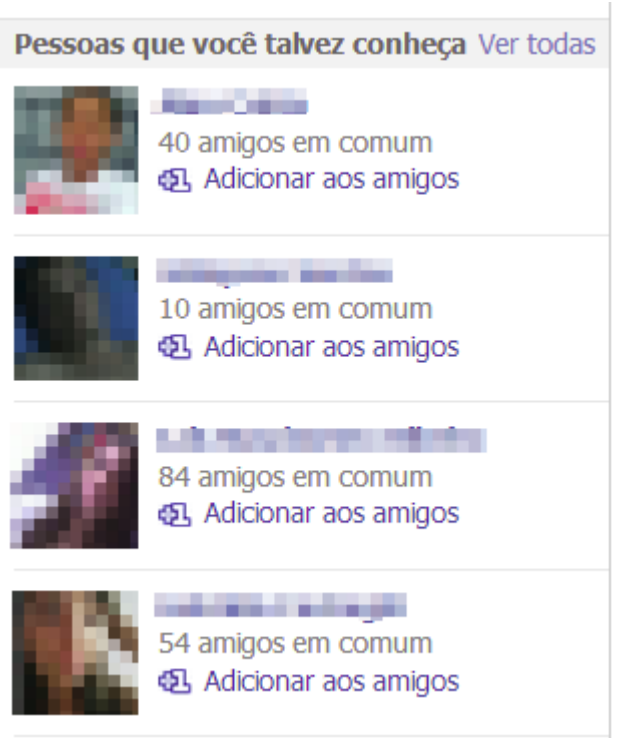
```
List<No> vizinhosDeRita = grafo.vizinhos(rita);
for(No no: vizinhosDeRita) {
    System.out.println(no.nome);
}
```

Recomendando Amigos



Sugestões de amizades para Rita?

Que tal recomendar os amigos dos amigos de Rita.




```
List<No> recomendacaoAmizade = new ArrayList<No>();
List<No> vizinhosDeRita = grafo.vizinhos(rita);
for(No vizinhoRita: vizinhosDeRita) {
    List<No> vizinhosDoVizinhoDeRita = grafo.vizinhos(vizinhoRita);
    for(No no: vizinhosDoVizinhoDeRita) {
        if(!rita.equals(no)) {
            recomendacaoAmizade.add(no);
            System.out.println(no.nome);
        }
    }
}
```

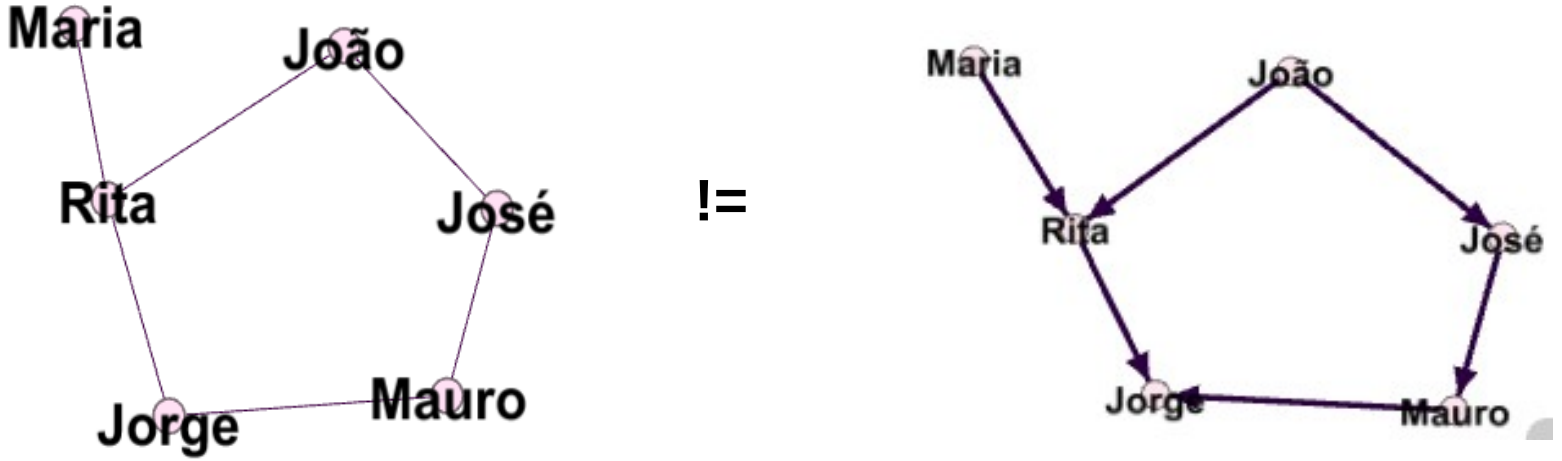
Praticando

- Usando Java, leia o arquivo
- Cada linha representa uma aresta
- Cada número representa o id de um nó
- Ex.: nó 50 se conecta com nó 52
- Crie o Grafo, com as arestas e nós
- Implemente o método ‘vizinhos’
- Encontre os vizinhos do vizinho
- Crie testes unitários para comprovar que sua implementação está correta

origem	destino
50	52
50	68
52	76
52	90
53	70
53	90
54	61
54	67
54	84
54	95

Para a próxima aula ...

- Grafos como Matrizes → Matriz de adjacência
- Grafo direcionado e não direcionado



- Propriedades
  - Grau de um nó (entrada e saída)
  - Pesos das arestas
  - Caminho
- Caminhando no Grafo → Introdução Busca em Largura

Dúvidas?