UNIVERSIDADE DO SUL DE SANTA CATARINA

GUILHERME MARTINS ALVAREZ

CRIAÇÃO E UTILIZAÇÃO DE UMA BASE DE DADOS ORIENTADA A GRAFOS: UM ESTUDO DE CASO SOBRE REDE SOCIAL

Palhoça

2013

GUILHERME MARTINS ALVAREZ

CRIAÇÃO E UTILIZAÇÃO DE UMA BASE DE DADOS ORIENTADA A GRAFOS: UM ESTUDO DE CASO SOBRE REDE SOCIAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Flavio Ceci, M. Eng.

Palhoça

2013

GUILHERME MARTINS ALVAREZ

CRIAÇÃO E UTILIZAÇÃO DE UMA BASE DE DADOS ORIENTADA A GRAFOS: UM ESTUDO DE CASO SOBRE REDE SOCIAL

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Graduação em Ciência da Computação da Universidade do Sul de Santa Catarina.

Palhoça, 20 de novembro de 2013.

Professor e orientador Prof. Flavio Ceci, M. Eng.
Universidade do Sul de Santa Catarina

Prof. Aran Bey Tcholakian Morales, Dr. Eng.
Universidade do Sul de Santa Catarina

Prof. Ricardo Villarroel Dávalos, Dr. Eng.
Universidade Federal de Santa Catarina

Dedico este trabalho a meus pais, Glenio e Iara, que foram os maiores incentivadores à minha formação acadêmica, tornando-se assim, responsáveis por esta conquista.

"Não desanime, em razão da crítica. Se a censura é serviço cabível de qualquer um, a realização elevada é obra de poucos." (André Luiz).

RESUMO

Com popularização dos sistemas de informação, a quantidade de dados armazenados tem crescido diariamente, analisar esse grande volume de dados se tornou um requisito importante de negócio. A necessidade de se obter o melhor desempenho ao menor custo está cada vez mais presente no dia a dia dos desenvolvedores de sistemas. Dessa forma, o grande desafio é desenvolver novas tecnologias e modelos de dados que consigam armazenar e analisar de forma eficiente a gigantesca massa de dados gerada nos dias de hoje. Tendo como base esse cenário de mudanças, este trabalho tem por objetivo modelar e implementar uma base de dados orientada à grafos, a fim de identificar suas diferenças em relação a abordagem relacional. Para isso, foi desenvolvido um estudo de caso baseado em análise social onde bases de dados foram carregadas com dados obtidos através do Facebook, a fim de verificar o desempenho dos dois modelos de dados em consultas recursivas. Os resultados apresentado pelos testes de performance foram satisfatórios fornecendo as informações necessárias sobre as principais diferenças na utilização de bases de dados relacionais e as orientadas a grafos para a análise de redes sociais e *Big Data*.

Palavras-chave: Modelagem Orientada a Grafos. Modelagem Relacional. Banco de Dados. Teoria de Grafos. Big Data.

ABSTRACT

With the popularization of information systems, the amount of data stored has been risen up daily, to analyse this huge volume of data has become an important requirement of business. The need to get the best performance at the lowest cost is increasingly present in the daily life of system developers. In this way, the great challenge is to develop new technologies and data models that are able to store and analyze efficiently the huge amount of data created nowadays. Based on this changing scenario, this paper aims to role a model and implement a graph database, in order to identify their differences to the relational approach. For this, a study of case was developed based on social analysis which databases were loaded with data obtained from Facebook, in order to verify the development of the two data models in recursive queries. The results showed by performance tests were satisfied, giving the necessary information about the main differences in the use of relational databases and graph databases for social network analysis and Big Data.

Keywords: Graphs-Oriented Modeling. Relational Modeling. Database. Graph Theory. Big Data.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Com a popularização dos sistemas de informação, a expansão de suas bases de dados e a necessidade de se obter o melhor desempenho ao menor custo, aumentou o nível de exigência e expectativa dos consumidores de tecnologia. Dessa forma, a inovação e a criatividade têm sido pontos chave nessa busca de se criar novas ferramentas, paradigmas e tecnologias capazes de facilitar cada vez mais as nossas vidas.

Para Cleve, Mens e Hainaut (2010, p. 1, tradução nossa) "a identificação de mudanças de requisitos, a sua tradução em mudanças no sistema, bem como a aplicação e implantação da segunda são chamados coletivamente de evolução do sistema".

A quantidade de dados armazenados em sistemas de informação tem crescido diariamente e analisar esse grande volume de dados se tornou um requisito importante de negócio. Segundo Eaton et al. (2012, tradução nossa), o termo *Big Data* é pode ser aplicado a um grande volume de dados que não podem ser processados ou analisados utilizando as ferramentas e processos tradicionais. Para Hurwitz et al. (2013, tradução nossa), gerenciamento e análise desses dados oferece grandes benefícios para as organizações de todos os tamanhos e setores, pois pode fornecer novas informações e responder questões relacionadas a sua área de negócios. Portanto, para se conservar no mercado e apresentar uma vantagem competitiva, a análise de informações implícitas nessa massa de dados se tornou algo muito importante nos dias de hoje. (CORRIGAN et al., 2013, tradução nossa).

Segundo Corrigan et al (2013), muitos analistas do campo da gestão de informação têm trabalhado para traduzir dados estruturados e não estruturados em conhecimentos úteis. Essas informações, estruturadas e não estruturadas, demandam um esquema diferente do modelo relacional. De acordo com Vicknair et al. (2010), utilizar um novo paradigma de banco de dados, como banco orientado à grafos, pode ser uma solução. Pois o volume de dados e seus relacionamentos não são um impeditivo para um ótimo desempenho.

Este trabalho propõe a utilização de bases de dados orientadas a grafos para o armazenamento de redes complexas de dados, como as encontradas em redes sociais e *Big Data*.

Neste capítulo são abordados o problema de pesquisa, os objetivos deste trabalho, justificativa e a estrutura deste trabalho.

1.1 PROBLEMA DE PESQUISA

O modelo relacional vem sendo utilizado há décadas para representar as relações entre os dados, facilitando a sua organização e armazenamento. A maior parte das aplicações disponíveis no mercado utiliza esse tipo de modelagem, por ser a mais tradicional e confiável forma de persistência de dados.

Conforme o aumento da massa de dados armazenada ao longo dos anos, os especialistas em banco de dados passaram a identificar alguns pontos de conflito nas bases que utilizam o modelo relacional. Para Angles e Gutierrez (2008, p. 7, tradução nossa), "o modelo relacional foi direcionado para o simples registro de tipos de dados com uma estrutura conhecida com antecedência. O esquema é fixo e extensibilidade é uma tarefa difícil".

Com a inclusão digital e o avanço expressivo das tecnologias de informação, há uma necessidade cada vez maior de se registrar novos tipos de dados, tornando os repositórios de dados mais ricos e complexos. No entanto, as bases de dados desenvolvidas segundo o modelo relacional, não possuem um esquema flexível no qual seja possível adicionar facilmente novos relacionamentos entre os objetos. Os dados e suas estruturas sempre mudam, portanto verificou-se que o modelo de dados está evoluindo, mas o esquema acaba não acompanhando essa evolução. Desse modo, o modelo relacional acaba se tornando uma barreira, pois impõe uma resistência a mudanças rápidas que os sistemas atuais necessitam.

Como há uma dificuldade de replicar alguns tipos de conexões entre objetos nos bancos de dados relacionais, em alguns casos, os desenvolvedores acabam tendo que mudar o modelo de domínio da aplicação para se adequar ao modelo físico.

> Em um banco de dados relacional, trabalhamos contra a corrente. Temos esquemas muitas vezes frágeis que rigidamente descrevem a estrutura dos dados apenas para ter um comportamento destinado àqueles esquemas substituídos pelo código, que aparentemente obedece às leis, mas dobram todas as regras. (ROBINSON, WEBBER e EIFREM, 2013, p. 32, tradução nossa).

Outro ponto a ser avaliado é o desempenho das consultas realizadas nos bancos de dados. Assim como a rede de dados tem se tornado mais complexa, algumas consultas envolvendo os objetos armazenados estão apresentando uma deficiência no tempo de resposta. No modelo relacional, a performance das consultas é diretamente proporcional ao número de relacionamentos envolvidos e tamanho da base de dados, pois o SGBD (Sistema Gerenciador de Banco de Dados) pesquisa através de todos os dados para verificar quais satisfazem os critérios estabelecidos na busca. Então, consultas que envolvem cálculos

relacionais e relacionamentos recursivos, acabam se tornando computacionalmente complexas devido à estrutura de organização do modelo relacional.

Para Hurwitz et al. (2013, tradução nossa), além dos dados estruturados armazenados nas bases de dados relacionais, existem informações não estruturadas, conhecidas como *Big Data*, provenientes de blogs, e-mails, mídias sociais, sensores, registros de transações de compras, fotos e outros. Essas fontes contém um volume de dados muito expressivo e difícil de armazenar e organizar na estrutura tradicional de bancos relacionais. Há uma quantidade enorme de registros que poderiam ser analisados para se encontrar novas relações e ideias em novos tipos de dados, mas os esquemas relacionais de dados impedem que essa verificação seja feita devido ao alto custo que apresentam.

Desse modo, chega-se a seguinte questão: a utilização de banco de dados orientado a grafos pode ser uma boa alternativa para trabalhar com uma grande quantidade de dados?

1.2 OBJETIVOS

Esta seção é reservada a apresentar os objetivos gerais deste trabalho e seus objetivos específicos.

1.2.1 Objetivo Geral

Modelar e implementar uma base de dados orientada à grafo, a fim de identificar suas diferenças em relação a abordagem relacional.

1.2.2 Objetivos Específicos

Os objetivos específicos são:

a) Identificar ferramentas computacionais para apoiar a implementação da abordagem orientada a grafos;

b) Formular um estudo de caso baseado em Análise Social;

c) Modelar e avaliar o cenário do estudo de caso utilizando modelagem relacional e orientada a grafos;

d) Documentar os resultados e constatações obtidas.

1.3 JUSTIFICATIVA

Para Angles (2012, tradução nossa), muitos estudos estão sendo realizados para avaliar o custo-benefício de se utilizar novos modelos de banco de dados e algumas empresas estão optando pelo desenvolvimento de seus próprios modelos de dados.

Grafos são estruturas de dados que podem ser utilizadas em diversas áreas de interesse e são muito úteis para modelagem de interações e objetos. Ultimamente, tem havido um interesse em grafos para representar as redes sociais. Como os bancos de dados orientados a grafos são baseados em estruturas nas quais a conexão entre os dados é uma característica do próprio modelo de dados, utilizar um banco de dados orientado a grafos traz algumas vantagens.

O desempenho é uma das principais preocupações dos engenheiros de software em relação aos seus sistemas. Dessa forma, sistemas web e aplicativos online devem responder rapidamente para se tornarem um sucesso comercial. Segundo Robinson, Webber e Eifrem (2013, tradução nossa) usando o índice livre de adjacência, um banco de dados orientado a grafos torna um relacionamento complexo em travessias rápidas no grafo,

mantendo assim um alto desempenho independentemente do tamanho total do conjunto de dados. Portanto, consultas que seriam complexas dentro de um cenário com banco de dados relacional se tornam simples e rápidas em um banco de dados orientado a grafos, apresentando um maior custo-benefício independente do tamanho da base de dados.

A análise de grandes volumes de dados, *Big Data*, é complexa e demorada utilizando o modelo relacional devido ao alto nível de processamento necessário, mas se torna mais eficiente em um esquema de dados que utiliza grafos, pois além de facilitar a identificação e modelagem de dados não estruturados, a estrutura livre de índice facilita nas consultas de alto desempenho e é um aspecto importante no design e na maneira que os dados são armazenados. (ROBINSON; WEBBER; EIFREM, 2013, tradução nossa). Esse formato de modelagem de dados suporta travessias extremamente rápidas utilizando algoritmos de busca em grafos. Dessa forma, grandes repositórios de dados que contem informações valiosas para as grandes companhias, podem ser mensurados e estudados de uma maneira eficaz.

As alterações no modelo de dados deixam de ser um problema, pois o modelo orientado a grafos aproxima o modelo de domínio ao modelo do banco de dados. Assim, as inclusões de novos objetos e interações se tornam simples, acelerando os ciclos de desenvolvimento de software.

Para Batra e Tyagi (2012, p. 511, tradução nossa) “em bases de dados orientadas a grafos, não há necessidade para reestruturar o esquema completo cada vez que um novo relacionamento é adicionado, apenas algumas arestas e os nós são adicionados ao grafo”.

Estruturar uma base de dados orientada a grafos é simples e se encaixa com a maneira geralmente utilizada para abstrair, especificar e modelar os dados. De forma que, além de expressar a forma como vemos os objetos conectados, ela também pode mostrar claramente os tipos de perguntas que se quer fazer ao esquema de dados. Pois os relacionamentos fazem parte das propriedades nativas do modelo de dados.

1.4 ESTRUTURA DO TRABALHO

Este trabalho esta estruturado da seguinte forma: o primeiro capítulo apresenta uma introdução ao assunto, descrevendo o problema de pesquisa, o objetivo geral, os objetivos específicos e a justificativa.

O capítulo dois tem como finalidade apresentar o referencial teórico, dando foco aos principais temas que norteiam a construção deste trabalho: grafos, modelagem de dados orientada à grafos, modelagem de dados relacional, análise social e *Big Data*.

O capítulo três é focado no método que guia o desenvolvimento da pesquisa e desenvolvimento bem como apresentar também as delimitações deste trabalho.

O capítulo quatro apresenta detalhes sobre a modelagem orientada a grafos e informações sobre o ferramental computacional disponível para esta abordagem.

O capítulo cinco é centrado no estudo de caso, que é focado na análise social sobre dados disponíveis em redes sociais, onde além do estudo de caso são apresentados os resultados obtidos pela avaliação. Por fim o capítulo seis apresenta as conclusões e os trabalhos futuros.

2 REFERENCIAL BIBLIOGRÁFICO

O seguinte capítulo descreve alguns elementos necessários para o entendimento deste trabalho. São abordados, primeiramente, conceitos básicos relacionados a grafos: conceito de grafos, referencial histórico, teoria de grafos e aplicações de grafos. Em seguida são abordadas informações relevantes sobre modelagem de dados e são apresentados alguns modelos de modelagem dados como: relacional, orientada a grafos, dimensional, orientada a objeto e orientada a documentos. Posteriormente, são aparentados conceitos relacionados aos estudos de análise social. Para concluir, é abordado o conceito de *Big Data* e suas principais características.

2.1 BIG DATA

Antes de tentar entender o que é *Big Data*, devemos entender por que ele é importante para os negócios. A busca de *Big Data* está diretamente ligada à análise de dados e é não podemos ignorar o impacto que a análise de dados teve em organizações durante a última década. Instituto da IBM Business Value publicou os resultados de um estudo em um artigo chamado The New Intelligent Enterprise, no qual concluiu que as organizações que realizam a analise de dados para obter vantagem competitiva, possuem duas vezes mais chances de superar seus concorrentes diretos da indústria. (CORRIGAN et al., 2013, tradução nossa).

Gerenciamento e análise de dados sempre ofereceram grandes benefícios para as organizações de todos os tamanhos e em todos os setores. As empresas têm lutado muito para encontrar uma abordagem pragmática para a captura de informações sobre seus clientes, produtos e serviços. Quando as empresas possuíam poucos clientes, que todos compraram sempre os mesmos produtos, as coisas eram bastante simples. Mas ao longo do tempo, as empresas e o mercado têm crescido e se tornado mais complexo. O grande desafio hoje é

realizar cruzamento de todos os diferentes tipos de dados obtidos por diversos meios. (HURWITZ et al., 2013, tradução nossa).

Os dados não estão se tornando somente mais acessíveis, estão mais compreensíveis para os computadores. Portanto, dados não estruturados que normalmente não são o objetivo dos bancos de dados tradicionais estão sendo priorizados por novas ferramentas de busca e analise de dados da era da internet. Nesse meio, as técnicas de inteligência artificial, como processamento e reconhecimento de padrões de linguagem, estão crescendo cada vez mais e expandindo seus horizontes. (LOHR, 2012, tradução nossa).

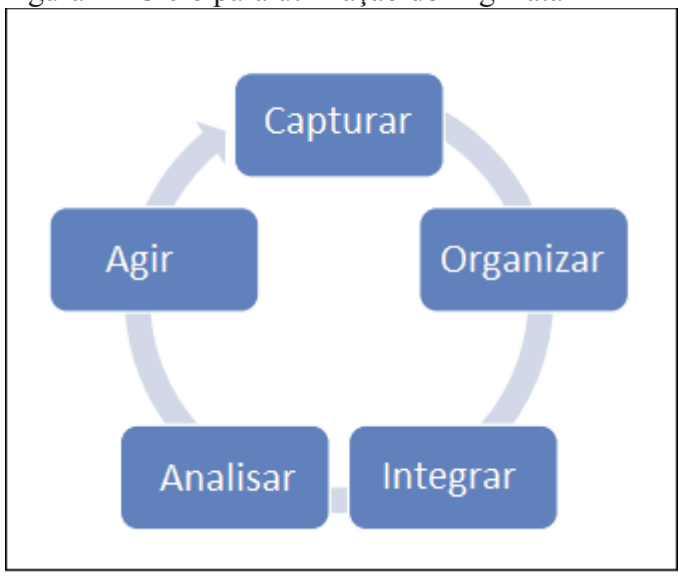
Segundo Hurwitz et al. (2013, tradução nossa) cada onda de gerenciamento de dados nasceu da necessidade de tentar resolver um tipo específico de problema. Quando as empresas começaram a armazenar dados não estruturados, os analistas precisavam de novas tecnologias como ferramentas de análise para obter visões que seriam úteis ao negócio.

Big Data é uma massa de dados muito grande para ser manipulado e analisado por protocolos de banco de dados tradicionais, como SQL. O volume, variedade e velocidade de dados disponíveis aumentou a complexidade necessária para as empresas gerenciarem essas informações. Estamos em uma época que a quantidade de dados gerados no mundo é medida em *exabytes* e *zettabytes*. Além disso, a variedade de fontes disponíveis e tipos de dados a serem gerados se expandem tão rapidamente quanto as novas tecnologias são criadas. (DAVIS; PATTERSON, 2012, tradução nossa).

De acordo com Eaton et al. (2012, tradução nossa) o volume de dados armazenados por uma rede social é gigantesco. Como exemplo pode-se citar o Twitter que gera mais de 7 *terabytes* de dados todos os dias, o Facebook que gera mais de 10 *terabytes* por dia e algumas empresas geram *terabytes* de dados a cada hora do dia, durante todos os dias do ano.

Para Boyd e Crawford (2011, tradução nossa) *Big Data* é importante, pois é um fenômeno analítico que mexe com a academia e a indústria. Esse tipo de dados estimula a prática de apofenia, a tendência de ver padrões onde eles existem, pois essa grande massa de dados pode oferecer conexões diversas em todas as direções. O *Big Data* não é mais somente um domínio de cientistas e grandes pesquisadores, as novas tecnologias os tornaram acessíveis para as pessoas incluindo acadêmicos de ciências sociais, comerciantes, organizações governamentais, instituições educacionais e indivíduos motivados a produzir, compartilhar, interagir e organizar os dados.

Figura 1 - Ciclo para utilização do Big Data



Fonte: O Autor (2013)

A Figura 1 representa o ciclo de utilização do *Big Data*. Os dados devem ser previamente capturados, e, em seguida, organizados e integrados. Após esta fase é os dados podem ser analisados com base na questão a ser abordada. Finalmente, os gestores do negócio tomam as medidas necessárias com base no resultado da análise. (HURWITZ et al., 2013, tradução nossa).

De acordo com Corrigan et al. (2013, tradução nossa), um importante diferencial entre as corporações é a capacidade de absorver e analisar dados que estão sendo desperdiçados. Estes tipos de dados podem produzir novas ideias e resultados incríveis. Esses dados são gerados em grandes quantidades, mas normalmente não são aproveitados nas áreas de negócio. Como exemplo, temos lojas online não conseguem captar todos os *terabytes* de *clickstreams* gerados em seus web sites. Essas informações podem ser analisadas a fim de otimizar a experiência de compra de seus visitantes e, dessa forma, entender o porquê cestas de compras com produtos estão sendo abandonadas e a compra não é finalizada. Dessa forma, uma enorme quantidade de dados poderia ser coletada e analisada para avaliar o desempenho de redes de vitais para os governos. Os arquivos de log de suas redes poderiam ser analisados verificando tendências quando algo deu errado para encontrar o ponto chave que indica problemas.

Para Hurwitz et al. (2013, tradução nossa) as empresas almejam que os dados armazenados sejam usados para responder a perguntas relacionadas a tomada de decisões. Com o aparecimento do *Big Data*, estamos presenciando o desenvolvimento de aplicativos que são projetados especificamente para tirar proveito das características únicas desse

fenômeno. Algumas dessas aplicações emergentes são voltadas a áreas como saúde, gestão de produção, gestão de tráfego, e assim por diante. Essas aplicações contam com algo em comum, os grandes volumes e variedades de dados para serem analisados. Na saúde, um sistema ser capaz de monitorar bebês prematuros e determinar quando uma intervenção é necessária. Na indústria, um sistema de dados pode ser usado para controlar a produção e minimizar seus custos. Na gestão de tráfego podemos reduzir o número de congestionamentos em rodovias movimentadas e diminuir os acidentes.

A maneira como lidamos com o surgimento de uma era de *Big Data* é fundamental, pois esse fenômeno está ocorrendo em um ambiente de mudanças rápidas e de muitas incertezas, e as decisões atuais impactarão no nosso futuro. (BOYD; CRAWFORD, 2011, tradução nossa).

Segundo Davis e Patterson (2012, tradução nossa), ao mesmo tempo em que as empresas estão entusiasmadas com os potenciais benefícios da criação e desenvolvimento de novos produtos e serviços a partir de ideias extraídas do *Big Data*, essa nova gama de informações disponíveis levanta novas questões. Algumas dessas perguntas são relacionadas às implicações do armazenamento e utilização de dados relacionados diretamente a pessoas, comportamentos, preferencias, locais frequentados e seus relacionamentos. Basicamente, estas questões éticas devem ser levantadas e respondidas ao se aplicar e utilizar esses dados, pois a promessa de novos benefícios pode trazer riscos e consequências não intencionais.

2.2 GRAFOS

Os grafos são estruturas de dados dotados de um poder tão expressivo que tornam a sua utilização rentável nas mais díspares áreas. Ao atribuir um significado adequado para os vértices e arestas de um grafo, é possível alcançar representações completas, de interesse em muitos domínios de aplicação, que vão desde a área científica à área humanística. (FOGGIA; SANSOME; VENTO, 2001, tradução nossa).

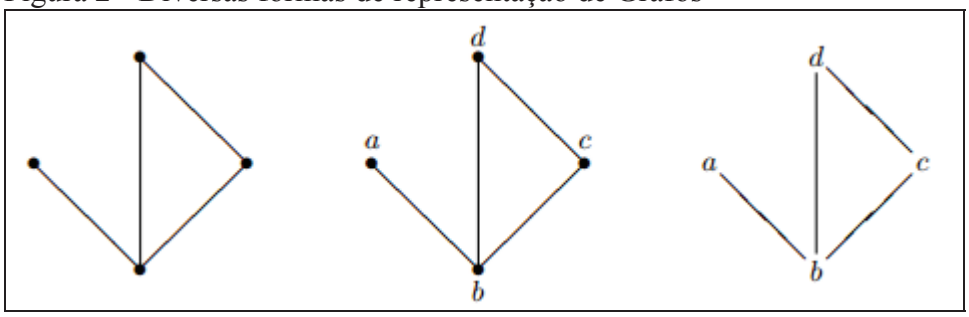
Convencionalmente, um grafo é apenas uma coleção de vértices e arestas, ou um conjunto de nós e as relações que os conectam. Grafos podem ser usados para modelar todos os tipos de cenários, desde a construção de um foguete espacial, a um sistema de estradas, e da cadeia de fornecimento de energia, e mais além. Os grafos são de uso geral e expressivo, o

que nos permite modelar as entidades como vértices e seus contextos semânticos usando as arestas. (ROBINSON; WEBBER; EIFREM, 2013, tradução nossa).

Segundo Boaventura Netto (2001), um grafo é uma estrutura $G = (X, U)$, onde X é um conjunto discreto e U é uma família cujos elementos u (não vazios) são definidos em função dos elementos de x de X. O elemento X é chamado de vértices ou nó e uma família U é a relação de adjacência conhecida como aresta nas estruturas orientadas e ligação nas estruturas não orientadas.

A representação dos grafos pode ser feita de diversas formas. A seguir, apresentamos na Figura 2, três formas diferentes de representação de um mesmo grafo. (SMITH; MARTINS, 2009).

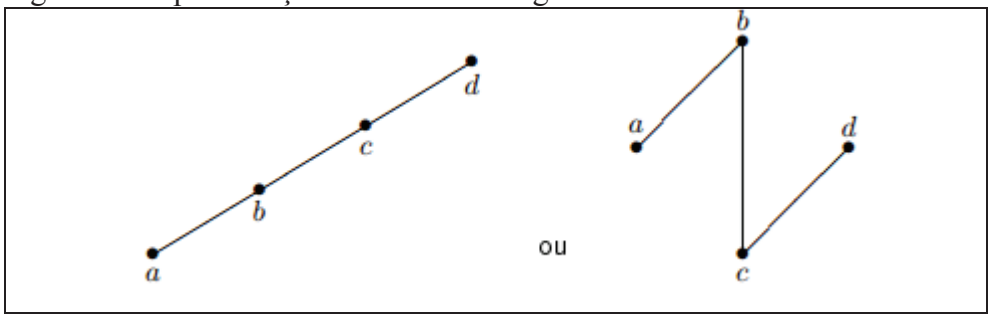
Figura 2 - Diversas formas de representação de Grafos



Fonte: Smith e Martins (2009, p.5)

Segundo Smith e Martins (2009), existem representações aparentemente diferentes de um mesmo grafo, como apresentado na Figura 3. Numa representação de um grafo, o importante é o número de vértices, número de arestas e o modo como estas se dispõem em relação aos vértices.

Figura 3 - Representações de um mesmo grafo

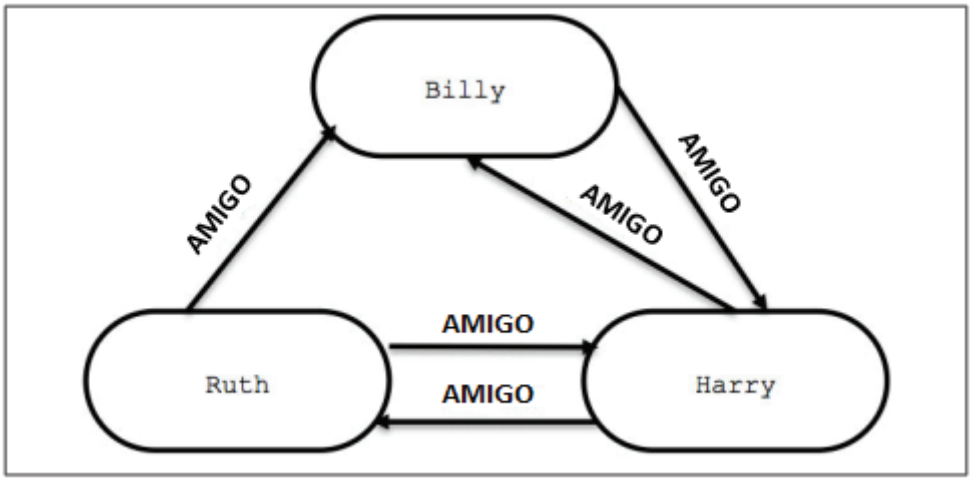


Fonte: Smith e Martins (2009, p.6)

Os grafos são geralmente representados por diagramas nos quais cada vértice é representado por um ponto ou um círculo pequeno (aberto ou sólido) e cada aresta é representada por um segmento de reta ou curva que une os pequenos círculos correspondentes. (CHARTRAND; ZHANG, 2009, tradução nossa).

Sousa (2010), afirma que grafos podem ser considerados uma representação abstrata de uma rede, em que os conceitos de distância, localização, orientação, forma e comprimento são substituídos por propriedades topológicas, como: acessibilidade, adjacência, centralidade, ligação e conectividade, e, a partir das quais, mediante algoritmos e o cálculo de matrizes é possível estudar os seus componentes (vértices e arestas) ou estudar a rede como um todo.

Figura 4 - Um pequeno grafo social



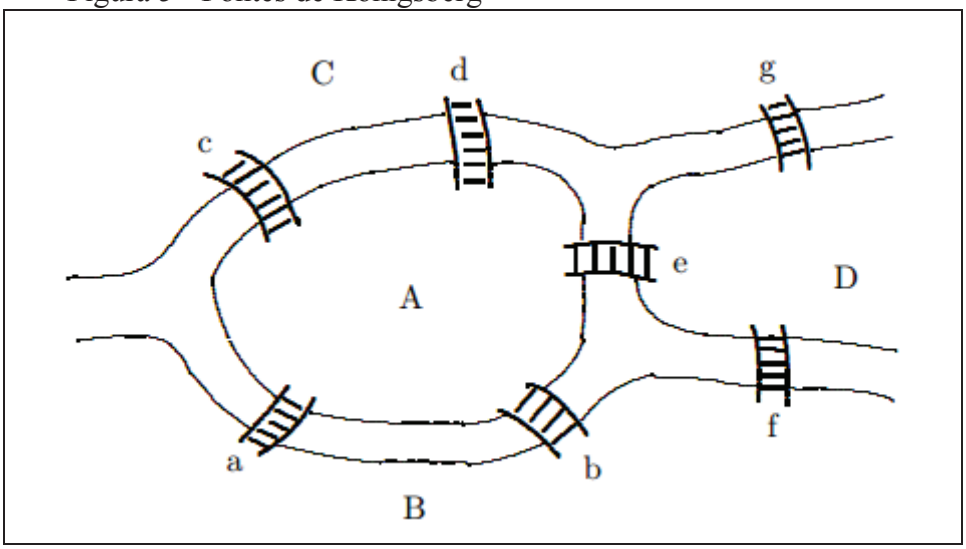
Fonte: Robinson, Webber e Eifrem (2013, p.5)

Por exemplo, os dados de uma rede social, como o Facebook, são facilmente representados em um grafo. Na Figura 4, apresenta-se uma pequena rede de amigos. Os relacionamentos são a chave deste contexto semântico: Billy considera Harry um amigo, e Harry, por sua vez, também considera Billy um amigo. Ruth e Harry também expressaram sua amizade mútua, mas, infelizmente, enquanto Ruth é amiga de Billy, Billy não corresponde essa amizade. O grafo verdadeiro de uma rede social é centenas de milhões de vezes maior do que o exemplo na Figura 4, mas funciona precisamente sobre os mesmos princípios. (ROBINSON, WEBBER e EIFREM, 2013, tradução nossa).

2.2.1 Histórico

Segundo Ostrorski e Menoncini (2009), a teoria dos grafos teve inicio no ano de 1736, quando Leonhard Euler se deparou com o problema das pontes de Königsberg. Königsberg era uma cidade da antiga Prússia, atual Kaliningrado (Rússia), onde havia duas ilhas que eram ligadas por sete pontes junto à parte continental. (Figura 5)

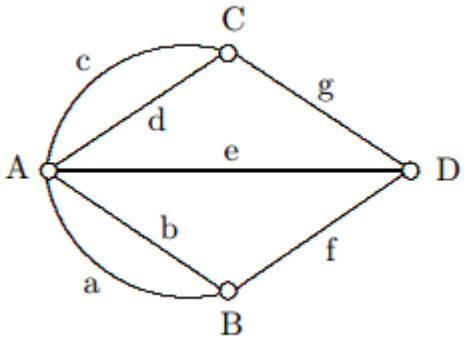
Figura 5 - Pontes de Königsberg



Fonte: Chartrand e Zhang (2009, p.73)

De acordo com Recuero (2006), conta-se que, na época, era uma diversão para seus habitantes tentar solucionar o problema de atravessar a cidade através das sete pontes, cruzando uma única vez por cada ponte. Euler demonstrou que cruzar as sete pontes sem repetir um caminho era impossível. Para isso, ele conectou as quatro partes terrestres (vértices) com as sete pontes (arestas), mostrando a inexistência da concernida rota e elaborando o primeiro teorema da teoria dos grafos. (Figura 6).

Figura 6 - Representação gráfica das Pontes de Königsberg



Fonte: Chartrand e Zhang (2009, p.74)

Segundo Szwarcfiter (1984), outro ponto importante da história dos grafos foi a formulação do problema de ciclo Hamiltoniano, por Hamilton. O problema consiste em determinar um trajeto que passe somente uma vez em cada vértice e retorne ao ponto inicial.

No século XIX, Kirchoff e Cayley, conceberam a teoria de árvores. Cayley desenvolveu árvores a partir do estudo da química orgânica, enquanto Kirchoff estudava as redes elétricas. (RABUSKE, 1992).

Boaventura Netto (2001) ressalta que pela pouca importância dessa análise das pontes de Königsberg diante das magnificas produções de Eüler, os estudos das relações dos conjuntos discretos só vieram a se tornar objeto atenção já no século XX, com a publicação, em 1936, do primeiro livro sobre a teoria dos grafos — *Theorie der Endlichen und Unendlichen Graphen*, de Denes König.

Ainda segundo Boaventura Netto (2001), desde o I Simpósio Brasileiro de Pesquisa Operacional realizado em 1968, pesquisadores de diversas universidades brasileiras tem apresentado trabalhos envolvendo teoria de grafos e aplicações que utilizam grafos.

2.2.2 Teoria de Grafos

Tobler (1970, apud SOUSA, 2010), afirma que a teoria dos grafos é o ramo da matemática que estuda a topologia das redes, que comprovou ser um importante método de análise de situações em que os fenômenos verificados estabelecem relações entre si.

Segundo Rabuske (1992), a teoria dos grafos apresenta um instrumento simples, acessível e poderoso para a modelagem e resolução de problemas relacionados a arranjos de objetos discretos.

A teoria de grafos estuda os objetos combinatórios, que são um bom modelo para muitos problemas em vários ramos, como matemática e informática. (FEOFILOFF; KOHAYAKAWA; WAKABAYSHI, 2011).

Um vértice sem arestas incidindo sobre o mesmo é dito isolado e dois vértices ligados por uma aresta são chamados de adjacentes. (RABUSKE, 1992).

O número máximo de ligações é de 2^n para um grafo não orientado e de n^2 para um grafo orientado. A relação entre o número de ligações e o maior número de ligações possível, é chamada de densidade. (BOAVENTURA NETTO, 2001).

Boaventura Netto (2001), explica que, há diferentes formas de representação matricial de grafos e sua utilização está ligada à necessidade de realizar cálculos envolvendo os dados do grafo.

A matriz de adjacência $M(G)$ é a matriz $n \times n$ na qual M_{ij} é o número de conexões unindo os vértices i e j. Computacionalmente essa representação tem o inconveniente de que a matriz despende uma área de memória expressiva, ela ocupa n^2 posições de memória. (BRAGA;GOMES;RUEDIGER, 2008).

A matriz de incidência é uma matriz de dimensões $n \times m$, onde cada linha corresponde a um vértice e cada coluna a uma relação. Ela é utilizada na produção de modelos de programação matemática envolvendo grafos, determinação de grafos adjuntos e de inserção. (BOAVENTURA NETTO, 2001).

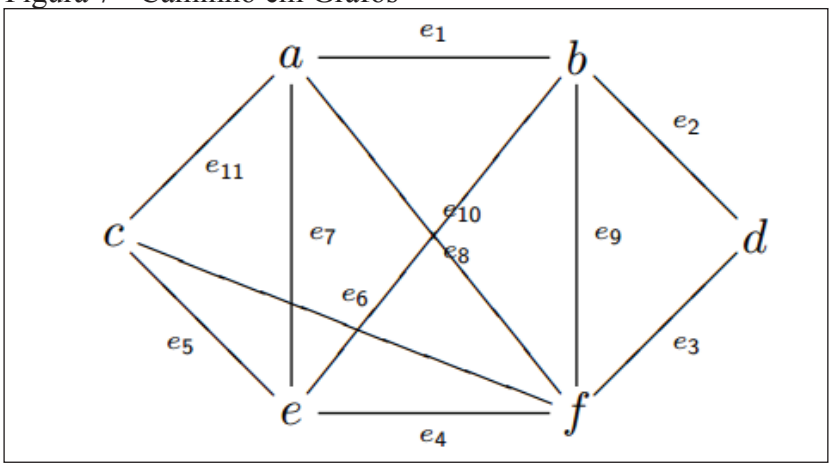
Para Boaventura Netto (2001), um conceito fundamental é a conexidade, que em grafos, está relacionada à possível passagem de um vértice a outro através das ligações dos vértices. Ainda que, não seja fácil desenvolver algoritmos destinados a explorar um grafo no que se diz respeito à conexidade, a técnica de busca em profundidade pode ser utilizada para desenvolver esses algoritmos.

De acordo com Rabuske (1992), caminho pode ser entendido como uma sequência de arestas onde o vértice final de uma aresta é o vértice inicial da próxima. Caso todos os vértices do caminho sejam distintos, então essa sequência é chamada de caminho simples.

Um caminho é qualquer grafo da forma $(\{v_1, v_2,..., v_n\},\{v_i\ v_{i+1}: 1 \leq i < n\})$. Em outras palavras, um caminho é um grafo G cujo conjunto de vértices admite uma permutação

$(v_1, v_2,..., v_n)$ tal que $\{v_1v_2, v_2v_3,..., v_{n-1}v_n\} = A(G)$. (FEOFILOFF; KOHAYAKAWA; WAKABAYSHI, 2011)

Figura 7 - Caminho em Grafos



Fonte: Smith e Martins (2009, p.10)

No grafo apresentado na Figura 7, o caminho {a,b,d,f,c,e,f,b} pode também ser representado por {e1,e2,e3,e6,e5,e4,e9}.(SMITH; MARTINS, 2009).

A respeito da ideia de caminho, Boaventura Netto (2001), considera que a importância do caminho, e a definição de distância agregada ao valor mínimo de um caminho, colocam a categoria de problemas de menor caminho em uma posição de destaque na teoria de grafos. As aplicações dos algoritmos envolvem problemas relacionados à tomada de decisões envolvendo um custo a ser minimizado.

Para solucionar problemas onde precisamos determinar o caminho entre dois vértices, podemos utilizar os algoritmos de Floyd ou de Dijkstra. (RABUSKE, 1992)

A ideia de árvores foi concebida em 1847 por Kirchhoff a fim de resolver o sistema de equações lineares simultâneas que dão corrente a cada ramo e circuito de uma rede eléctrica. Em 1857, Cayley desenvolveu a ideia de árvores considerando as mudanças de variáveis no cálculo diferencial na área da química. (VASUDEV, 2006, tradução nossa).

Árvore, que é uma ideia fundamental em teoria de grafos, aparece em aplicações de diversas áreas que aparentemente não tem ligação com grafos, como comunicações, redes de energia, esgoto, água, química e etc. (RABUSKE, 1992).

2.2.3 Algoritmos de Caminho Mínimo

Segundo Szwarcfiter e Markenzon (1994) um algoritmo é um processo ordenado para a resolução de um problema. O desenvolvimento de algoritmos é particularmente importante para questões a serem solucionadas em um computador, pela própria natureza da ferramenta utilizada.

A busca visa resolver um problema básico, o de como explorar um grafo. Isto é, através de um grafo, obter um processo sistemático de como caminhar pelos vértices e arestas de um grafo. (SZWARCFITER, 1984).

Existem diversos algoritmos para se determinar o menor caminho ou caminho de menor custo em um grafo. A essência do problema consiste em determinar a distância do menor caminho entre dois vértices v_i e v_j. (RABUSKE, 1992).

Os algoritmos de busca de menor caminho não são utilizados somente nas áreas da matemática e computação, em diversas áreas como transportes, redes de comunicação, projetos de redes de saneamento e etc, pode-se encontrar problemas relacionados a busca de menor caminho. (VASUDEV, 2006).

2.2.3.1 Algoritmo de Busca em Largura

Segundo Szwarcfiter (1984) a busca é dita de largura quando o critério de escolha de vértice marcado obedece a regra na qual dentre todos os vértices marcados e incidentes a alguma aresta ainda não explorada, escolhe-se aquele menos recentemente alcançado na busca.

No algoritmo de busca em largura ou Breath-first search, uma árvore enraizada será construída, formando um grafo não direcionado. A ideia do algoritmo de busca de largura é visitar todos os vértices em um determinado nível antes de ir para o próximo nível. (VASUDEV, 2006).

A busca de largura pode ser implementada com o auxílio de uma fila e da mesma forma que a busca de profundidade, o critério de escolha das arestas é arbitrário. (SZWARCFITER, 1984).

2.2.3.2 Algoritmo de Busca em Profundidade

De acordo com Szwarcfiter (1984) a busca é dita de profundidade quando o critério de escolha de vértice marcado obedece a regra na qual dentre todos os vértices marcados e incidentes a alguma aresta ainda não explorada, escolhe-se aquele mais recentemente alcançado na busca.

Uma alternativa para o algoritmo de busca em largura é o algoritmo de busca em profundidade, que procura passar a níveis sucessivos, em uma árvore, na primeira oportunidade possível. (VASUDEV, 2006).

2.2.3.3 Algoritmo de Dijkstra

O algoritmo de Dijkstra, publicado em 1950, trabalha apenas com valores positivos, o algoritmo determina os caminhos do vértice inicial até os outros vértices e para verificar qual é a distância mínima fazendo a somatória das distâncias das arestas envolvidas. (BOAVENTURA NETTO, 2001).

Para Rabuske (1992), em um grafo G(V,E) e uma função distância L que agregue cada aresta (v,w) a um número real não negativo L(v,w) e também um vértice fixo v_0 em V, busca-se determinar os caminhos de v_0 para cada vértice v de G, de tal forma que a somatória das distâncias das arestas envolvidas em cada caminho seja mínima. Isto é equivalente a determinar o caminho $v_0, v_1, \ldots, v_k$ tal que a soma das distâncias envolvidas seja mínima.

2.2.3.4 Algoritmo de Floyd

Para Boaventura Netto (2001), o algoritmo de Floyd é um algoritmo matricial que pode trabalhar com grafos contendo arcos de valor negativo. Este algoritmo utiliza um vértice de base *k* para a construção de triplas com os pares que serão analisados por desigualdades triangulares.

Neste algoritmo, cria-se uma matriz D^0 de custos das arestas, na qual os laços possuem custo zero e à não existência de arestas aplica-se o custo infinito. O algoritmo de Floyd constrói *n* matrizes a partir de D^0. Para definição do caminho, parte-se do final para o inicio, considerando-se os vértices intermediários incluídos durante o processo. (RABUSKE, 1992).

2.2.4 Aplicações

Rabuske (1992) afirma que há muitas aplicações para grafos. Modelos de grafos podem ser utilizados para resolver problemas que necessitam da construção de sistemas complexos, devido às combinações de seus componentes.

Zhao e Han (2010, tradução nossa) afirmam que nos últimos anos presenciamos uma rápida proliferação de redes, tais como as redes de comunicação, redes biológicas, redes sociais e web, as quais podem ser naturalmente modeladas através de grafos.

Os grafos são importantes modelos da área da matemática e possuem um grande papel em áreas como engenharia e pesquisa operacional. Eles fornecem as ferramentas necessárias para tratar problemas como: caminho mínimo, fluxo máximo, alocação de elementos e etc. (HERNANDES, 2007).

Relações complexas representadas por grafos podem ser utilizadas para identificação de objetos. O uso de modelos baseados em grafos é essencial no reconhecimento

de padrões. Em química e bioinformática, os cientistas usam grafos para representar os compostos e proteínas. Desse modo, os pesquisadores são capazes de fazer o rastreio, design, e descoberta de conhecimento a partir de um composto ou bancos de dados moleculares. (YAN; YU; HAN, 2004, tradução nossa).

Para Graves, Bergeman e Lawrence (1995, tradução nossa), os dados do genoma humano são altamente interconectados e tem uma estrutura complexa que é difícil de replicar em qualquer modelo de dados. Os grafos simplificam o design em representações da área genética e fornecem aos biólogos uma linguagem confortável para descrever a ciência.

Grafos podem ser utilizados para modelar todos os tipos de cenários, desde a construção de um avião, estradas, redes de fornecimento de água e energia, redes de distribuição de alimentos, dados históricos de determinada população, entre outros. (ROBINSON; WEBBER; EIFREM, 2013, tradução nossa).

2.3 MODELAGEM DE DADOS

Date (2000), afirma que um modelo de dados é uma definição abstrata, independente e lógica dos objetos, operadores e outros elementos que compõem o meio com o qual os usuários interagem.

Barbieri (1994), afirma que modelagem de dados é uma atividade desenvolvida em diversas fases do processo de desenvolvimento de software, com o objetivo de encontrar informações para a definição do modelo de dados.

De acordo com Barbieri (1994), a modelagem de dados foi desenvolvida nos anos 70 e suas técnicas são usadas para registrar dados nos mais diversos níveis de abordagem. As fronteiras de seus conceitos não são estritamente definidas, pois o objetivo da modelagem é dar fidelidade a representações da realidade através do computador. Dessa forma, os dados podem assumir várias formas dependendo das regras de negócio e da necessidade dos sistemas.

Alguns dados são estruturados e armazenados em uma base de dados tradicional, enquanto outros dados, como registros da internet, documentos e até mesmo arquivos multimídia como fotos e vídeos, não são estruturados. Empresas também têm que considerar novas fontes de dados, como os gerado por sensores. Outras novas fontes de informação são produzidas pelo ser humano, como dados de mídias sociais e os dados gerados a partir de interações com uma página na web. Além disso, o desenvolvimento de novos dispositivos, juntamente com o acesso generalizado a redes globais irá acarretar na criação de novas fontes de dados. (HURWITZ et al, 2013, tradução nossa).

2.3.1 Modelagem Relacional

O modelo relacional foi concebido por E.F Codd, pesquisador da IBM, que publicou uma série de documentos relacionados a esse modelo. Outros pesquisadores têm expandido e reforçado a pesquisa original de Codd, trazendo o modelo de banco de dados relacional para onde está hoje. (POWELL, 2006, tradução nossa).

Date (2000) explica que, a introdução do modelo relacional em 1969-1970, por Codd, foi um evento marcante na história da pesquisa e desenvolvimento de bancos de dados. O modelo relacional de dados, a base da moderna tecnologia de banco de dados, evoluiu e cresceu ao longo dos anos se dedicando a analise de três aspectos principais: a manipulação de dados, a estrutura de dados e a integridade de dados.

As bases de dados relacionais preencheram a necessidade de ajudar as empresas a organizar e armazenar melhor seus dados e comparar transações de um país ou região para outra. Além disso, ajudaram gestores de empresas, que precisavam ser capazes de examinar informações para fins de tomada de decisão. (HURWITZ et al, 2013, tradução nossa).

Baseando-se fortemente na lógica e matemática, o modelo relacional satisfaz um princípio chamado *Princípio da Informação*, onde todo o conteúdo de informação do banco de dados é representado em linhas e colunas de tabelas. Em um sistema relacional, as tabelas são a estrutura lógica e não a física. No nível físico, o sistema pode armazenar os dados da maneira que preferir. (DATE, 2000).

Entidade, para Guimarães (2003), é o objeto do mundo real cujas particularidades ou características queremos armazenar. Podem ter existência abstrata ou física e podem ter

diversas propriedades específicas, denominadas atributos. Um atributo chave ou chave primaria, é um atributo projetado para identificar de forma única qualquer entidade.

Conceitualmente os relacionamentos e entidades são distintos, no entanto, no banco de dados, a diferença entre os eles deve ser estabelecida em seus atributos. A chave primária nos permite diferenciar as várias entidades em um conjunto de entidades. A chave estrangeira nos permite distinguir os vários relacionamentos em um conjunto de relacionamentos. (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

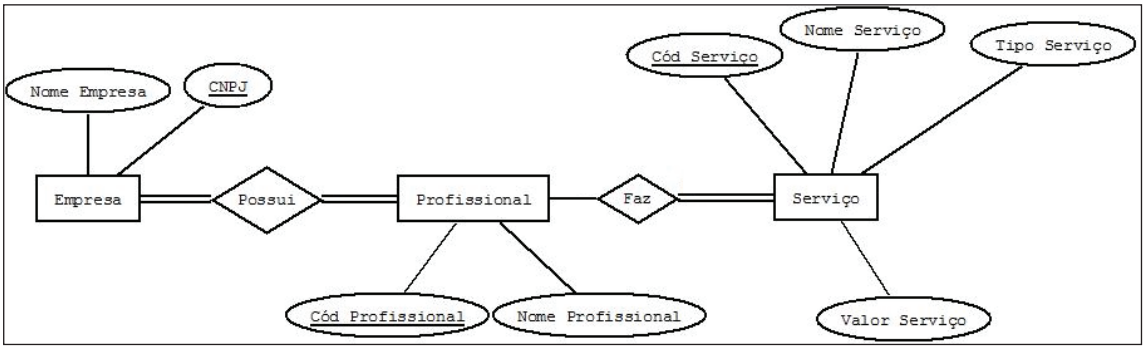
Relacionamentos, segundo Barbieri (1994), são os agrupamentos entre entidades de dados. Eles representam as ligações entre os blocos definidos de dados e estão ligados às ações realizadas sobre esses dados, representando os caminhos a serem percorridos no modelo de dados.

As tabelas podem ser ligadas entre si, independentemente da sua posição hierárquica. Para evitar divergências no modelo de dados, deve haver uma relação razoável entre as duas tabelas, mas as mesmas não são restringidas por uma rígida estrutura hierárquica. (POWELL, 2006, tradução nossa).

Ainda segundo Powell (2006, tradução nossa), modelo relacional aperfeiçoa as restrições de uma estrutura de dados hierárquica, não abandonando completamente a hierarquia dos dados. Tabelas podem ser acessadas diretamente, sem ter que acessar todos os objetos pai. O segredo é saber o que você deseja consultar, se você precisa encontrar os dados de um funcionário específico, você deve saber qual funcionário procurar, ou você pode analisar todos os funcionários. Dessa forma, você não precisa procurar em toda a hierarquia da empresa para encontrar os dados necessários de um único empregado.

Conforme Silberschatz, Korth e Sudarshan (1999), a estrutura lógica do banco de dados pode ser expressa graficamente pelo diagrama de Entidade-Relacionamento, onde são representados os conjuntos de entidades e seus atributos, relacionamentos e atributos chave. (Figura 8).

Figura 8 - Diagrama E-R correspondente à empresa, profissional e serviço.



Fonte: Elaboração do autor, 2013

Guimarães (2003) afirma que a modelagem relacional de dados se difundiu facilmente na comunidade de banco de dados não só pela sua simplicidade, mas também pelo encanto das linguagens de manipulação. Elas convergem ao principal ponto de observação de um usuário, a formulação de consultas complexas no banco de dados.

Em um modelo relacional utilizamos linguagens de consulta para obter informações do banco de dados. As linguagens de consulta podem ser divididas como procedurais e não procedurais. Em uma linguagem procedural, o usuário do banco de dados informa ao sistema como realizar determinada sequência de operações na base de dados para obter-se o resultado esperado. Nas linguagens não procedurais, o usuário apresenta a informação desejada sem fornecer um procedimento específico para obter as informações. (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

Guimarães (2003) define a álgebra relacional, como uma álgebra de expressões envolvendo relações, na qual a partir de uma ou mais relações da base de dados outras relações podem ser construídas, resultando na consulta sobre o banco de dados.

Silberschatz, Korth e Sudarshan (1999) afirmam que as operações principais da álgebra relacional são o *select*, *project*, *union*, *set difference*, *cartesian produc*t e *rename*.

Para Powell (2006, tradução nossa), a definição acadêmica de normalização é o formatar a base de dados no formato aceito das definições de Formas Normais. Muitos designers de banco de dados não entendem todas as facetas da normalização, portanto acabam não aplicando todas as suas regras da melhor maneira. O resultado da normalização é uma melhor organização e utilização do espaço físico da base de dados.

O estudo da normalização nos ajuda a projetar bases de dados com menos possibilidades de divergências, inconsistências e redundância de informações. A

normalização nos ajuda a determinar certos tipos de restrições sobre atributos de uma relação e as suas possíveis chaves. (GUIMARÃES, 2003).

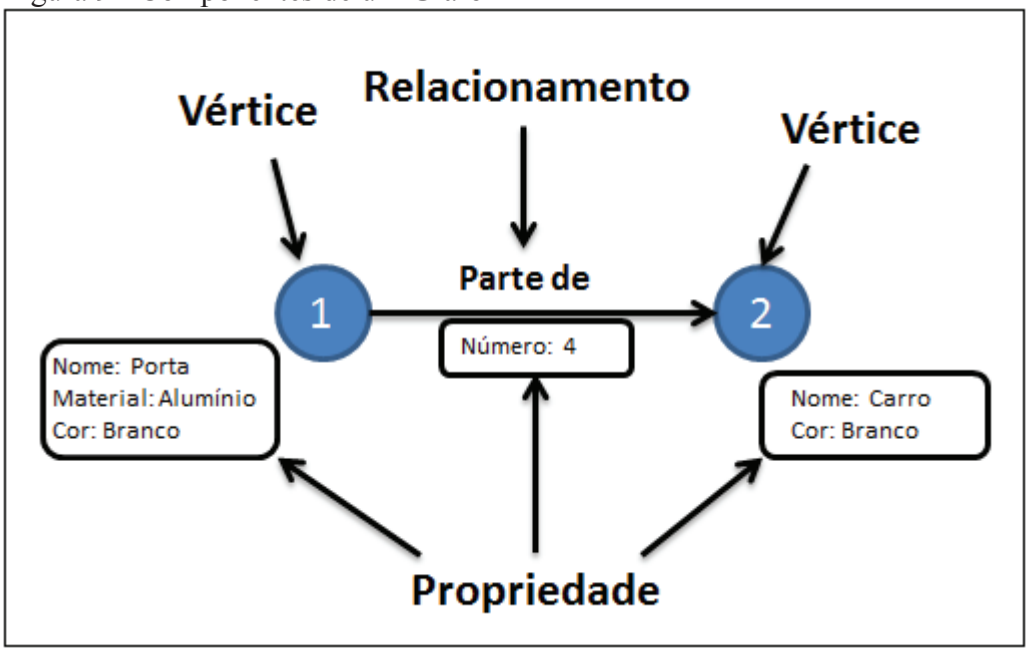
2.3.2 Modelagem Orientada a Grafos

Modelos de dados do orientados a grafo podem ser caracterizados como aqueles onde as estruturas de dados são modeladas na forma de grafos, e a manipulação de dados é expressa por operações orientadas a grafos. Estes modelos tiveram seu inicio no final dos anos oitenta e início dos anos noventa, em paralelo aos modelos orientados objeto. A necessidade de gerenciar informações com natureza inerente a grafos trouxe de volta a relevância da área, visto que toda uma nova onda de aplicações para bancos de dados orientados a grafos emergiu com o desenvolvimento de grandes redes como web, sistemas geográficos, transporte, telefones, redes sociais e redes biológicas. (ANGLES; GUTIERREZ, 2008, tradução nossa).

Em uma base de dados orientada a grafos, os dados são representados por nós, arestas e propriedades. Os nós representam as entidades ou objetos, as arestas demonstram a relação entre os nós e as propriedades que retratam as características específicas dos nós e arestas. (BATRA; TYAGI, 2012, tradução nossa).

A Figura 9 apresenta os componentes de um grafo e fornece uma representação visual de como eles se relacionam entre si. Nela é possível verificar que os vértices possuem propriedades e são organizados por relações que também possuem propriedades. (MILLER, 2013, tradução nossa).

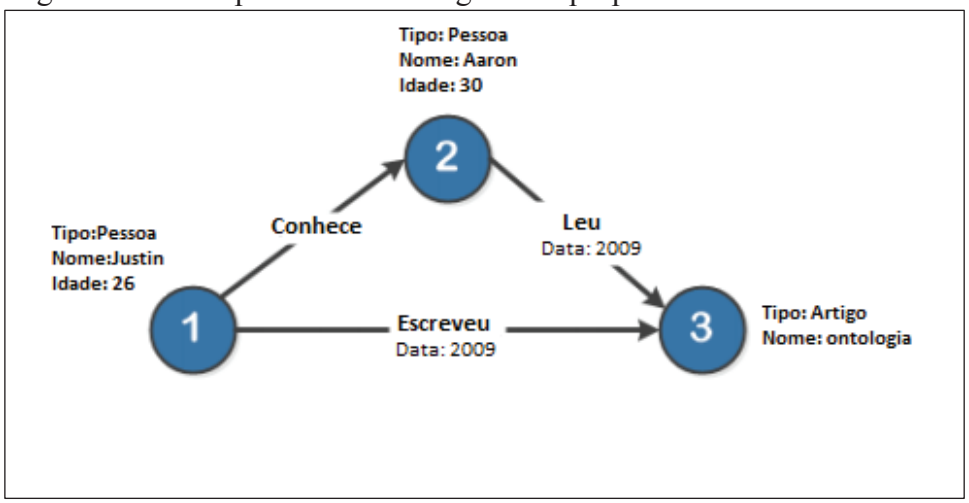
Figura 9 - Componentes de um Grafo



Fonte: O Autor (2013)

Segundo Miller (2013, tradução nossa), um tipo de grafo comum e apoiado pela maioria dos sistemas é o grafo de propriedade. Grafos de propriedade são multigrafos dirigidos e rotulados. A Figura 10 apresenta um exemplo visual de um grafo de propriedade que representa as interações entre pessoas e objetos. Nesse caso pode ser verificado que Justin escreveu um artigo no ano de 2009, que em 2010 foi lido por Aaron, conhecido de Justin.

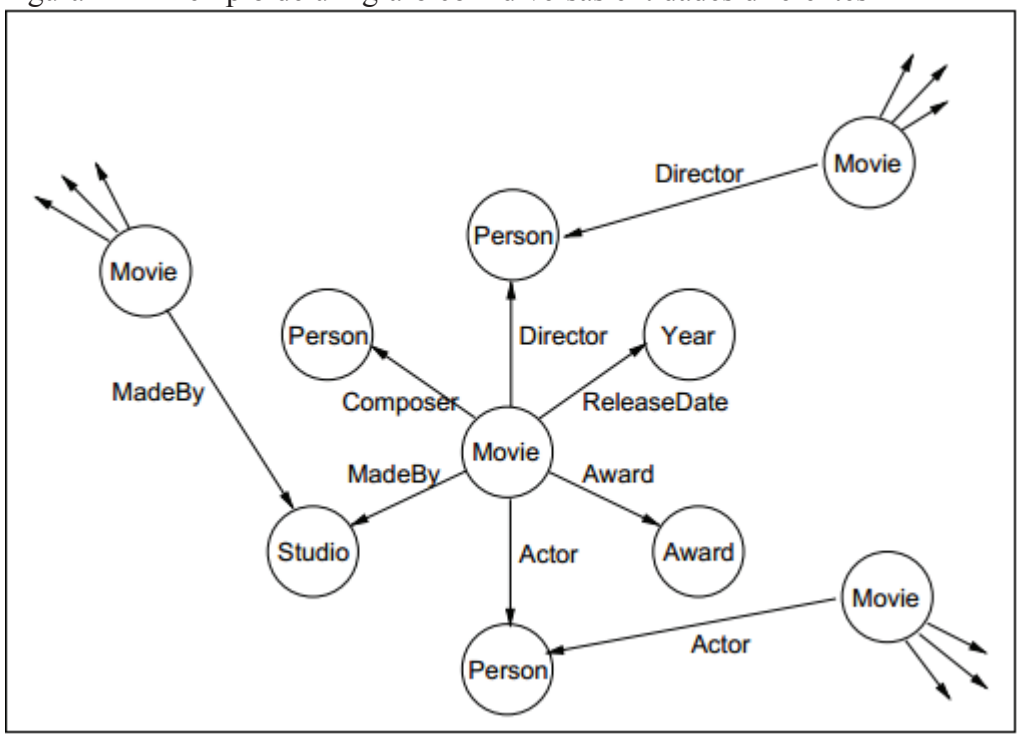
Figura 10 - Exemplo visual de um grafo de propriedade



Fonte: O Autor (2013)

Ao conectar pessoas, filmes, estúdios e outros objetos que têm relações entre si, um único grafo pode ser construído. Por exemplo, a Figura 11 demonstra como diferentes filmes podem ter diretores, atores e estúdios em comum. Da mesma forma, diferentes atores podem aparecer no mesmo filme, formando uma relação entre essas entidades. Analisando este grafo podemos responder a perguntas como: "Quais relações em comum podemos encontrar entre as entidades no banco de dados?". (COOK; HOLDER, 2007, tradução nossa).

Figura 11 - Exemplo de um grafo com diversas entidades diferentes



Fonte: Cook e Holder (2007, p.5).

Bancos de dados orientados a grafo garantem a evolução suave do seu modelo de dados. Novas entidades e novas composições se tornam novos nós e relacionamentos. Ao contrário do modelo relacional, não temos que optar entre uma estrutura de alta fidelidade ou o compromisso de alto desempenho. Com o modelo de dados orientado a grafos podemos manter a estrutura original de alta fidelidade do grafo, enquanto adicionamos elementos que atendem às novas necessidades. (ROBINSON; WEBBER; EIFREM, 2013, tradução nossa).

A ontologia do grafo define os tipos de vértices do mesmo e os tipos das arestas que ligam estes vértices. A ontologia também define os atributos para cada vértice. Diversas ontologias podem ser suportadas ao mesmo tempo e pode haver múltiplos grafos associados a cada ontologia. Dessa forma, podemos ter grafos com inúmeros tipos de entidades se relacionando entre si. (KAPLAN et al, 2008, tradução nossa).

A natureza livre de índice é a chave para percursos de alto desempenho, dessa forma, melhoram as consultas de alto desempenho e inserções na base de dados. Um aspecto importante do design de uma base de dados orientada a grafos é a forma em que os grafo são armazenados. (ROBINSON; WEBBER; EIFREM, 2013, tradução nossa).

A base de dados orientada a grafos processa com eficiência densos conjuntos de dados e o seu design permite a construção de modelos preditivos e análise de correlações e padrões de dados. Este modelo de dados, onde todos os nós estão ligados por relações, permite travessias rápidas ao longo das arestas entre os vértices. Bases de dados orientadas a grafo são muito eficientes quando se lida com áreas onde a interconectividade dos dados é importante. Muitas empresas têm desenvolvido soluções para lidar com a necessidade de se ter sistemas de banco de dados orientados a grafo. Como exemplo pode-se citar o Open Graph do Facebook, Grafo de Conhecimento do Google e FlockDB do Twitter. (MILLER, 2013, tradução nossa).

Para Aggarwal e Wang (tradução nossa, 2010), uma das características comuns de uma enorme gama de aplicações emergentes, incluindo redes sociais, gestão de ontologia, redes biológicas, redes de percursos e etc, é que os dados que essas aplicações armazenam são dados estruturados em grafos. Como o fluxo e a quantidade de dados aumentam de tamanho e complexidade, torna-se importante que eles sejam gerenciados por um sistema de banco de dados.

Hipergrafo é um modelo grafo generalizado, onde uma relação pode conectar qualquer número de nós. Mais especificamente, enquanto a propriedade do modelo orientado a grafos permite que a relação tenha um único nó de início e fim, o modelo hipergrafo permite um número qualquer de nós no início e fim de um relacionamento. (ROBINSON; WEBBER; EIFREM, 2013, tradução nossa).

No caso das redes de computadores e das redes da web, o número de nós no grafo pode ser enorme e isto pode levar a um grande número de arestas distintas. Nesses casos, o número de relacionamentos distintos pode ser tão grande, que manter esses dados no espaço de disco disponível se torne uma tarefa complicada. (AGGARWAL; WANG, 2010, tradução nossa).

Recuperar a informação a partir de um grafo requer algo que é conhecido como travessia, que envolve o conceito de caminhar ao longo dos vértices do grafo, uma operação necessária para a recuperação de dados. Uma diferença importante entre uma travessia e uma consulta SQL é que travessia é uma operação localizada, pois nesse caso não há índice de adjacência global. Cada vértice e aresta no grafo armazenam um índice dos objetos

conectados a ele, dessa maneira, o tamanho do grafo não ocasiona em perda de desempenho em uma travessia. Em um banco de dados orientado a grafos os índices globais existem, mas eles são usados somente quando se tenta encontrar o ponto de partida de uma travessia. (MILLER, 2013, tradução nossa).

Segundo Robinson, Webber e Eifrem (2013, tradução nossa), quando utilizamos um banco de dados orientado a grafo para a resolução de um problema do mundo real, com as limitações técnicas e de negócios do mundo real, as empresas escolhem o modelo orientado a grafos pelas seguintes razões: o alto desempenho independentemente do tamanho total do conjunto de dados (garantido pelas travessias no grafo), o modelo de grafos aproxima os domínios técnicos e de negócios facilitando a modelagem dos dados e a facilidade de se alterar o esquema de dados incluindo novas entidades e novos relacionamentos.

2.3.2.1 Modelagem Dimensional

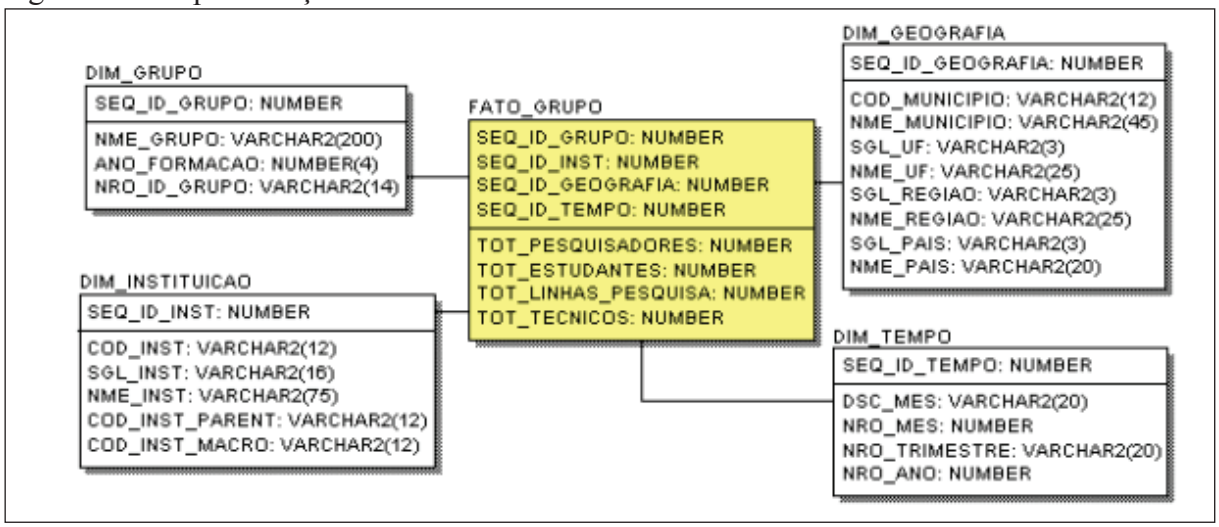
A informação é a matéria-prima para os Sistemas de Apoio à Decisão e a sua disponibilidade determina a eficácia em *Business Intelligence*. A implantação de um *Data Warehouse* envolve a união de diversas fontes de dados e a transformação desses dados em informações de qualidade, para permitir seu uso pelo usuário final no suporte à tomada de decisões. (CAMPOS; BORGES, 2002)

Para Ceci (2012), uma das fases iniciais do desenvolvimento de um DW é a identificação das necessidades do negócio. Por meio dessa fase que são identificadas as perguntas que se deseja responder com a análise dos dados e como o DW será constituído. Após definirmos e entendermos as necessidades e funcionamento do *Data Warehouse*, podemos iniciar o desenvolvimento da arquitetura do DW e a sua modelagem dimensional.

O modelo dimensional é composto basicamente por dois tipos de tabelas: as de fato e de dimensão. As tabelas de fato são grandes tabelas centrais, compostas basicamente das ocorrências do negócio, por exemplo, vendas, produção e defeitos. As tabelas de dimensão armazenam as descrições do negócio, como dados de um produto, tempo e dados de

clientes. Cada tabela de dimensão possui uma única chave primária, e o conjunto dessas chaves primárias formará a chave composta da tabela de Fato. (FORTULAN; FILHO, 2005)

Figura 12 – Representação de um Modelo Dimensional



Fonte: SELL (2006, p.30)

Na Figura 12 podemos visualizar um exemplo de um modelo dimensional, no qual são apresentadas as dimensões de instituições de ensino, de grupos de pesquisa, do local da instituição e data. Na tabela de fato fato_grupo estão representados o total de pesquisadores, total de estudantes, total de técnicos e quantidade de linhas de pesquisa desses grupos em um dado mês. (SELL, 2006)

Segundo Silva (2011), o modelo dimensional organiza os dados em uma estrutura padrão direcionada ao alto desempenho de consultas e dirigida à mineração de informações. Baseando-se na denormalização da estrutura de dados, o modelo dimensional não se atenta à redundância de dados, associando dados em dimensões e tabelas de fato, em vez de entidades e relacionamentos.

O modelo dimensional facilita o processamento das consultas e permite uma melhor visualização dos dados envolvidos, pois possui uma forma simples de organização e fornece uma grande flexibilidade em ajustes no modelo (KIMBALL; ROSS, 2002, tradução nossa).

2.3.2.2 Modelagem Orientada à Objeto

Na década de 90, houve um enorme interesse nos bancos de dados orientados a objeto. Os bancos de dados orientados a objetos foram concebidos com base nas linguagens de programação orientadas a objeto. (DATE, 2000)

Os bancos de dados orientados a objeto são a união de dois conceitos: orientação a objetos e bancos de dados. Iniciaram-se em projetos e pesquisas nas universidades e posteriormente se tornaram produtos comerciais. (KHOSHAFIAN, 1994).

Um dos princípios básicos do paradigma de banco de dados orientado a objetos é de que tudo é um objeto. Objetos podem ser mutáveis ou imutáveis, e cada objeto possui um tipo. Outra característica dos objetos é que eles são encapsulados, dessa forma, a representação interna dos objetos pode ser alterada sem a necessidade de ajuste nas aplicações que utilizam o mesmo. (DATE, 2000)

Todos os objetos possuem um identificador exclusivo e esses identificadores podem ser utilizados no banco de dados como ponteiros para fazer alusão ao objeto. Os códigos identificadores não são exibidos diretamente para o usuário, mas podem ser atribuídos a variáveis de instância dentro de outros objetos e variáveis de sistemas. (DATE, 2000).

Conforme Silberschatz, Korth e Sudarshan (1999), as interações entre os objetos e a aplicação são feitas através de mensagens. Nesse contexto o termo mensagens se refere à transmissão de pedidos entre os objetos, sem considerar particularidades da aplicação. Dessa forma, um objeto tem associado a ele um conjunto de mensagens, um conjunto de variáveis e um conjunto de métodos.

Normalmente, existem objetos que respondem as mesmas mensagens, usam os mesmos métodos e possuem variáveis do mesmo nome e tipo. Portanto para organizar a base de dados, esses objetos similares são agrupados formando uma classe. Cada objeto é chamado de uma instância da sua classe e todos possuem uma definição comum. (SILBERSCHATZ; KORTH; SUDARSHAN, 1999).

Um modelo de dados orientado a objeto proporciona uma estrutura onde qualquer item da base de dados pode ser recuperado a partir de qualquer ponto rapidamente. No entanto, ao recuperar mais do que um único item, o modelo relacional executa a tarefa de uma forma mais eficiente. O modelo orientado a objetos resolve alguns pontos complexos, como a

remoção da necessidade de tipos de dados e as tabelas de reposição dos relacionamentos muitos-para-muitos. (POWELL, 2006, tradução nossa).

Outra vantagem deste modelo é a sua capacidade para gerenciar e atender a aplicações complexas e modelos de banco de dados. Isto é devido ao princípio básico de objeto pelo qual os elementos complexos podem ser decompostos em estruturas básicas. (POWELL, 2006, tradução nossa).

3 MÉTODO

Método pode ser entendido como uma orientação para se atingir determinado objetivo. Portanto, possui um conjunto de etapas, ordenadamente dispostas, que devem ser concluídas para se chegar ao objetivo. (GALLIANO, 1979).

Para Gil (1999), os métodos definem os procedimentos lógicos que deverão ser realizados no processo de investigação científica dos fatos da natureza e da sociedade.

Neste capítulo, define-se o tipo de pesquisa do trabalho proposto e o porquê do mesmo ser classificado neste grupo, definimos a lista de etapas, que serão os próximos passos para a conclusão do trabalho. Apresenta-se, também, o esquema de solução, assim como as delimitações do trabalho, e por fim, mostra-se o cronograma criado com base na lista de etapas definidas.

3.1 CARACTERIZAÇÃO DO TIPO DE PESQUISA

Para Andrade (2001, p. 121), "pesquisa científica é um conjunto de procedimentos sistemáticos, baseados no raciocínio lógico, que tem por objetivo encontrar soluções para os problemas propostos mediante o emprego de métodos científicos".

Segundo Silva e Menezes (2005), a pesquisa pode ser classificada de diversas formas. Do ponto de vista da natureza, ela pode ser dividida em básica ou aplicada. Do ponto de vista da forma de abordagem do problema pode ser quantitativa ou qualitativa.

A pesquisa básica tem por objetivo buscar novos conhecimentos para o avanço da ciência sem a aplicação prática da mesma. No caso da pesquisa aplicada, o objetivo é a obtenção de novos conhecimentos para fins práticos com o objetivo de solucionar problemas específicos. (SILVA; MENEZES, 2005).

Com base nos objetivos propostos por este trabalho, pode-se definir que este representa uma pesquisa aplicada. Pois tem o objetivo de gerar conhecimentos para a aplicação prática dirigida a solução de um problema. Do ponto de vista dos procedimentos técnicos, este trabalho utiliza a pesquisa bibliográfica, pesquisa experimental e um estudo de caso.

De acordo com Gil (1999), os objetivos de uma pesquisa podem ser exploratórios, descritivos ou explicativos. Esse projeto é classificado como pesquisa exploratória, pois envolve pesquisa bibliográfica e uma análise comparativa do uso de um banco de dados relacional e um banco de dados orientado a grafos tendo como estudo de caso a análise social.

"As pesquisas exploratórias têm como principal finalidade desenvolver, esclarecer e modificar conceitos e ideias, tendo em vista a formulação de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores." (GIL, 1999, p.27).

Em relação à forma de abordagem, este projeto pode ser classificado como quantitativo. Considerado quantitativo devido à utilização de recursos e técnicas para análise de resultados comparativos.

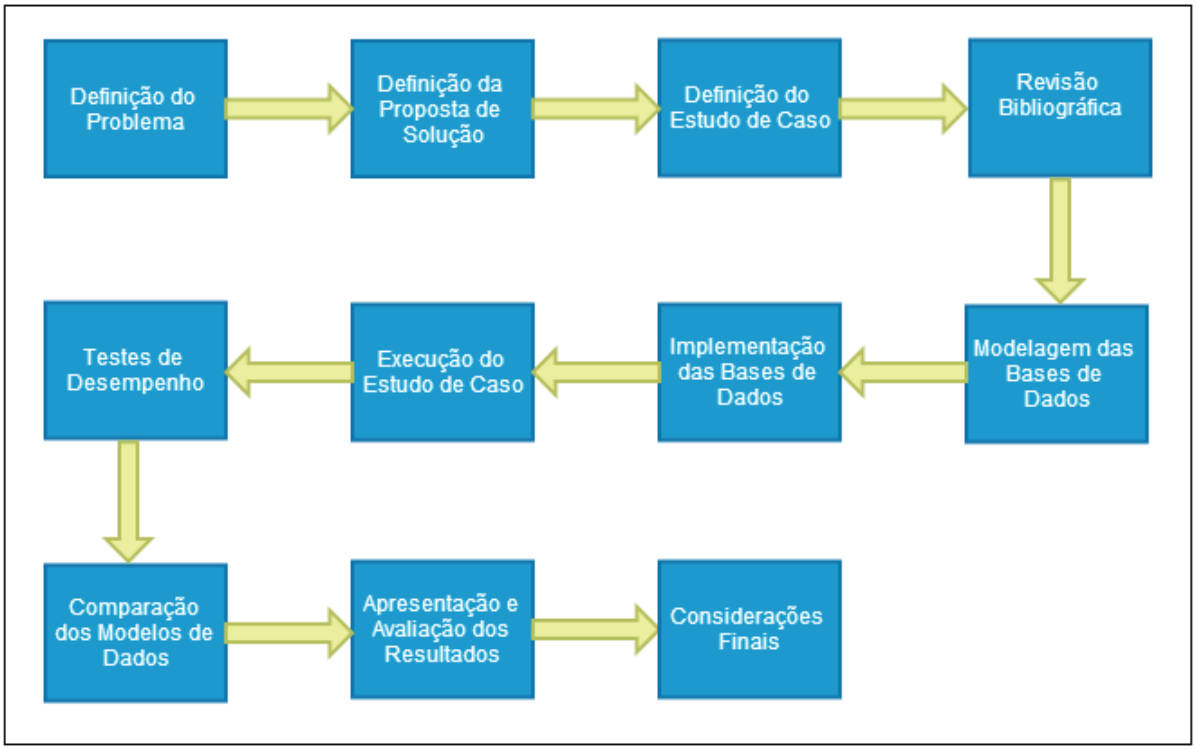
3.2 ETAPAS

O processo de pesquisa e elaboração deste trabalho organiza-se de acordo com as seguintes etapas:

1) Definição do Problema;
2) Definição da Proposta de Solução;
3) Definição do Estudo de Caso;
4) Revisão Bibliográfica;
5) Modelagem das Bases de Dados;
6) Implantação das Bases de Dados;
7) Execução do Estudo de Caso;
8) Testes de Desempenho;
9) Comparação dos Modelos de Dados;
10) Apresentação e Avaliação dos Resultados;
11) Considerações Finais.

O Fluxograma, ilustrado na Figura 13, ilustra as principais etapas deste projeto.

Figura 13 - Fluxograma das Etapas do Projeto



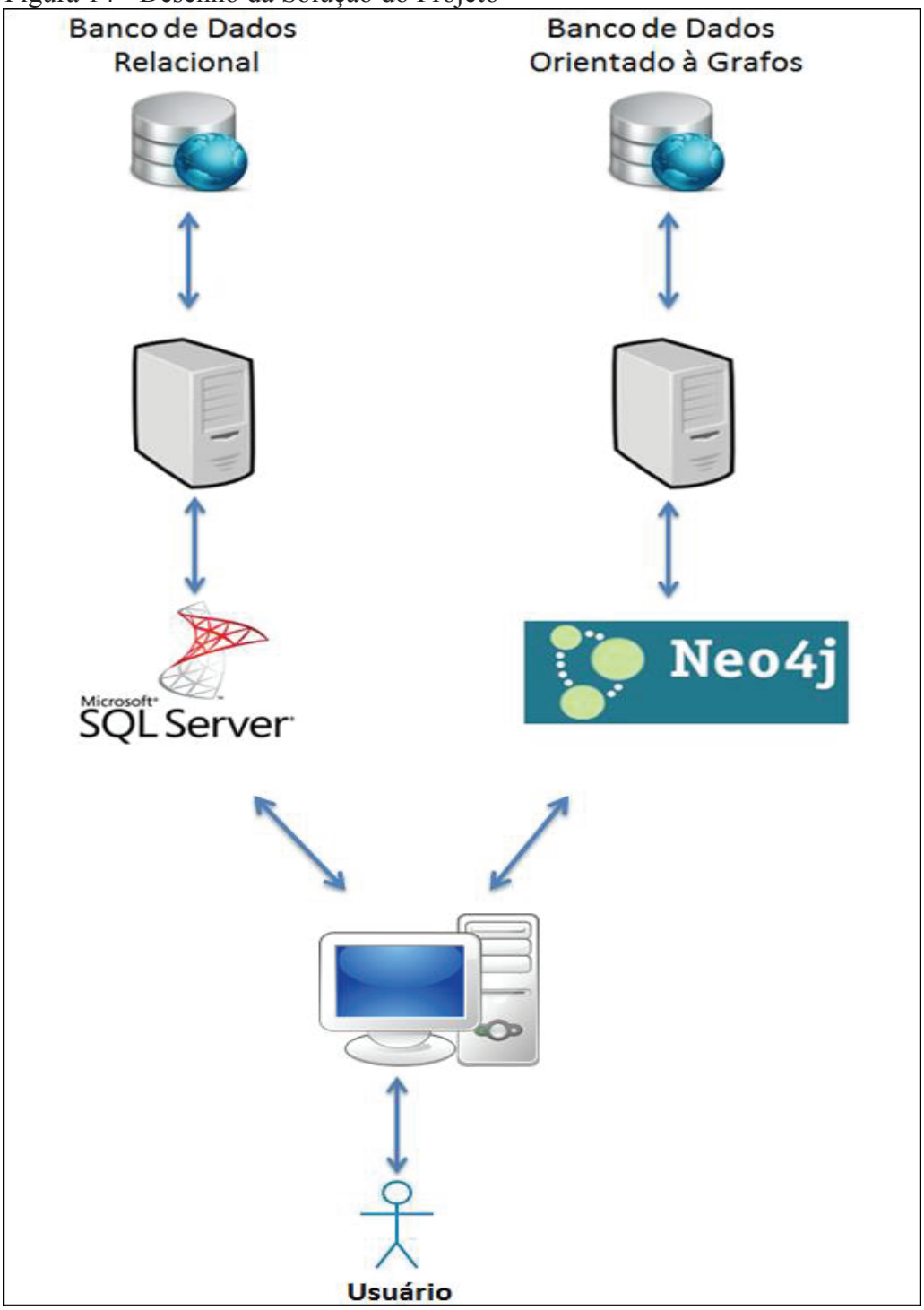
Fonte: O Autor (2013)

As etapas demonstradas na Figura 13 delimitam e organizam o processo de pesquisa e desenvolvimento do trabalho.

3.3 DESENHO DA SOLUÇÃO

Considerando os objetivos pretendidos, o estudo de caso e as tecnologias escolhidas, foi desenvolvida uma proposta de solução para o problema, como pode ser visualizado na Figura 14.

Figura 14 - Desenho da Solução do Projeto



Fonte: O Autor (2013)

A solução desenvolvida utiliza os dados retirados de uma rede social como fonte de informações para o desenvolvimento dos bancos de dados. Essas informações vinculadas a pessoas e suas relações de amizade são modeladas conforme os paradigmas de banco de dados escolhidos, pois são à base do estudo de caso em análise social.

São implementadas duas bases de dados com as mesmas informações armazenadas, uma utilizando o modelo relacional e a outra utilizando o modelo orientado a grafos.

As bases de dados ficarão armazenadas em um mesmo servidor para que se tenham as mesmas condições nas consultas e testes necessários.

O banco de dados relacional utiliza o SQL Server 2008 como Sistema Gerenciador de Banco de Dados e o banco de dados orientado a grafo utilizará o Neo4J.

Após a conclusão do desenvolvimento das bases de dados, são aplicadas consultas complexas envolvendo relacionamentos entre os dados armazenados para que os resultados obtidos sejam analisados e comparados.

3.4 DELIMITAÇÕES

Não foi desenvolvida nenhuma aplicação para registro de informações nas bases de dados. Os registros são inseridos através de importação de dados de arquivos JSON realizada por meio de uma aplicação desenvolvida utilizando a linguagem de programação Java e biblioteca Google GSON.

O SGBD utilizado para o banco de dados relacional é o SQL Server 2008 Enterprise, pois se trata de uma aplicação robusta, segura, o registro é disponibilizado pela UNISUL e se trata do banco de dados no qual o autor deste trabalho possui maior experiência.

Para a modelagem e desenvolvimento do banco de dados orientado a grafos utiliza-se o Neo4J, porque é um banco de dados que possui uma versão de fácil instalação, configuração, manipulação de dados e possui licença livre para sistemas que não estão em produção.

4 MODELAGEM

Nesta seção são apresentadas as definições de técnica e metodologia, alguns conceitos básicos como UML, levantamento de requisitos, casos de uso, modelo de domínio, modelo E.R e modelo em grafo, o estudo de caso para validar a proposta de solução e o esquema físico e lógico do projeto de solução.

4.1 UML

Segundo Booch, Rumbaugh e Jacobson (2000, p. XIII), "a UML, Linguagem Unificada de Modelagem, é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software".

A criação da UML iniciou oficialmente em outubro de 1994 quando Rumbaugh juntou-se a Booch em um esforço para a unificação dos métodos Booch e OMT (Object Modeling Technique) de modelagem. Em 1995, Jacobson se juntou a eles e em 1997 eles lançaram a versão 1.0 da linguagem UML que foi aprovada pelo OMG (Object Management Group). Desde então, a UML tem tido grande aceitação pela comunidade de desenvolvedores de sistemas. (BOOCH; RUMBAUGH; JACONSON, 2000).

De acordo com Furlan (1998), a modelagem UML vai além de uma simples padronização unificada, pois agregou conceitos das técnicas de modelagem de dados, modelagem de objetos e componentes, modelagem de negócios e integrou as melhores práticas de desenvolvimento do mercado.

A UML independe de linguagens de programação e processos de desenvolvimento, dessa forma, pode ser utilizada na modelagem de sistemas desenvolvidos em qualquer linguagem de programação. (BEZERRA, 2002)

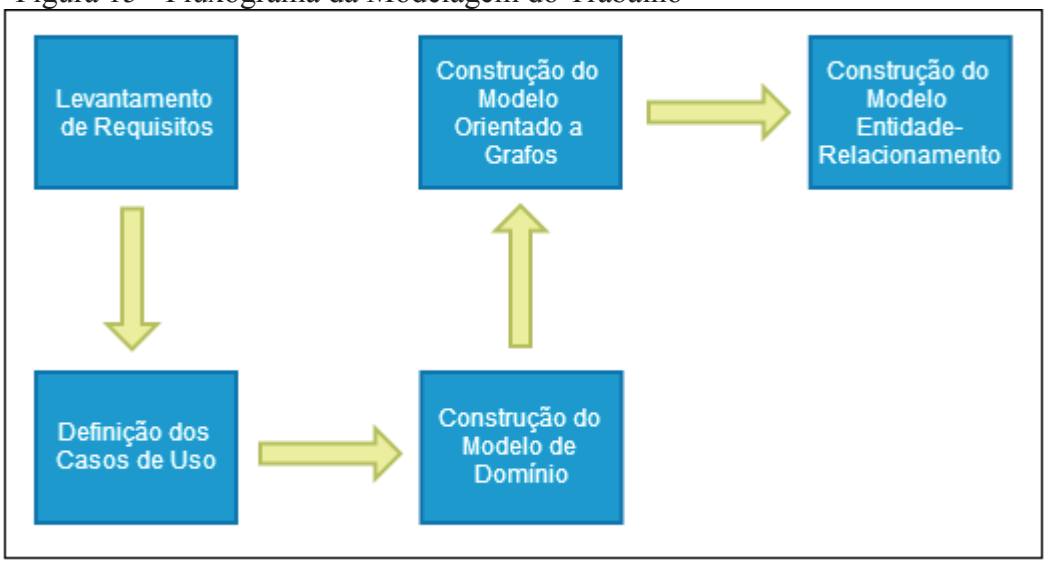
Booch, Rumbaugh e Jacobson (2000) afirmam que os diagramas UML mais utilizados são: diagrama de classe, diagrama de casos de uso, diagrama de estados, diagrama de sequência, diagrama de implantação, diagrama de atividade e diagrama de componentes.

- Diagrama de classe apresenta conjuntos de classes, colaborações e interfaces, bem como seus relacionamentos. (BOOCH; RUMBAUGH; JACONSON, 2000).
- Diagrama de casos de uso descreve as funcionalidades do sistema percebidas por atores externos. Os atores, que podem ser usuários, outro sistema ou dispositivos, interagem com o sistema e o diagrama demonstra o relacionamento atores e os casos de uso. (FURLAN, 1998).
- Diagrama de estados representa a análise das transições entre os estados dos objetos de um software. Ele nos permite descrever o ciclo de vida de objetos, eventos que causam a transição de um estado para o outro e a realização de operações. (BEZERRA, 2002).
- Diagrama de sequência apresenta a relação de sequência de tempo dos objetos. Ele é um diagrama de interação no qual o destaque está na ordenação temporal das mensagens, mostrando a colaboração dinâmica entre os objetos. . (FURLAN, 1998).
- Diagrama de implantação demonstra a arquitetura dos pontos de processamento em tempo de execução e os componentes neles existentes. Apresenta também o uso físico do software considerando dispositivos e suas interconexões. (BOOCH, RUMBAUGH, JACONSON, 2000).
- Diagrama de atividades exibe o fluxo de uma atividade para outra. Seu objetivo é estudar o fluxo dirigido do sistema, representando as atividades desempenhadas em uma operação. (FURLAN, 1998).
- Diagrama de componentes descreve os componentes de um sistema e suas dependências. Envolve a visão estática da implantação de um sistema. (BEZERRA, 2002).

4.2 MÉTODO DA MODELAGEM DO DESENVOLVIMENTO

Nesta seção são abordados conceitos relacionados a levantamento de requisitos, casos de uso, modelo de domínio, modelo E.R e modelo orientado a grafos. Além dos conceitos básicos, também é apresentada a modelagem do sistema desenvolvido para realizar a persistência dos dados obtidos através da API do Facebook com as bases de dados, relacional e orientada a grafos, utilizadas neste trabalho.

Figura 15 - Fluxograma da Modelagem do Trabalho



Fonte: O Autor (2013)

A Figura 15 ilustra um fluxograma com o processo de modelagem de dados utilizado para a construção das bases de dados utilizadas neste trabalho, que é apresentada nessa seção.

4.2.1 Levantamento de Requisitos

Para Engholm (2010, p.69) “os requisitos definem as expectativas e necessidades dos envolvidos no projeto, podendo ser divididos em requisitos funcionais e não funcionais”.

Os requisitos funcionais definem as funções que o sistema deve oferecer, como deve agir diante de entradas específicas e como deve se comportar em determinadas situações.(SOMMERVILLE, 2003).

Sommerville (2003) define como requisitos não funcionais, as restrições sobre os serviços ou funcionalidades oferecidas pelo sistema. Nesse contexto, podemos destacar as restrições sobre o processo e de tempo, padrões, desempenho, usabilidade, confiabilidade, escalabilidade e extensibilidade.

O levantamento de requisitos é extremamente importante para o sucesso do sistema, é nessa fase inicial de qualquer projeto que devemos definir com clareza os objetivos e restrições do mesmo para entregar ao cliente um sistema que atenda suas necessidades e expectativas. (ENGHOLM, 2010).

No Quadro 1 são ilustrados os requisitos funcionais do sistema desenvolvido para realizar a inserção dos dados obtidos através da API do Facebook e as bases de dados utilizadas neste trabalho.

Quadro 1 - Requisitos Funcionais

Identificador	Descrição
RF01	O sistema deve permitir o cadastro de pessoas
RF02	O sistema deve armazenar os dados relacionados aos indivíduos cadastrados
RF03	O sistema deve permitir o cadastro de relacionamento entre as pessoas cadastradas
RF04	Cada pessoa deverá ser cadastrada a um único código identificador
RF05	O sistema deve permitir a carga de arquivos JSON com dados serem importados para banco de dados

Fonte: Elaboração do autor, 2013.

No Quadro 2, são apresentados os requisitos não funcionais do sistema desenvolvido:

Quadro 2 - Requisitos Não Funcionais

Identificador	Descrição
RNF01	O sistema deve permitir a integração com banco de dados SQL Server 2008
RNF02	O sistema deve permitir a integração com banco de dados Neo4J
RNF03	O sistema deve manter sigilo sobre os dados relacionados a usuários do Facebook, obtidos através da API do Facebook.
RNF04	O sistema deve ser fácil de utilizar por usuários experientes e deve ser organizado de modo que os erros dos usuários sejam minimizados

Fonte: Elaboração do autor, 2013.

4.2.2 Casos de Uso

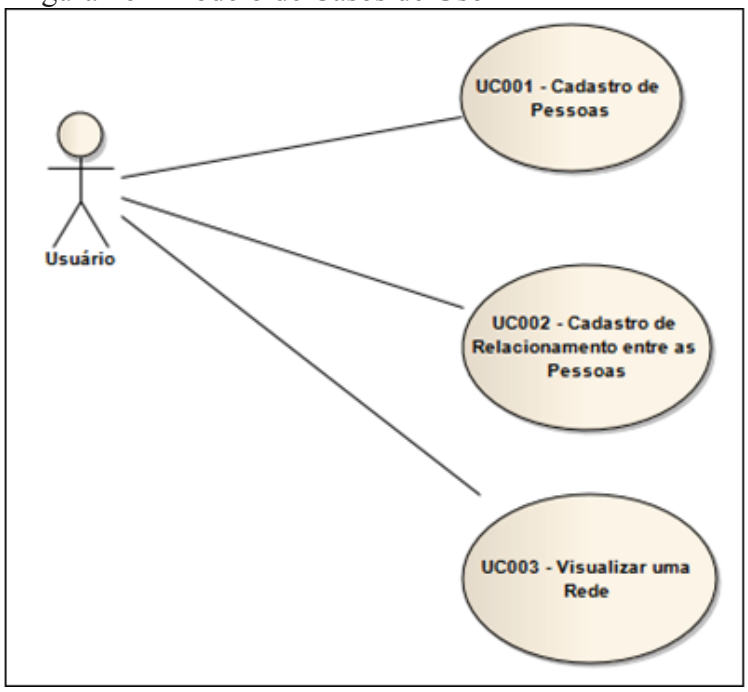
"Um caso de uso é um documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo." (LARMAN, 2000).

Ivar Jacobson, engenheiro de software sueco, idealizou técnica de modelagem através de casos de uso na década de 1970, enquanto trabalhava no desenvolvimento de um sistema de uma empresa de telefonia. Anos depois, Jacobson uniu-se a Booch e Rumbaugh, e a modelagem através de casos de uso foi incorporada a UML. Desde então, esse modelo tem se tornado cada vez mais conhecido devido a sua notação simples, que facilita na comunicação e desenvolvimento de sistemas. (BEZERRA, 2002).

Segundo Engholm (2010), os casos de uso podem ser considerados como comportamentos e funcionalidades desejadas para o sistema visíveis aos usuários. A notação simplificada dos diagramas de casos de uso tem a finalidade de fornecer uma maneira intuitiva para se entender as funcionalidades de um sistema, provendo uma visualização sucinta do comportamento do mesmo.

Casos de uso bem estruturados apresentam somente os comportamentos essenciais de um sistema. Eles descrevem um conjunto de sequências representando a interação de atores com o sistema. (BOOCH, RUMBAUGH, JACONSON, 2000).

Figura 16 - Modelo de Casos de Uso



Fonte: Elaboração do autor, 2013.

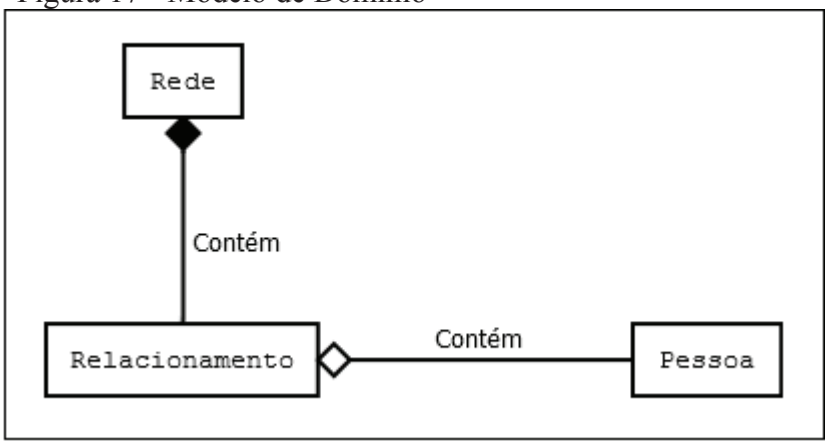
Na Figura 16, podem-se visualizar os casos de uso definidos para o sistema de persistência de dados desenvolvido para este trabalho.

O sistema de persistência de dados será responsável por realizar a carga dos dados obtidos pelo sistema de captura de dados nas bases de dados utilizadas neste trabalho.

4.2.3 Modelo de Domínio

A Figura 17 ilustra o modelo de domínio do estudo de caso. Nela podemos identificar a entidade Pessoa, que pode conter ou não relacionamentos. Os relacionamentos entre as entidades Pessoa formam a rede estudada nesse projeto.

Figura 17 - Modelo de Domínio



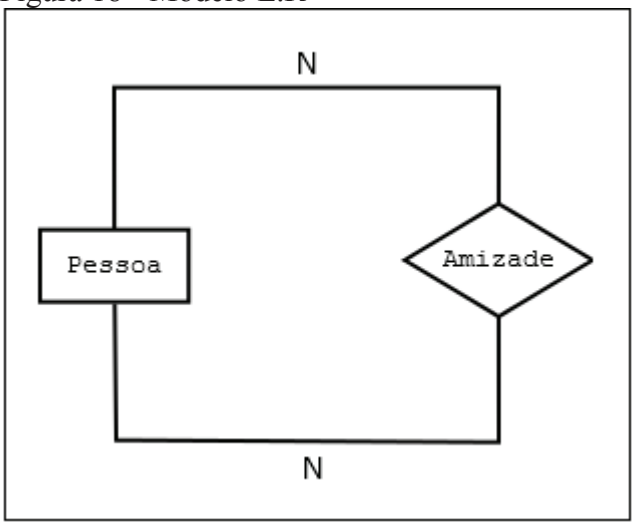
Fonte: Elaboração do autor, 2013.

O modelo de domínio apresenta a compreensão e informação adquirida acerca do domínio. Trata-se de uma perspectiva conceitual de objetos em uma situação real do mundo que auxilia na compreensão dos termos e conceitos, bem como seus relacionamentos.

4.2.4 Modelo E.R

A Figura 18 ilustra a modelagem E.R, entidade-relacionamento, do banco de dados relacional utilizado neste trabalho. A modelagem foi feita através dos recursos da ferramenta DIA Diagram Editor.

Figura 18 - Modelo E.R



Fonte: Elaboração do autor, 2013.

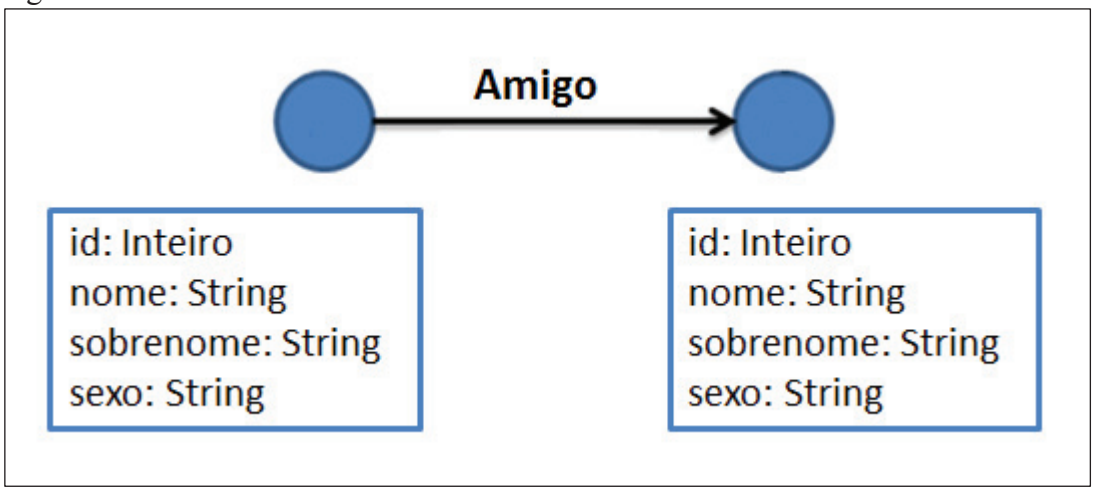
Pode-se verificar que se trata de um Auto Relacionamento, no qual a entidade Pessoa se relaciona com ela mesma através do relacionamento Amizade. Dessa forma, o relacionamento Amizade irá estabelecer uma conexão entre os indivíduos da rede social.

4.2.5 Modelo em Grafo

A Figura 19 ilustra a modelagem da base de dados orientada a grafos utilizada neste trabalho. Nela pode-se identificar as entidades pessoa representada pelos vértices do

grafo, ligadas através do relacionamento “Amigo”. Os atributos das entidades pessoa são relacionados como propriedades das mesmas.

Figura 19 - Modelo Orientado a Grafos



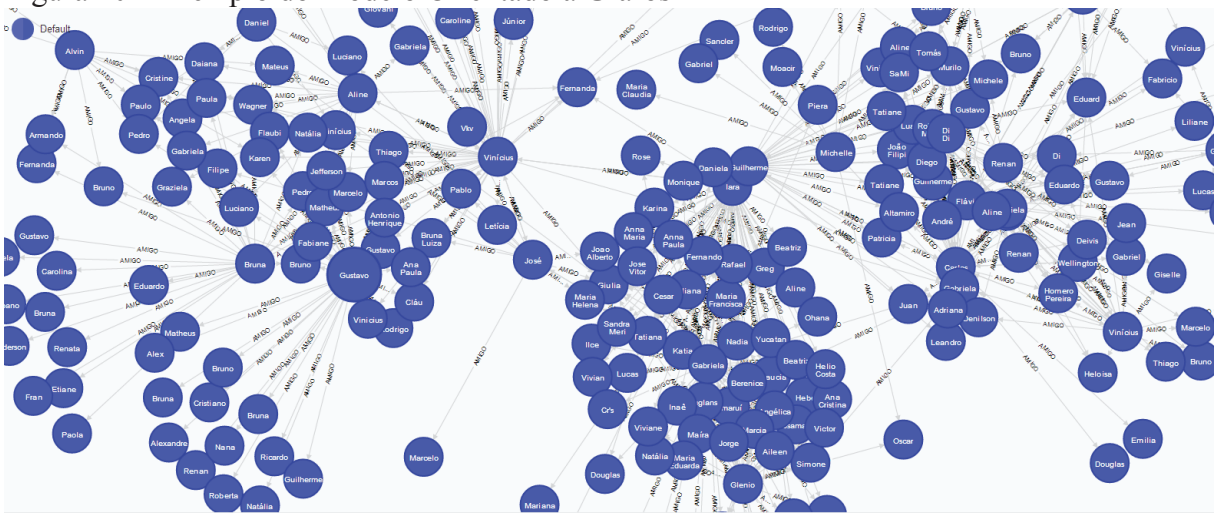
Fonte: Elaboração do autor, 2013.

Para Foggia, Sansome e Vento (2001, tradução nossa) os grafos são estruturas de dados dotadas de um poder tão expressivo que seu uso torna-se rentável nas mais diversas áreas.

A ontologia do grafo define as propriedades do mesmo e os seus atributos. Diversas ontologias podem ser utilizadas ao mesmo tempo e diversos grafos podem estar associados a cada ontologia. Dessa forma, podemos ter grafos com inúmeros tipos de entidades se relacionando entre si.

A Figura 20 ilustra um exemplo expandido da modelagem da base de dados orientada a grafos utilizada neste trabalho. Através da Figura 20 pode-se identificar os clusters formados pelas entidades agrupadas através de sua relação de amizade.

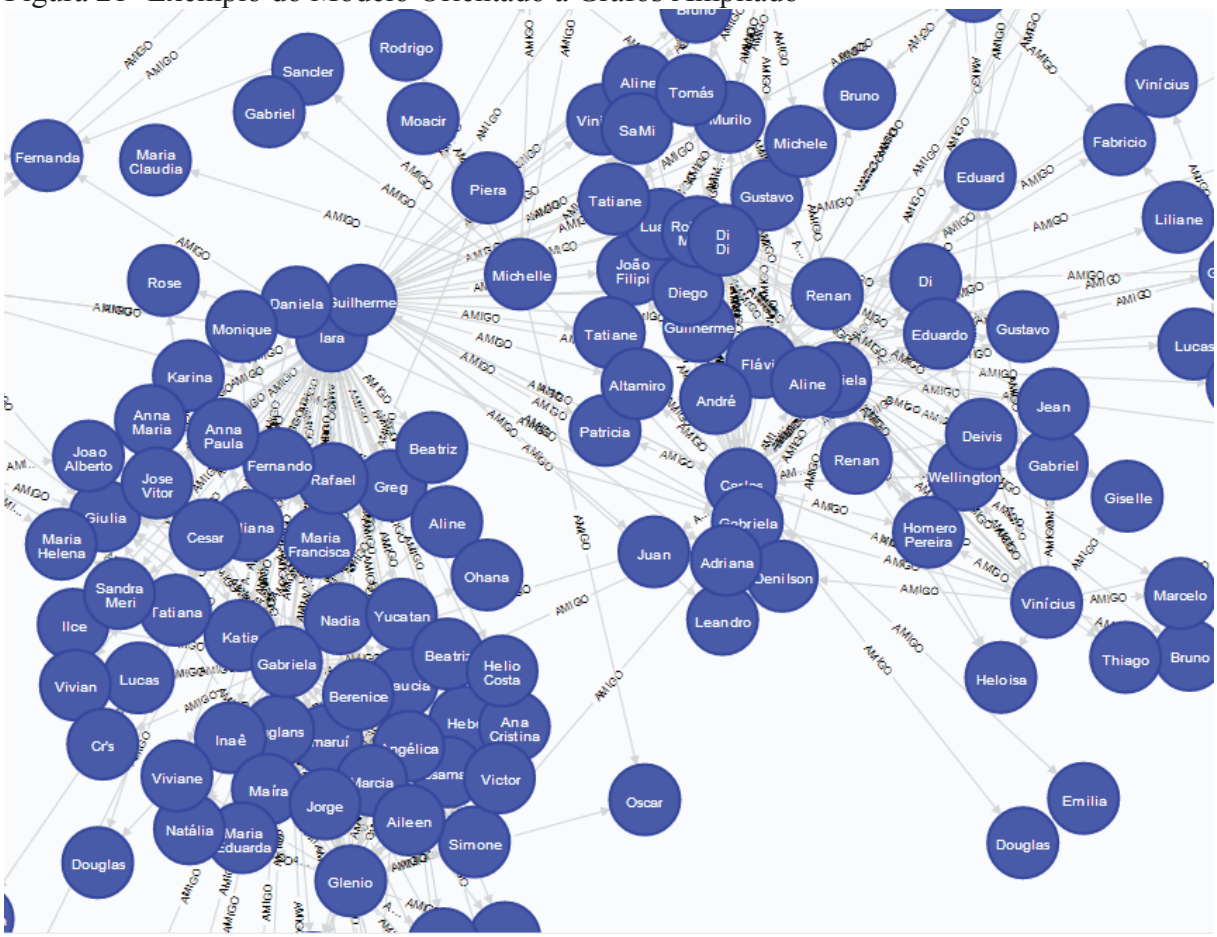
Figura 20 - Exemplo do Modelo Orientado a Grafos



Fonte: Elaboração do autor, 2013.

A Figura 21 apresenta uma visão ampliada da figura anterior. Nela pode-se identificar a propriedade nome de cada nó do grafo e o relacionamento amigo que liga os mesmos.

Figura 21- Exemplo do Modelo Orientado a Grafos Ampliado



Fonte: Elaboração do autor, 2013.

Um grafo de propriedade é composto de nós, relacionamentos e propriedades. Esta estrutura simples é tudo o que se necessita para criar modelos sofisticados e semanticamente ricos. O modelo de dados orientado a grafo é uma ferramenta eficaz para a modelagem de dados quando um foco no relacionamento entre as entidades é o ponto principal da concepção do modelo de dados

São muitos os benefícios de se utilizar um modelo de dados orientado a grafo, dentre eles pode-se citar a introdução de um nível de abstração que permite uma modelagem mais natural de dados do grafo, linguagens de consulta e métodos desenvolvidos para consultar diretamente a estrutura do grafo e algoritmos e técnicas da teoria de grafos para armazenar e consultar grafos.

5 DESENVOLVIMENTO

Nesta seção do trabalho, é apresentado o histórico do desenvolvimento do estudo de caso, as tecnologias utilizadas no desenvolvimento da solução proposta e a infraestrutura criada para conceber o estudo de caso e os resultados dos testes comparativos entre a abordagem relacional e orientada a grafos. Para demonstrar as diversas possibilidades da utilização de bases orientadas a grafo para redes sociais, também serão apresentados casos de uso de análise social.

5.1 HISTÓRICO DO DESENVOLVIMENTO

Para o desenvolvimento do estudo de caso, inicialmente, foi necessário realizar um levantamento das possíveis redes sociais de onde os dados seriam extraídos. Primeiramente, o Facebook, Mendelay e o Twitter foram às redes identificadas como

possíveis alvos para o desenvolvimento da pesquisa. Diante desse contexto, algumas características necessárias foram estabelecidas para definir e validar a rede social escolhida:

- Forma em que as relações estabelecidas entre os usuários da rede;
- Disponibilidade de API para desenvolvimento de uma aplicação;
- Facilidade na extração de dados;
- Dados fornecidos pela rede social;
- Forma em que os dados da rede social são fornecidos;
- Quantidade de usuários cadastrados na rede social;
- Visibilidade da rede social na internet.

Após a análise das redes sociais levando em consideração os requisitos citados, foi definido que a rede social mais indicada nesse projeto seria o Facebook. Como essa rede social atende os requisitos necessários e está extremamente difundida no Brasil e no mundo, haveria uma enorme quantidade de indivíduos que poderiam ser analisados. As relações de amizade, parentesco, trabalho, estudo, interesses e outras, poderiam ser analisadas, enriquecendo a rede montada nesse projeto e dando abertura a trabalhos futuros.

Para atender os requisitos e objetivos estabelecidos neste projeto, os dados selecionados para montar o grafo social foram o nome, sobrenome, sexo e código dos usuários na base de dados do Facebook, além da relação de amizade estabelecida na rede social que os conecta, estabelecendo-se como as arestas do grafo social.

O Facebook disponibiliza em sua área de desenvolvimento a Graph API, utilizada para estabelecer conexão com aplicações, aceso e segurança de dados, publicação e pesquisa de dados e apresentação de resultados. Através dessa API os dados das consultas realizadas na base de dados do Facebook são retornados em forma de arquivo JSON, dessa forma podem ser explorados e integrados a diversos sistemas.

Durante os estudos da Graph API, verificou-se que a rede social impõe algumas regras e necessidade de algumas autorizações como medida de segurança e de garantir a privacidade de seus usuários. Portanto, para que os dados da lista de amigos fossem disponibilizados pela base de dados da rede social, seria necessária a autorização do individuo que quisesse colaborar com a pesquisa.

Realizar a configuração de tokens para autorização de forma manual na ferramenta Graph API Explorer, fornecida pelo Facebook, seria um impeditivo para os usuários participarem deste projeto e fornecerem seus dados. Pois muitos usuários não

possuem acesso e conhecimento as ferramentas de desenvolvimento disponibilizadas pela rede social. Dessa forma, o usuário teria que acessar a ferramenta Graph API Explorer, realizar as configurações de tokens para liberação de acesso aos dados da lista de amigos, executar a consulta de dados, salvar os dados obtidos em um arquivo de texto e enviar o arquivo para o autor deste trabalho.

Com o intuito de facilitar o processo de autorização de acesso aos dados dos usuários do Facebook, foi desenvolvida uma página em PHP utilizando a SKD do Facebook para esta linguagem de programação, que conecta na base de dados da rede social e envia uma solicitação de autorização ao usuário automaticamente. Após obter a permissão, o mesmo envia uma solicitação com a consulta de dados ao Facebook que retorna o resultado em forma de dados JSON. Os arquivos gerados com as informações dos usuários são enviados automaticamente para o e-mail do autor deste trabalho.

No principio houveram algumas dificuldades em relação à configuração de algumas funções do PHP necessárias para o desenvolvimento dessa aplicação, pois por motivos de segurança, os serviços de hospedagem de web sites deixam esses métodos desabilitados e algumas portas de conexão bloqueadas. Após solicitar algumas liberações ao serviço de hospedagem, a conexão com o Facebook foi estabelecida e os dados foram retornados com sucesso.

Como os bancos de dados utilizados neste trabalho não possibilitam a leitura, identificação e persistência direta dos dados através de um arquivo JSON, houve a necessidade de implementar uma aplicação para realizar essa tarefa. Foi desenvolvido um sistema em JAVA que faz a leitura dos arquivos JSON disponibilizados pelo Facebook e insere os dados dos usuários e seus relacionamentos nas bases de dados Neo4J e SQL Server 2008. O sistema teve por objetivo automatizar a leitura e inserção de dados nas bases a serem utilizadas nos testes, evitando que a análise e replicação desses dados fossem realizadas manualmente.

Durante o desenvolvimento do sistema para persistir os dados dos usuários, verificou-se que seria mais eficiente utilizar uma biblioteca para converter arquivos JSON em objetos JAVA, do que criar os tratamentos necessários a partir do zero. Durante algumas semanas, algumas bibliotecas JAVA de conversão de arquivos JSON foram estudadas para que a mais eficiente e simples, fosse escolhida e utilizada nesse processo. Após levantamento e análise das bibliotecas disponíveis, a escolhida foi a GSON desenvolvida pela Google. A utilização dessa biblioteca simplificou e facilitou o desenvolvimento desta aplicação, além de trazer segurança a persistência de dados obtidos do Facebook.

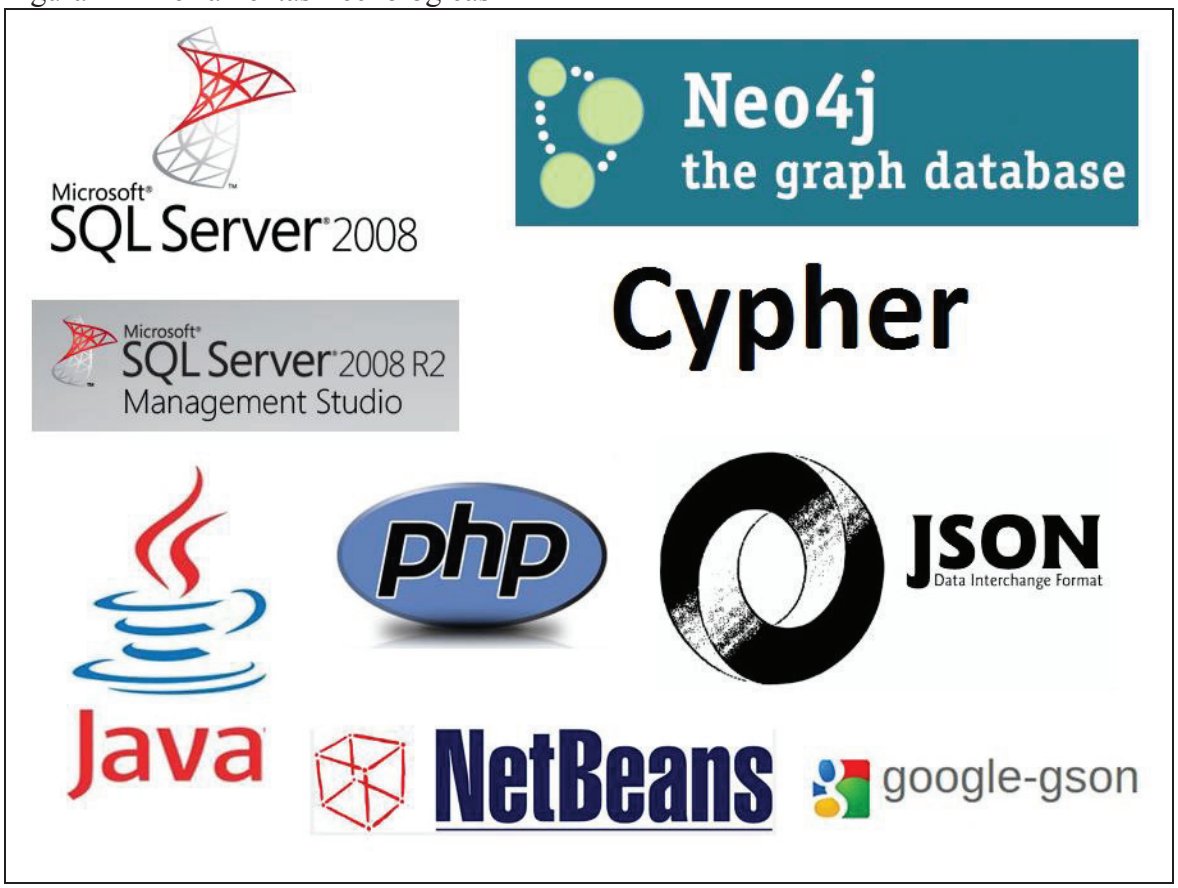
Após os primeiros testes de integração com o banco orientado a grafos Neo4J, constatou-se que devido aos dados serem estruturados de forma diferente, algumas alterações deveriam ser feitas na aplicação de persistência de dados, principalmente no tocante a registro de índices para os novos nós. Devido à diferença na estrutura de dados, a Graphdb Index API teve que ser estudada com muito cuidado, pois dados poderiam ser duplicados, persistidos de maneira incorreta ou então ignorados e não persistidos na base de dados. Após testes e estudos relacionados à criação de índices no Neo4J, verificou-se a necessidade de criar índices para os novos nós criados e para as suas propriedades. Dessa forma, a duplicidade de dados não aconteceria e o desenvolvimento das consultas de dados no Neo4J seria simplificado devido à indexação das propriedades. Como este trabalho utiliza somente um tipo de relacionamento em seu estudo de caso, a criação de um índice para os relacionamentos não se fez necessária.

5.2 FERRAMENTAS TECNOLÓGICAS

A seguir, serão descritas as tecnologias e ferramentas utilizadas no desenvolvimento do estudo de caso desse projeto.

A Figura 22 ilustra os softwares e as tecnologias utilizadas neste trabalho.

Figura 22 - Ferramentas Tecnológicas



Fonte: Elaboração do autor, 2013.

As tecnologias apresentadas na Figura 22 foram utilizadas na construção do sistema de captura de dados, sistema de persistência de dados e nos testes de desempenho realizados no estudo de caso.

5.2.1 SQL Server 2008

O SQL Server 2008 é um banco de dados relacional desenvolvido pela *Microsoft*, uma plataforma de dados produtiva, confiável e inteligente. Ele é destinado a aplicações com arquitetura cliente-servidor fornecendo uma visão da plataforma de dados da Microsoft e ajudando a gerenciar quaisquer dados. Ele permite armazenar dados estruturados, semiestruturados e não estruturados, como documentos, imagens e música, diretamente no banco de dados.

Esse banco de dados fornece um conjunto sofisticado de serviços integrados que permitem explorar os dados de diversas formas, como consultas, pesquisas, sincronização, relatórios e análises. Os dados podem ser armazenados e acessados através de servidores ou em desktops e dispositivos móveis, permitindo controle sobre os dados onde quer que eles estejam armazenados.

Para este projeto foi utilizado o SQL Server 2008 Enterprise, a versão mais completa deste banco de dados.

5.2.2 Neo4J

Neo4j é um banco de dados orientado a grafos *open source* desenvolvido pela empresa Neo Technology. Ele armazena dados em nós conectados entre si através de relacionamentos, dessa forma, uma única instância do servidor pode gerenciar bilhões de nós e relacionamentos em um mesmo grafo. O banco de dados Neo4J pode ser distribuídos entre vários servidores fornecendo uma configuração de alta disponibilidade. (NEO4J, [2013?], tradução nossa).

Neo4j concentra-se mais sobre os relacionamentos estabelecidos entre os valores do que as semelhanças entre conjuntos de valores. Deste modo, é possível armazenar dados altamente variáveis de uma forma natural e fácil, sem que a estrutura da base de dados tenha que ser alterada a cada novo tipo de dado inserido.

Bases de dados orientadas a grafo são extremamente eficientes ao lidar com um número expressivo de dados, pois as travessias no grafo são executadas com uma velocidade constante, independentemente do tamanho total do mesmo.

A fim de garantir a integridade dos dados e seu comportamento ACID, ele impõe que todas as operações que modificam dados ocorram através uma transação. Esse comportamento se estende desde uma única instância de grafo embarcado até estruturas com vários servidores de alta disponibilidade.

Neo4j é um banco de dados orientado a grafos de alto desempenho, robusto, escalável e pequeno o suficiente para ser incorporado em praticamente qualquer aplicativo. (BATRA; TYAGI, 2012, tradução nossa).

5.2.3 Cypher

Cypher é uma linguagem de consulta declarativa para bases de dados orientadas a grafos que permite a busca e a manipulação de dados de forma eficiente, sem que o usuário necessite projetar as travessias realizadas no grafo internamente no código do sistema.

Ele foi projetado para desenvolvedores de sistemas e profissionais de banco de dados. É uma linguagem que está alinhada diretamente com a forma como nós intuitivamente descrevemos os grafos. (ROBINSON; WEBBER; EIFREM, 2013, tradução nossa).

A sua estrutura é baseada em uma rede e sua linguagem é formatada em Inglês, o que a torna uma linguagem de fácil compreensão e autoexplicativa. O seu desenvolvimento foi inspirado em uma série de abordagens diferentes e a maioria das suas expressões são baseadas no SQL.

Essa linguagem de consulta de dados, busca expressar de forma simples o que o usuário deseja buscar em um grafo, não a forma que a busca deve ser realizada, isso faz com que a preocupação com a otimização e desempenho da consulta não fique a cargo do usuário.

Ao utilizar o Cypher, é possível realizar uma busca na base de dados com o intuito de encontrar dados de um determinado padrão facilmente. Pois ele define uma ou mais partes do padrão de dados para especificar o local de inicio da pesquisa no grafo e em seguida flexiona a busca nas partes ao redor para encontrar dados correspondentes.

O Cypher tem o objetivo de ser familiar aos usuários para que eles se desenvolvam rapidamente ao longo da curva de aprendizado. Ao mesmo tempo, é diferente o suficiente para enfatizar que estamos lidando com grafos, e não com dados relacionais. (ROBINSON; WEBBER; EIFREM, 2013, tradução nossa).

5.2.4 SQL Server Management Studio

O SQL Server Management Studio é a ferramenta de cliente que você usa para desenvolver e executar códigos T-SQL no SQL Server. (BEN-GAN, 2012, tradução nossa).

Ele é um sistema de gerenciamento de dados produzido pela *Microsoft,* que fornece um conjunto avançado de recursos, segurança de dados e ferramentas de monitoramento de desempenho para aplicativos e repositórios locais de dados.

O Management Studio fornece um ambiente de desenvolvimento integrado para acessar, administrar, configurar, gerenciar e desenvolver os componentes do SQL Server. Ele combina um vasto grupo de ferramentas e editores de scripts, para fornecer acesso ao SQL Server a desenvolvedores e administradores de bancos de dados. (MICROSOFT, 2009).

Combinando os recursos do Enterprise Manager, Query Analyzer e Analysis Manager, em um único ambiente, o SQL Server Management Studio é uma ferramenta completa e confiável, fornecendo um ambiente completo para análise de dados. Além disso, ele trabalha com todos os componentes do SQL Server, como Integration Services e Reporting Services.

5.2.5 Java

Java é a linguagem de programação orientada a objetos mais utilizada no mundo e revolucionou o desenvolvimento de software, pois se trata de uma linguagem multiplataforma, distribuída de forma gratuita, com suporte a técnicas de engenharia de software e que ajuda os desenvolvedores a libertarem a sua criatividade. (DEITEL, 2003).

Java é uma tecnologia de código aberto independente de plataforma. Essa tecnologia dá a possibilidade de desenvolver programas de mais alta qualidade, como jogos aplicativos entre outros. Quando se desenvolve em Java, utiliza-se sua linguagem de programação que será executada em um ambiente de distribuição Java chamado Maquina Virtual.

5.2.6 PHP

PHP, que significa Hypertext Preprocessor, é uma linguagem de programação interpretada, que é especialmente interessante para desenvolvimento para a Web e pode ser mesclada dentro do código HTML. O objetivo principal do PHP é permitir que os desenvolvedores escrevam páginas que serão geradas dinamicamente. A melhor coisa em usar PHP está no fato de ele ser extremamente simples para um iniciante, mas oferece muitos recursos para o programador profissional. (PHP, 2013).

Essa linguagem pode ser utilizada na grande maioria dos sistemas operacionais, incluindo Microsoft Windows, Mac OS X e Linux, e também é suportada pela maioria dos servidores web atuais, incluindo Microsoft Internet Information Server, Apache, Personal Web Server, Oreilly Website Pro Server, Netscape, iPlanet Servers, Caudium, e outros.

Utilizando o PHP, você tem a total liberdade para escolher o sistema operacional e o servidor web que desejar. Da mesma forma, você pode escolher entre utilizar programação estrutural ou programação orientada a objeto, ou ainda uma mistura deles.

5.2.7 NetBeans

NetBeans IDE é uma ferramenta de desenvolvimento modular para uma ampla gama de tecnologias de desenvolvimento de aplicações. Ele oferece diversas funcionalidades de apoio ao desenvolvedor de software, além de modelos de código, dicas de codificação, e ferramentas de refatoração. Se trata de uma IDE completa, que suporta várias linguagens como Java, C, C++, XML, HTML, PHP, JavaScript e JSP. NetBeans oferece suporte abrangente para as mais recentes tecnologias Java, onde podemos desenvolver aplicações web, desktop e para dispositivos móveis. É a primeira IDE fornecendo suporte para JDK 7, Java EE 7 e JavaFX 2. (NETBEANS, [2013?], tradução nossa).

Além de poder ser instalado em todos os sistemas operacionais que suportam Java, é gratuito e de código aberto, características que o fazem ter uma grande comunidade de usuários e desenvolvedores ao redor do mundo.

5.2.8 JSON

JSON, JavaScript Object Notation, é um formato leve de intercâmbio de dados baseado em um subconjunto da linguagem de programação JavaScript. No principio, esse formato de dados foi concebido como uma alternativa mais simples a utilização do padrão XML para troca de dados, pois é menor e mais rápido e fácil de analisar. Se trata de uma notação simples, que independe de linguagem de programação, porem ela utilizada padrões comuns aos programadores da linguagem C. (CROCKFORD, 2006, tradução nossa).

Atualmente, a maioria das linguagens de programação suporta o formato de arquivo JSON em suas importações e exportações de dados. Manipuladores de arquivos e bibliotecas JSON já são encontrados para diversas linguagens de programação.

5.2.9 Google-GSON

Gson Java é uma biblioteca desenvolvida pelo Google que pode ser utilizada para converter objetos Java em representação de dados através do padrão JSON. Também pode ser utilizada para converter um arquivo de dados no padrão JSON para um objeto Java. (GOOGLE-GSON, [2013?], tradução nossa).

Ele se trata de um projeto de código aberto que facilita a conversão de dados, pois não exige que se utilize *annotations* no código Java, suporta totalmente o uso de Java Generics, permite a representação personalizada de objetos e suporta o uso de objetos complexos.

5.3 ESTUDO DE CASO

Esta seção apresentará o referencial teórico de análise social utilizado para fundamentar o estudo de caso deste trabalho. Além disso, será demonstrada a infraestrutura montada para o desenvolvimento da solução proposta para o estudo de caso.

5.3.1 Análise Social

A perspectiva de análise de uma rede social engloba teorias, aplicações e modelos que são expressos em conceitos ou processos relacionais. Dessa forma, as relações definidas por vínculos entre os atores são fundamentais para as teorias de rede. A principal característica de teorias de redes sociais é que as unidades sociais estejam ligadas uma as outras por várias relações. (WASSERMAN; FAUST, 1994, tradução nossa).

Análise social é estudo das interações sociais humanas. Então, a análise das redes sociais pode ser utilizada para investigar padrões de parentesco, estruturas de uma comunidade, organização de outras redes sociais e etc. Estas redes podem ser associadas a diversas áreas como: empresas, grupos familiares, as estruturas de comando e controle, redes acadêmicas e outros. (COOK; HOLDER, 2007, tradução nossa).

Segundo Wasserman e Faust (1994, tradução nossa), as redes sociais e os métodos de análise das mesmas têm atraído um considerável interesse e curiosidade por parte da comunidade das ciências sociais e comportamentais nas últimas décadas. Grande parte desse interesse pode ser atribuído ao foco atraente da análise de redes sociais nas relações entre entidades sociais, e sobre os padrões e as implicações dessas relações. Muitos pesquisadores perceberam que a perspectiva do desenvolvimento de redes sociais permite a criação de uma nova metodologia para responder a perguntas de investigação em ciências sociais e comportamentais, dando uma definição formal e precisa para aspectos do ambiente estrutural, político, econômico ou social. Do ponto de vista da análise de redes sociais, o ambiente social pode ser expresso como padrões ou regularidades nas relações entre as unidades que o envolvem.

Um dos primeiros estudos neste contexto incluem o trabalho de Rice na análise de comunidades de indivíduos com base em seus preceitos políticos e padrões de voto. Um estudo mais recente com foco na estrutura da rede de blogs políticos foi discutido por Adamic e Glance. Weiss e Jacobson analisaram grupos de trabalho dentro de uma agência do governo. O estudo de um clube de karate realizado pelo antropólogo Zachary, um grafo bem conhecido e regularmente usado, inclui um exemplo de fissão de comunidade bem conhecido e, portanto, é objeto de muitos estudos. Um estudo realizado por Bech e Atalay analisa a rede social de empréstimos entre instituições financeiras para entender como as interações entre as várias comunidades irá afetar a saúde do sistema como um todo. A maior parte desses estudos estão focados simplesmente em compreender a estrutura social e sua evolução das redes sociais. (AGGARWAL, 2011, tradução nossa).

Segundo Abraham, Hassanien e Snásel (2009, tradução nossa), as relações sociais são componentes importantes da vida humana e historicamente foram estabelecidas de acordo com limitações de tempo e espaço, estas restrições foram parcialmente removidas devido à evolução da internet e sua propagação através dos anos. A internet permitiu que as pessoas se organizassem em redes sociais virtuais da mesma forma que eles se organizam em redes sociais no mundo real. Do ponto de vista da sociologia, as redes sociais são definidas com base em suas características físicas ou a força do tipo de relacionamento.

Analistas de redes sociais muitas vezes representam a rede através de um grafo. Na sua forma mais simples, um grafo de rede social contém nós que representam atores e arestas que representam relações ou interações entre os atores. (COOK; HOLDER, 2007, tradução nossa).

Um dado de uma rede social consiste em uma variável estrutural mensurada a partir de um conjunto de atores. As teorias de motivação do estudo de rede geralmente determinam quais variáveis devem ser mensuradas, e, muitas vezes, que técnicas são as mais adequadas para a sua medição. Existem dois tipos de variáveis que podem ser incluídas em um conjunto de dados da rede: estruturais e de composição. Variáveis estruturais são medidas em pares de atores e são fundamentais para os conjuntos de dados de redes sociais, pois elas medem ligações entre pares de um tipo específico de atores. As variáveis de composição são medidas de atributos dos atores, que são definidas individualmente para cada ator. (WASSERMAN; FAUST, 1994, tradução nossa).

O advento das redes sociais online tem expandido os horizontes da análise social nessa última década. Muitas redes sociais online como Twitter, LinkedIn, e Facebook, tem se tornado cada vez mais populares. Além disso, um grande número de redes multimídia, tais

Flickr e Instagram também cresceram em popularidade nos últimos anos. Muitas dessas redes sociais são extremamente ricas em conteúdo, e possuem uma quantidade enorme de dados publicados que podem ser aproveitados para a análise. A riqueza de informações destas redes oferece oportunidades enormes para a análise de dados no contexto das relações sociais. (AGGARWAL, 2011, tradução nossa).

As redes sociais virtuais permitem que às pessoas troquem informações por meio de uma interação fácil, universal, barata e confiável. Essas interações estão ligadas a possibilidade de acessar informações específicas sobre as diferentes áreas de interesse. Dentro de redes sociais virtuais, as pessoas podem fornecer informações ou obter informações. Portanto, essas fontes de informação são socialmente úteis, pois permitem que as pessoas estabeleçam com facilidade um contato entre si. A ideia de interação estabelecida nas redes sociais remete a possibilidade das pessoas compartilharem, uns com os outros, seus interesses e experiências de vida. (ABRAHAM; HASSANIEN; SNÁSEL, 2009, tradução nossa).

De acordo com Aggarwal (2011, tradução nossa), a popularidade das mídias sociais é enorme. Como exemplo, pode-se citar o Facebook, que durante os primeiros seis anos de operação atingiu mais de 400 milhões de usuários ativos. Assim, os dados disponíveis nessas redes sociais virtuais podem nos apresentar dados relacionados às redes sociais e comunidades que antes não eram possíveis verificar nessa escala e extensão. Portanto as mídias digitais podem transcender limites para estudar as relações humanas de algumas comunidades sem pesquisas explícitas.

Para Wasserman e Faust (tradução nossa, 1994), existem várias maneiras de se representar uma rede social. Os grafos são uma ótima forma para representar os atores e suas relações. Estruturar redes sociais através de grafos é uma maneira simples para se analisar os atores e relações. Matemáticos e estatísticos, como Bock, Harary, Katz, e Luce foram os primeiros a representar as redes sociais através grafos dirigidos.

A representação através de grafos é natural para os dados extraídos de sites de redes sociais onde as pessoas criam uma rede de amigos, colegas ou parceiros de negócios. Esse tipo de modelagem permite a aplicação da teoria matemática de grafos, métodos tradicionais de análise de redes sociais e de mineração de dados em grafos. No entanto, o enorme tamanho de um grafo utilizado para representar as redes sociais pode apresentar desafios para o processamento automatizado dos dados devido à necessidade de uma poderosa estrutura de hardware. Outro desafio na aplicação de processos automatizados para a mineração de dados em redes sociais inclui lidar a constante mudança de conteúdo e estrutura. (AGGARWAL, 2011, tradução nossa).

A característica mais básica da medição de uma rede é o uso de informações estruturais para estudar ou testar teorias, pois muitos métodos de análise de redes fornecem definições formais e descrições das propriedades estruturais dos atores da rede. Todos esses conceitos utilizados para analisar a estrutura de uma rede são quantificados, considerando as relações de medida aplicadas aos atores dessas redes. Os dados obtidos nas redes sociais requerem medições em laços entre atores, no entanto, os atributos dos atores também podem ser coletados e mensurados. (WASSERMAN; FAUST, 1994, tradução nossa).

5.3.2 Infraestrutura

Esta seção irá demonstrar a infraestrutura montada para o desenvolvimento da solução proposta no estudo de caso. A estrutura desenvolvida é dividida em captura de dados, persistência de dados e consulta de dados.

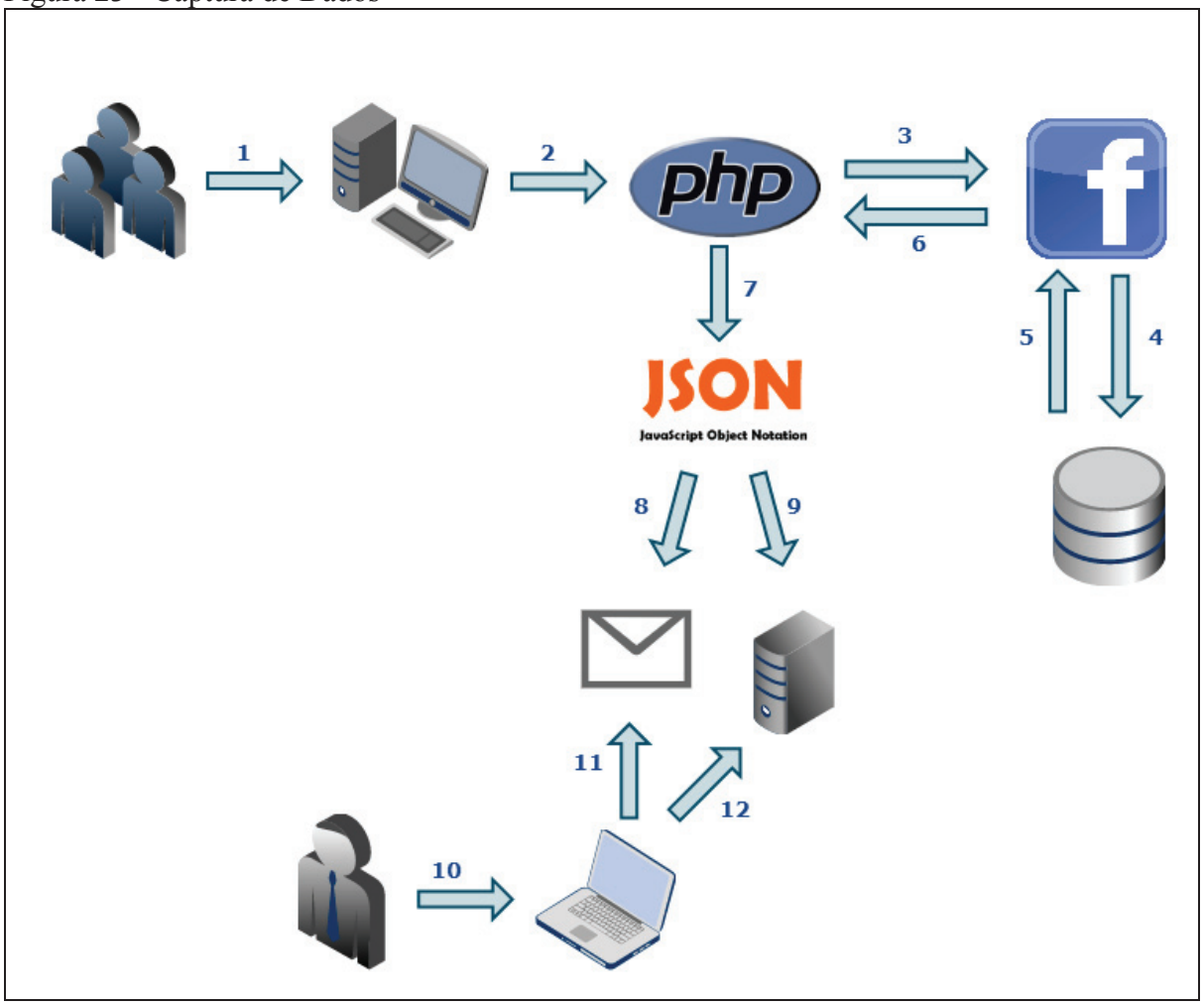
5.3.2.1 Captura de Dados

Após a definição que os dados utilizados no estudo de caso seriam provenientes do Facebook, iniciaram-se os testes na Graph API disponibilizada por essa rede social a desenvolvedores de sistemas. Durante esse período, verificou-se que a extração manual desses dados seria um impeditivo para a colaboração dos usuários da rede social que quisessem colaborar com este trabalho. Como um dos objetivos do desenvolvimento do estudo de caso era atingir o maior número de colaborações possível, esse processo não poderia ser complicado para o usuário.

A fim de realizar a captura automática de dados do Facebook, foi desenvolvido um sistema que conecta a base de dados da rede social, consulta os dados necessários e os retorna em forma de arquivo JSON.

Abaixo na Figura 23 podemos visualizar a estrutura do sistema de captura de dados.

Figura 23 - Captura de Dados



Fonte: Elaboração do autor, 2013.

Abaixo segue o processo detalhado realizado nessa fase do projeto:

1 - Os usuários do Facebook dispostos a colaborar fornecendo seus dados para a pesquisa deverão acessar os seus computadores.

2 - Os usuários deverão acessar o site www.guilhermealvarez.com.br onde serão direcionados para a aplicação desenvolvida em PHP que realiza a conexão com o Facebook. No final da página de apresentação, há um botão que da início ao acesso de dados da rede social.

3 - Após clicar no botão, a conexão com o Facebook é iniciada e a solicitação de autorização de acesso aos dados do perfil público e da lista de amigos do usuário é enviada ao mesmo. Ao obter a autorização, o sistema envia uma

solicitação HTTP GET ao Facebook com o token da autorização e os dados a serem consultados.

4 - Assim que o token é confirmado, o Facebook envia a requisição de consulta de dados na linguagem FQL, Facebook Query Language, a base de dados da rede social.

5 - Os dados são consultados no banco de dados e retornados para a Graph API.

6 - A Graph API converte as informações recebidas da base de dados em formato de dados JSON e envia a aplicação PHP do site.

7 - O sistema recebe o retorno de dados do Facebook, ajusta seu conteúdo removendo informações desnecessárias e grava os dados em um arquivo no formato JSON.

8 - A aplicação envia o arquivo de dados para o e-mail do autor desse trabalho.

9 - A aplicação disponibiliza o arquivo de dados no servidor onde a aplicação está hospedada.

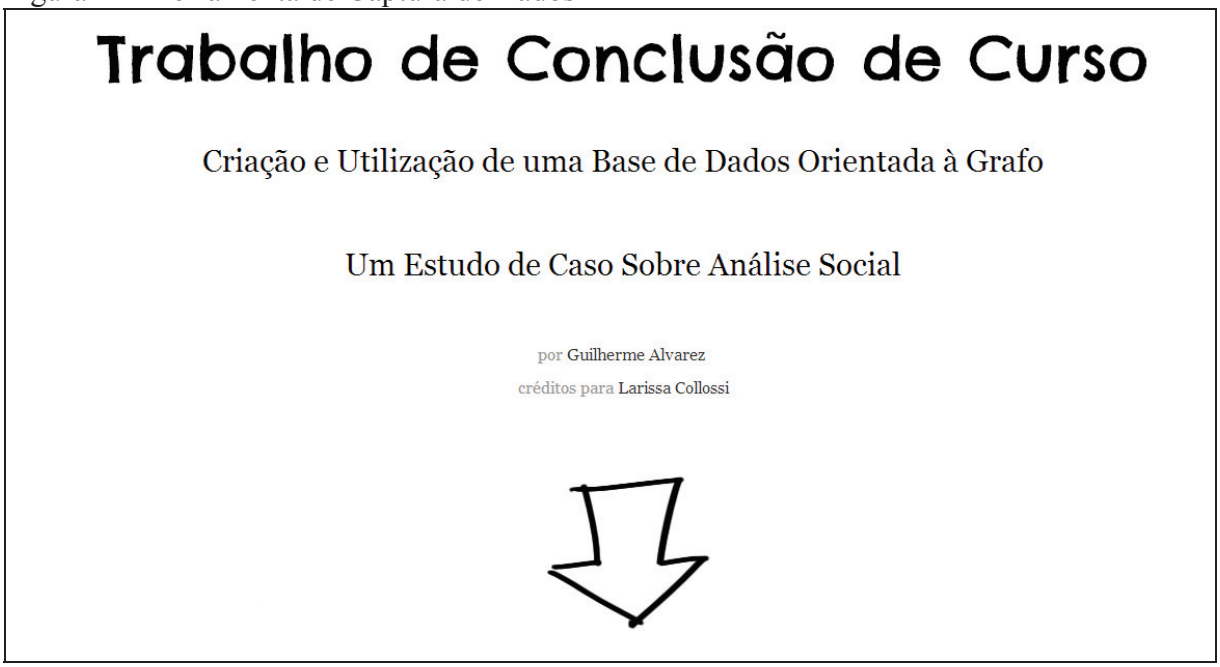
10 - O autor desse trabalho acessa o seu computador para apanhar os arquivos de dados.

11 - Os dados enviados por e-mail são acessados pelo autor do trabalho.

12 - Os dados armazenados no servidor são acessados pelo autor do trabalho.

A Figura 24 ilustra a tela de apresentação do sistema de captura de dados hospedado no site guilhermealvarez.com.br. No site há uma breve apresentação do trabalho e do estudo de caso para que os usuários entendam o contexto do projeto para o qual estão fornecendo seus dados.

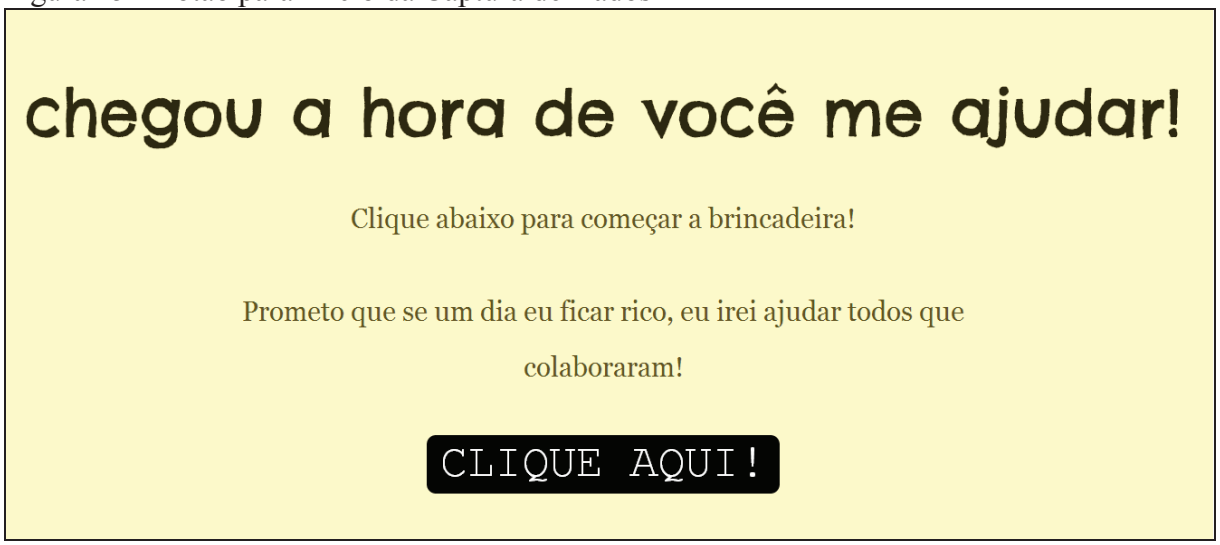
Figura 24 - Ferramenta de Captura de Dados



Fonte: Elaboração do autor, 2013.

A Figura 25 apresenta o botão que inicia a conexão com o Facebook e a captura de dados dos usuários.

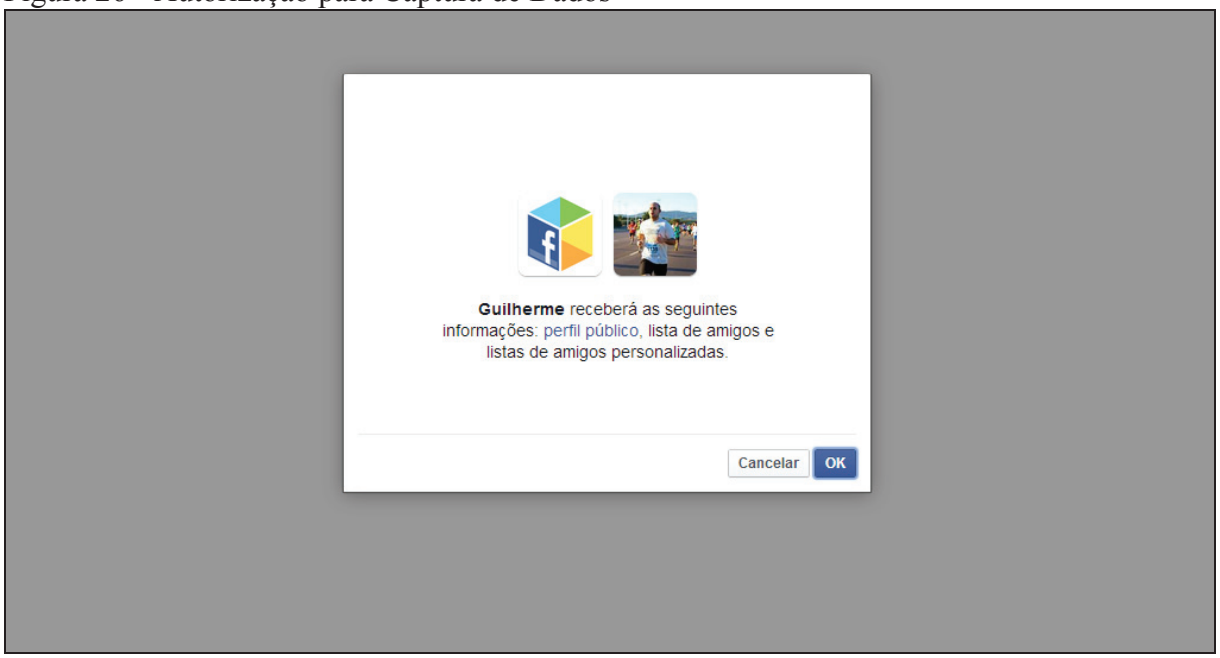
Figura 25 - Botão para Inicio da Captura de Dados



Fonte: Elaboração do autor, 2013.

Após clicar no botão localizado no final do site, o sistema envia a requisição de autorização para captura de dados do perfil público do usuário e a sua lista de amigos, conforme é ilustrado na Figura 26.

Figura 26 - Autorização para Captura de Dados



Fonte: Elaboração do autor, 2013.

Além de ter simplificado e facilitado à conexão com o Facebook para a consulta de dados dos usuários dispostos a colaborar com o trabalho, o web site da ferramenta de captura de dados trouxe informações sobre a pesquisa relatando, como e para que, as informações seriam utilizadas.

5.3.2.2 Persistência de Dados

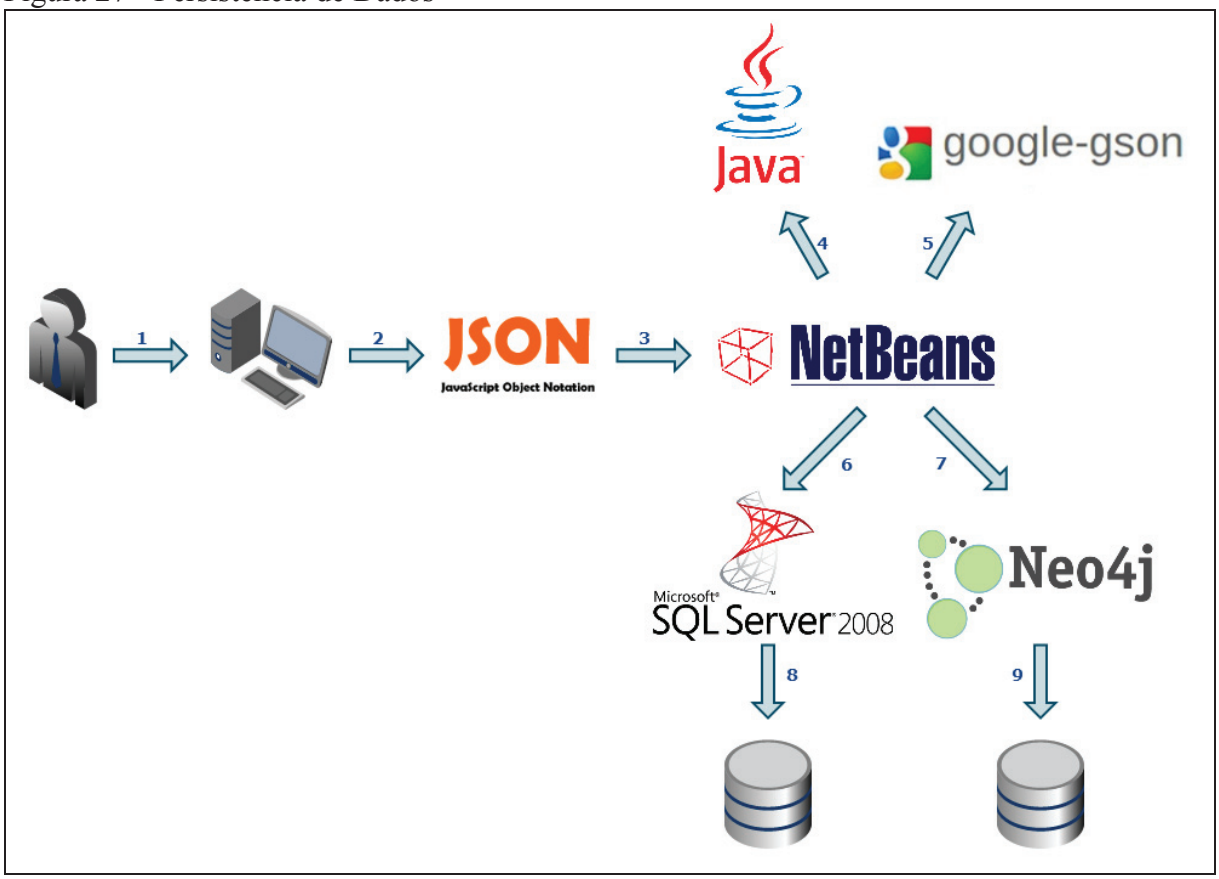
Para dar inicio a analise e validação de dados, é necessário que as informações estejam devidamente inseridas nas bases de dados criadas para o estudo de caso. Como nenhuma das bases de dados utilizadas neste trabalho realiza a importação direta de dados através de um arquivo JSON, seria necessário criar uma estrutura que realizasse essa tarefa.

A fim de que os dados disponibilizados pelos usuários fossem persistidos de maneira eficiente nas bases de dados utilizadas neste trabalho, foi desenvolvido um sistema que realiza a leitura e identificação das informações contidas nos arquivos JSON e as insere nas bases de dados. Esse sistema foi desenvolvido utilizando a linguagem de programação Java através da IDE NetBeans. Para realizar a identificação de dados nos arquivos do padrão JSON, foi utilizada a biblioteca GSON, desenvolvida pela empresa multinacional Google.

O GSON possui métodos que fazem a leitura dos dados contidos no arquivo JSON e os converte em objetos Java, além de possibilitar a criação de uma lista com esses objetos, o que simplifica a persistência de dados.

Segue abaixo a Figura 27, na qual podemos visualizar a estrutura do sistema de persistência de dados.

Figura 27 - Persistência de Dados



Fonte: Elaboração do autor, 2013.

Abaixo segue o processo detalhado realizado na fase de persistência de dados deste trabalho:

1 - O autor deste trabalho acessa o seu computador para iniciar a persistência de dados.

2 - Os arquivos JSON com os dados disponibilizados pelos usuários do Facebook são inseridos em uma pasta do sistema de persistência de dados.

3 - O sistema de persistência de dados faz a consulta aos arquivos disponibilizados.

4 - O sistema de persistência utiliza a linguagem de programação Java para realizar as suas funções.
5 - O sistema desenvolvido utiliza a biblioteca GSON para realizar a leitura e conversão dos dados JSON em objetos Java.
6 - O sistema estabelece uma conexão JDBC com o SQL Server 2008 para realizar a persistência de dados.
7 - Durante a fase persistência de dados o banco de dados Neo4J trabalha embarcado no sistema desenvolvido, utilizando a classe EmbeddedGraphDatabase da API do Neo4J.
8 - Os objetos convertidos dos arquivos de dados são persistidos na base de dados do SQL Server 2008 utilizando a conexão JDBC estabelecida.
9 - Os objetos convertidos dos arquivos de dados são persistidos na base de dados Neo4J.

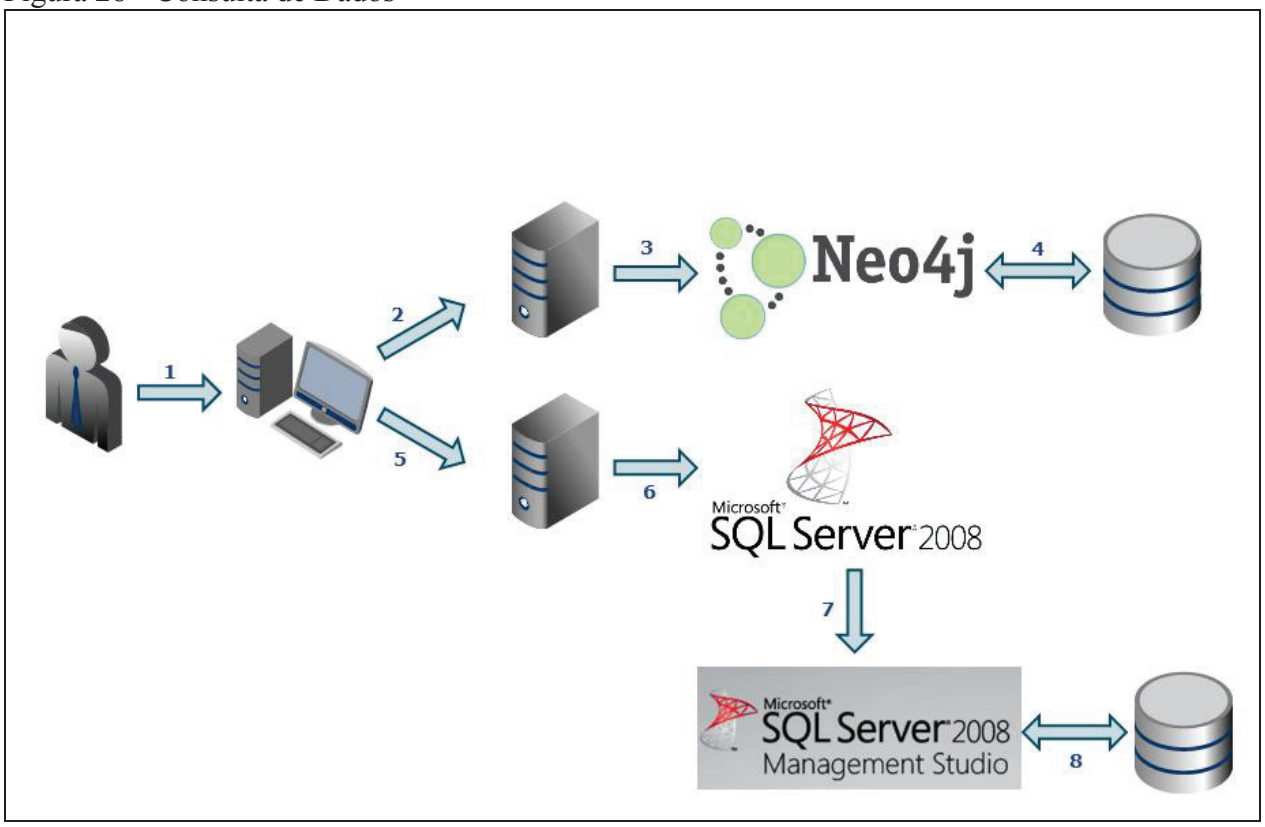
5.3.2.3 Consulta de Dados

Para concluir o estudo de caso proposto neste trabalho, os dados inseridos nas bases de dados precisam ser mensurados e analisados. A fase de consulta de dados é uma das partes mais importantes deste trabalho, pois os resultados obtidos serão utilizados a fim de validar a solução proposta para o estudo de caso.

A consulta de dados foi realizada diretamente nos servidores onde as bases de dados foram criadas utilizando os Sistemas Gerenciadores de Bancos de Dados disponibilizados para cada banco de dados. Para as consultas de dados na base relacional SQL Server 2008 foi utilizado o SQL Server Management Studio e para a base orientada a grafos foi utilizado o console disponível na interface gráfica do Neo4J.

Segue a Figura 28, na qual podemos visualizar a estrutura do sistema de persistência de dados.

Figura 28 - Consulta de Dados



Fonte: Elaboração do autor, 2013.

Abaixo segue o processo detalhado realizado na fase de consulta de dados deste trabalho:

1 - O autor deste trabalho acessa o seu computador para iniciar a consulta de dados.

2 - O servidor onde a base de dados Neo4J está alocada é acessado.

3 - Através do servidor o serviço do banco de dados Neo4J é iniciado e a interface gráfica para controle e acesso aos dados é acessada.

4 - Através do console disponível na interface gráfica, a consulta de dados é realizada utilizando a linguagem Cypher, desenvolvida para a base orientada a grafos Neo4J. Após a base de dados processar o resultado da consulta, o mesmo é retornado e apresentado na interface gráfica.

5 - O servidor onde a base de dados SQL Server 2008 está alocada é acessado.

6 - Através do servidor o serviço do banco de dados SQL Server 2008 é iniciado.

7 - Após o serviço de dados do SQL Server 2008 estar ativo, o SQL Server Management Studio é iniciado no servidor.

8 - A consulta de dados é realizada utilizando a linguagem SQL através do SGBD e enviada ao banco de dados. Após a base de dados processar o resultado da consulta, o mesmo é retornado no SGBD.

5.4 GRAFO

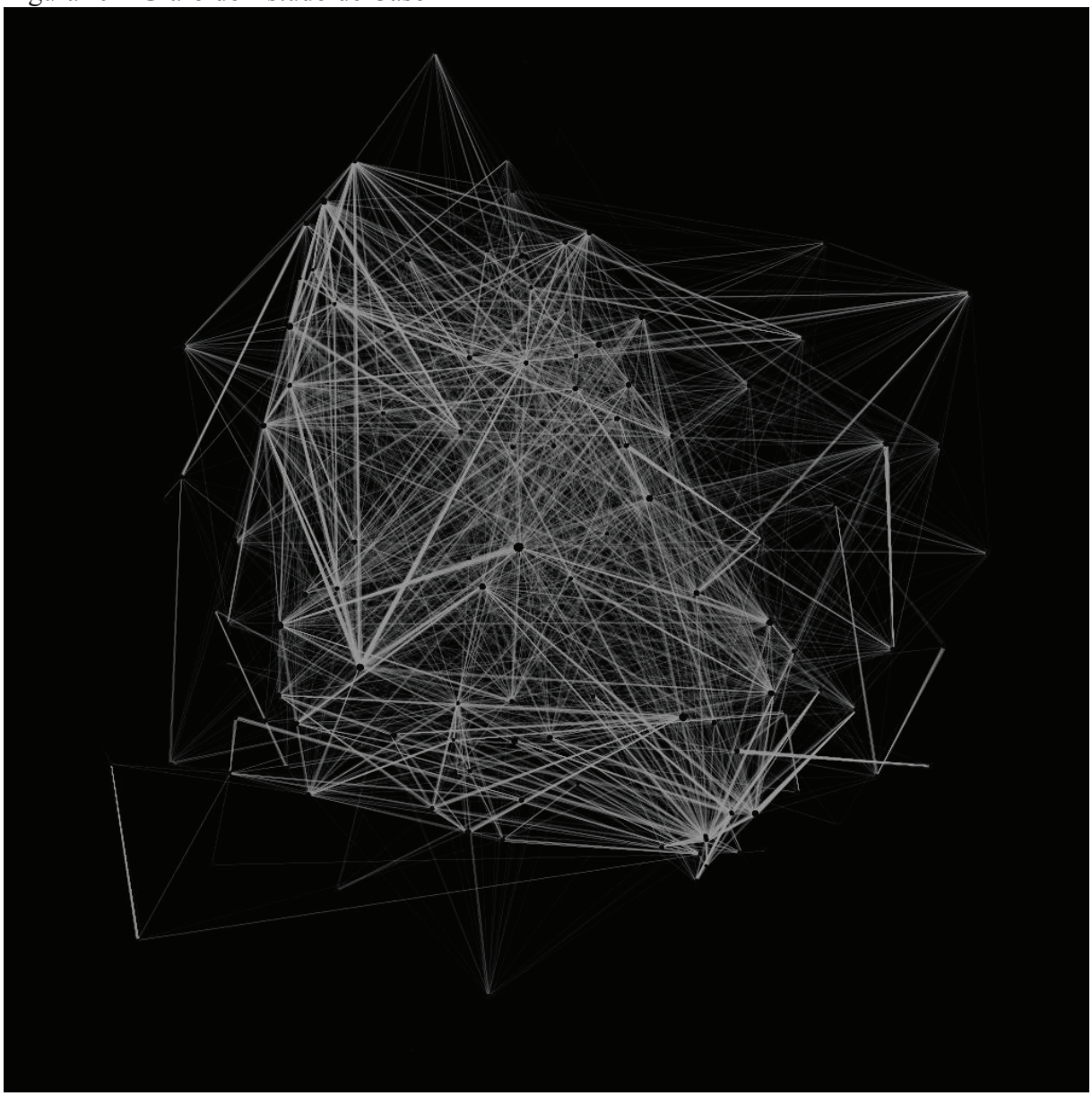
Os dados obtidos através do sistema de captura de dados foram utilizados na construção de um grafo que representa a rede de relações entre os indivíduos que colaboraram com este trabalho e seus amigos do Facebook. O grafo gerado possui 57673 vértices e 119353 relações de amizade estabelecidas entre os vértices.

Para apresentar a imagem do grafo foi utilizada a ferramenta Gelphi, que permite importar bases de dados Neo4J e aplicar algoritmos para a visualização de redes complexas. Como o grafo possui um grande número de vértices e arestas, foi aplicado o algoritmo para de força direcionada para representação de grafos de larga escala contido no *plugin* OpenOrd.

Devido ao tamanho e complexidade do grafo construído, não foi possível visualizar o mesmo na ferramenta interativa disponibilizada pelo Neo4J e bibliotecas JavaScript para representação de grafos como o Sigma.js.

Na Figura 29 pode-se visualizar o grafo utilizado no estudo de caso deste trabalho.

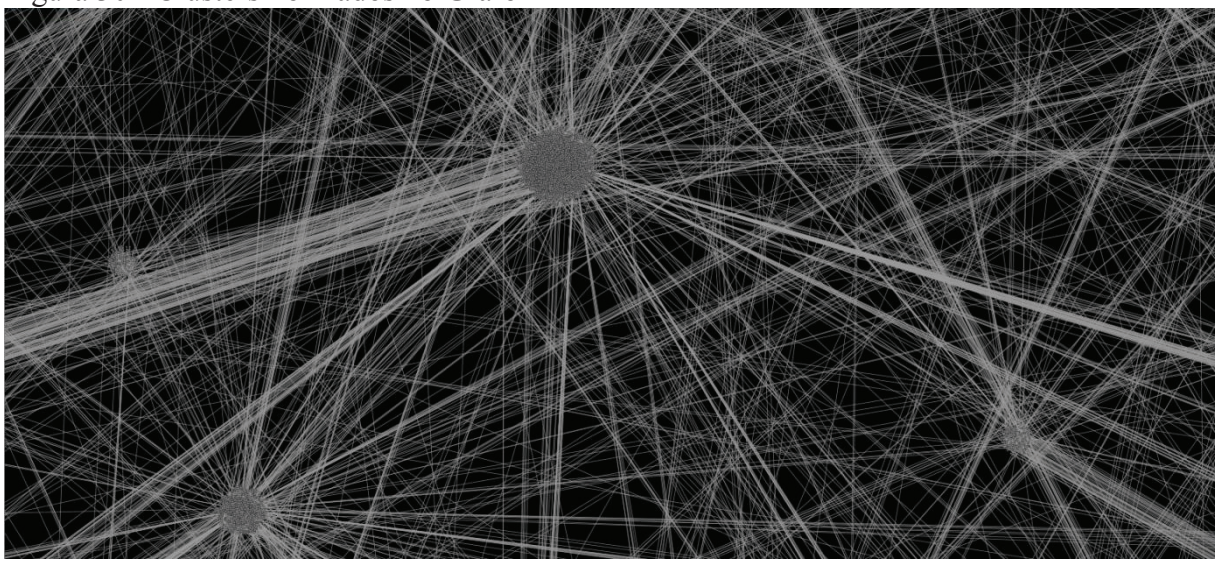
Figura 29 - Grafo do Estudo de Caso



Fonte: Elaboração do autor, 2013.

Após o algoritmo de força direcionada ter sido aplicado no grafo, os clusters formados puderam ser visualizados facilmente. A Figura 30 ilustra alguns dos clusters formados e os seus relacionamentos.

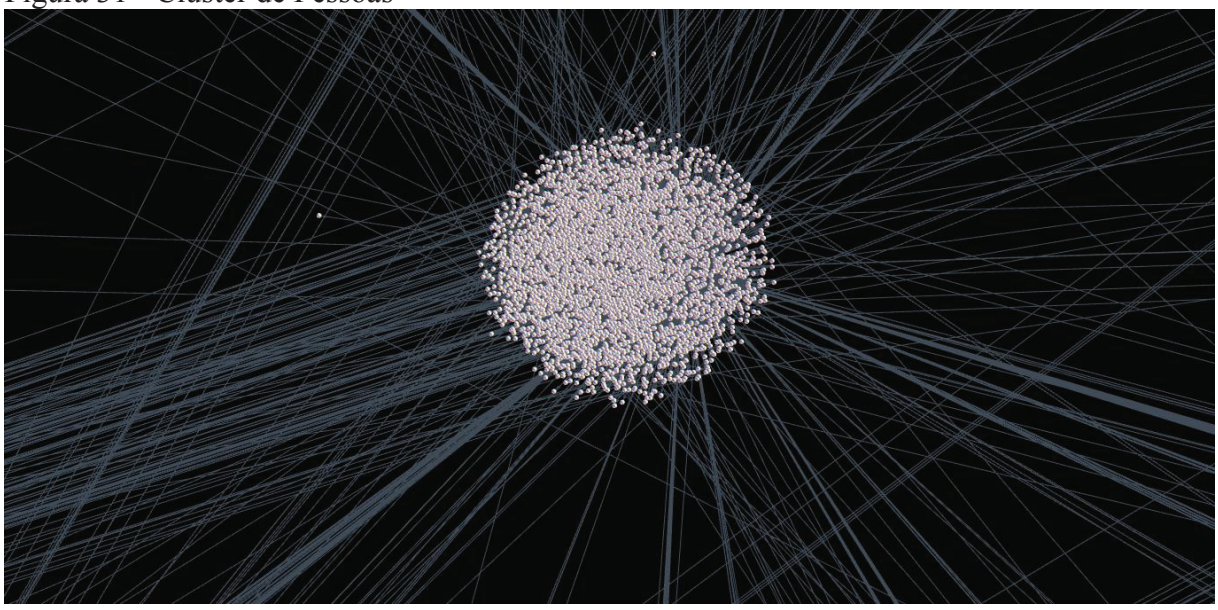
Figura 30 - Clusters Formados no Grafo



Fonte: Elaboração do autor, 2013.

A Figura 31 ilustra claramente a grande quantidade de vértices agrupados em um dos clusters criados no grafo do estudo de caso.

Figura 31 - Cluster de Pessoas



Fonte: Elaboração do autor, 2013.

A representação do grafo torna possível a visualização da sua estrutura, facilitando a identificação de clusters e possibilitando a construção de modelos preditivos e a análise de correlações e padrões de dados.

5.5 VALIDAÇÃO

As relações sociais são uma parte essencial da vida humana e historicamente estão ligadas de acordo com limitações de tempo e espaço. Devido à evolução da internet e sua difusão entre as pessoas, estas restrições foram parcialmente removidas. (ABRAHAM; HASSANIEN; SNÁSEL, 2009, tradução nossa).

Devido às redes sociais virtuais terem se tornado extremamente populares nos últimos anos, quantidades enormes de dados de diversos tipos tem sido compartilhados entre os indivíduos diariamente. Ela se tornou uma área interdisciplinar de pesquisa, na qual psicólogos, antropólogos, sociólogos, estatísticos e outros, podem visualizar os dados da rede, a fim de analisar as relações entre pessoas, grupos, organizações e entidades.

Um aspecto chave de muitas das redes sociais online é que elas são ricas em dados, proporcionando desafios e oportunidades sem precedentes a partir da perspectiva de descoberta de conhecimento e mineração de dados. (AGGARWAL, 2011, tradução nossa).

Dentro do contexto de analise das redes e relações sociais, existem dois tipos primários de dados que podem ser analisados:

- Análise de conteúdo da rede, baseada nas informações contidas na rede, na qual as propriedades dos nós e relacionamentos serão analisados revelando características e comportamentos dos indivíduos inseridos na rede.
- Análise estrutural da rede, baseada na estrutura de ligação da rede, na qual podemos desenvolver uma análise do comportamento de ligações da rede, a fim de determinar os nós importantes, comunidades, links e regiões da rede em evolução.

No estudo de caso apresentado neste trabalho às duas formas de análise serão abordadas demonstrando a versatilidade e facilidade da utilização de bases de dados orientadas a grafo.

5.5.1 Testes de Desempenho

Ao implementar uma rede social no modelo de dados relacional, normalmente tem-se duas tabelas para armazenar dados, uma para armazenar informações dos indivíduos, e outra que armazena as relações entre os mesmos. Na base relacional utilizada neste estudo de caso, a tabela Pessoa contém os dados dos usuários do Facebook e a tabela Amizade que armazena a relação de amizade entre os indivíduos, referenciando a tabela Pessoa através de chave estrangeira.

O Facebook apresenta sugestões aos usuários de possíveis amigos até um determinado grau de profundidade, baseando-se nas relações entre os indivíduos da rede. Porém a busca desse tipo de dados em bases de dados relacionais não se trata de algo simples, pois para encontrar os amigos de amigos de um usuário, é necessário o uso das operações de junção de dados. Portanto, para se alcançar o terceiro nível de amizade, é preciso utilizar três operações de junção, e assim por diante.

Nos casos em que se deseja localizar amigos em comum ou amigos de amigos, o objetivo da pesquisa é consultar apenas os amigos de amigos de um determinado usuário. Porém ao utilizar as junções, a base de dados avalia todos os dados contidos na tabela Amizade, descartando posteriormente os dados os desnecessários, o que gera uma perda de desempenho.

No Quadro 3 podemos visualizar os resultados obtidos nos testes de desempenho realizados na base de dados relacional, nos quais foram utilizados diferentes profundidades e quantidades de dados.

Quadro 3 - Testes Base de Dados Relacional

Profundidade	10095 Usuários	20349 Usuários	30522 Usuários	40053 Usuários	57673 Usuários
2	10ms	11ms	13ms	15ms	24ms
3	24ms	25ms	41ms	59ms	48ms
4	41ms	71ms	82ms	119ms	193ms

Fonte: O Autor (2013)

Pode-se verificar que a base de dados relacional trata as consultas com profundidades 2 e 3 de maneira aceitável, mas ao utilizar profundidades acima de 4, há uma queda significativa no desempenho. Dessa forma, é possível identificar que em consultas que

envolvem várias junções recursivas, as bases de dados relacionais normalmente apresentam queda de desempenho.

Ao realizar uma pesquisa para encontrar todos os amigos de amigos de um indivíduo em profundidade 4, o banco de dados relacional gera o produto cartesiano da tabela Amizade quatro vezes. Aplicando esse contexto à tabela Pessoa utilizada no estudo de caso, com 57673 registros, o conjunto resultante terá 57673^4, ou seja, 11.063.439.658.308.155.000 linhas. Para obter esse tipo de resultado o banco de dados necessita de muito tempo de processamento, que acaba sendo desperdiçado, pois a base de dados descarta mais de 95% dos registros para retornar apenas os dados relacionados aos amigos de amigos de um indivíduo até a profundidade 4.

Na estrutura utilizada na base de dados orientada a grafos, em vez de tabelas, colunas e chaves estrangeiras, utiliza-se um modelo de dados com nós, relacionamentos, propriedades e índices. No contexto deste trabalho, os usuários do Facebook são os nós do grafo e as suas informações são as propriedades desse nó, as relações de amizade estabelecidas na rede social são os relacionamentos entre os nós e os índices são as chaves únicas utilizadas para identificar esses dados.

Para realizar as consultas de dados, o banco de dados utiliza as travessias no grafo, um conceito matemático retirado da teoria de grafos. Ao realizar a travessia, a base de dados acessa um conjunto de dados movendo-se entre os nós através das arestas formadas pelos seus relacionamentos, dessa forma, a travessia só considera os dados necessários para a consulta, sem a necessidade de envolver o conjunto inteiro de dados.

No grafo do estudo de caso, a travessia é iniciada em um nó chamado de nó de referencia, e percorre seu caminho através dos nós conectados pela relação de amizade, armazenando somente os nós visitados durante o caminho traçado. Da mesma forma que o modelo relacional, a consulta possui regras aplicadas a consulta de dados, portanto assim que a travessia não encontrar dados conectados onde essas regras sejam aplicadas, a travessia irá parar.

No Quadro 4 podemos visualizar os resultados obtidos nos testes de desempenho realizados na base de dados orientada a grafos, nos quais foram utilizados diferentes profundidades e quantidades de dados.

Quadro 4 - Testes Base de Dados Orientada a Grafos

Profundidade	10095 Usuários	20349 Usuários	30522 Usuários	40053 Usuários	57673 Usuários
2	7ms	8ms	8ms	8ms	11ms
3	8ms	9ms	16ms	25ms	38ms
4	9ms	15ms	26ms	39ms	62ms

Fonte: O Autor (2013)

Através do quadro de resultados apresentado, é possível identificar que o desempenho na consulta de dados de profundidade 2, o banco de dados orientado a grafos é semelhante ao relacional. Mas quando a profundidade e a quantidade de dados envolvida na consulta aumenta, a base de dados orientada a grafos apresenta um desempenho significativamente melhor que o modelo relacional.

Ao contrário da base de dados relacional, na qual o desempenho das consultas de grande profundidade é prejudicado devido às operações de produto cartesiano, a base de dados orientada a grafos possui o controle dos nós acessados através do caminho percorrido, carregando somente os dados desejados.

A fim de simular um ambiente com um volume maior de dados, foram criadas duas bases de dados com registros fictícios aplicando a mesma estrutura de uma rede social demonstrada anteriormente neste capítulo. As estruturas geradas possuem 1 milhão de registros de pessoas e cada pessoa possui em torno de 100 relacionamentos de amizade.

No Quadro 5 pode-se visualizar os resultados obtidos nos testes de desempenho realizados na base de dados relacional, nos quais foram utilizados diferentes profundidades.

Quadro 5- Teste Base de Dados Relacional com um milhão de dados

Profundidade	Tempo de Execução
2	5586ms
3	8550ms
4	46818ms
5	638635ms

Fonte: O Autor (2013)

Aplicar os testes de desempenho utilizando uma maior massa de dados na base de dados relacional se mostrou uma tarefa demorada. Pois pesquisas que envolvem relacionamentos recursivos, acabam se tornando computacionalmente complexas devido à estrutura de organização do modelo relacional.

Com 1 milhão de indivíduos cadastrados, o produto cartesiano gerado na consulta de profundidade 5 possui 10^{30} registros. Filtrar todos os registros da consulta para verificar quais se encaixam nos requisitos da pesquisa é um desperdício de tempo e desempenho.

O desempenho é uma das principais preocupações dos engenheiros de software em relação aos seus sistemas. Portanto, sistemas web e aplicativos online como as redes sociais, devem responder rapidamente para se tornarem um sucesso comercial.

No Quadro 6 pode-se visualizar os resultados obtidos nos testes de desempenho realizados na base de orientada a grafos, nos quais foram utilizados diferentes profundidades.

Quadro 6 - Teste Base de Dados Orientada a Grafos com um milhão de dados

Profundidade	Tempo de Execução
2	16ms
3	328ms
4	2497ms
5	11218ms

Fonte: O Autor (2013)

Através dos resultados apresentados no Quadro 6, é possível identificar que o desempenho na consulta com 1 milhão de indivíduos na profundidade 2 é semelhante ao resultado obtido nos testes iniciais com 57673 indivíduos na mesma profundidade.

A análise de grandes volumes de dados é complexa utilizando o modelo relacional devido ao alto nível de processamento necessário, mas se torna mais simples em um esquema de dados que utiliza grafos, pois as travessias utilizadas ao realizar as consultas foram estruturadas baseando-se em conceitos matemáticos da teoria de grafos, tornando a pesquisa de dados mais eficientes quando se trabalha com redes de dados. Além disso, a estrutura livre de índice facilita consultas de alto desempenho e é um aspecto importante no design e na maneira que os dados são armazenados.

Independentemente do número de nós e relacionamentos existentes no grafo, a travessia só irá acessar os nós que são relevantes para a consulta. Quanto maior profundidade estabelecida na consulta, maior será a quantidade de nós que a travessia precisa visitar, tornando a consulta mais lenta. No entanto, este aumento de tempo é linear independente do tamanho total do grafo.

As consultas foram realizadas com cache quente, ou seja, cada consulta foi repetida 10 vezes e o menor resultado obtido foi apresentado.

Os testes de desempenho apresentados foram realizados em uma máquina com o processador AMD FX 6100 de 3.3 GHz com 6 núcleos, 8 GB de memória RAM DDR3 1600

Dual Chanel da Corsair, fonte CX600 de 600 watts reais da Corsair, placa mãe ASUS M5A97 PRO e placa de vídeo XFX Radeon HD 7850 Core Edition 2048MB DDR5.

5.5.2 Cases

As bases de dados orientadas a grafo podem ser utilizadas em diversos contextos e situações diferentes. Com o intuito de demonstrar outras possibilidades de uso desse tipo de estrutura na área da análise social, a seguir, serão apresentados alguns casos de uso.

5.5.2.1 Busca de Menor Caminho

O problema de menor caminho é muito conhecido no meio matemático e computacional. Para Vasudev (2006), ele consiste em encontrar o caminho de menor custo entre um nó de origem e um nó de destino em um grafo.

Existem muitos algoritmos para encontrar o caminho mais curto em um grafo ponderado. Um algoritmo utilizado para realizar busca em grafos e estruturas do tipo árvore é a Busca em Largura, no qual se estabelece um vértice de origem e percorrem-se todos os vértices vizinhos de uma profundidade x. Após todos os vértices de profundidade x serem percorridos, o algoritmo passa para os vértices de profundidade $x+1$, e assim por diante. O algoritmo da Busca em Largura tem o objetivo de calcular a distância (menor número de arestas) desde o vértice (raiz) até todos os vértices acessíveis.

O algoritmo de Busca em Largura serviu como base para um algoritmo muito conhecido desenvolvido por Dijkstra e publicado em 1956. Esse algoritmo tem por objetivo encontrar o caminho mais curto em um grafo com peso não negativo associado em suas arestas, enumerando explicitamente todos os caminhos possíveis. Este algoritmo baseia-se em uma técnica conhecida como programação dinâmica.

Devido às buscas e travessias no grafo serem princípios básicos da teoria de grafos, os algoritmos de Busca em Largura, Dijkstra e A* foram agregados a API do Neo4J possibilitando a utilização dos mesmos na busca de menor caminho entre dois nós.

Em bases de dados como a do estudo de caso abordado neste trabalho, na qual não existe um custo definido nas arestas do grafo, é indicado que a Busca em Largura seja utilizada.

Abaixo segue um exemplo de Busca de Menor Caminho no grafo formado pela relação de amizade entre os usuários do Facebook utilizado neste trabalho através do Neo4J.

```
START inicio = node(9), destino = node(15373)
MATCH p = allShortestPaths (inicio-[r:AMIGO*]->destino)
WHERE inicio <> destino  AND LENGTH(NODES(p)) > 2
RETURN EXTRACT(n IN NODES(p): n.first_name);
```

Na consulta acima, inicialmente são definidos um nó de inicio e um nó de destino para a busca no grafo, nós 9 e 15373. Então é utilizado o método allShortestPaths da API do Neo4J, que utiliza a Busca em Largura otimizada para localizar os menores caminhos até o nó de destino levando em consideração a aresta estabelecida através do relacionamento AMIGO.

Figura 32 - Consulta Menor Caminho Neo4J

```
neo4j-sh (?)$ START inicio = node(9), destino = node(15373)  MATCH p = allShortestPaths(inicio-[r:AMIGO*]-
>destino)  WHERE inicio <> destino  AND LENGTH(NODES(p)) > 2 RETURN EXTRACT(n IN NODES(p): n.first_name);
==> +------------------------------------------------------------+
==> | EXTRACT(n IN NODES(p): n.first_name)                       |
==> +------------------------------------------------------------+
==> | ["Guilherme","Larissa","Maria Inés","Cris","Rafael Angelo"]  |
==> | ["Guilherme","Larissa","Fernanda","Cris","Rafael Angelo"]    |
==> | ["Guilherme","Flávio","Maria Inés","Cris","Rafael Angelo"]   |
==> | ["Guilherme","Flávio","Mauro","Cris","Rafael Angelo"]        |
==> | ["Guilherme","Flávio","Fernanda","Cris","Rafael Angelo"]     |
==> | ["Guilherme","Graziela","Maria Inés","Cris","Rafael Angelo"] |
==> | ["Guilherme","Denis","Maria Inés","Cris","Rafael Angelo"]    |
==> | ["Guilherme","Denis","Fernanda","Cris","Rafael Angelo"]      |
==> | ["Guilherme","André","Fernanda","Cris","Rafael Angelo"]      |
==> +------------------------------------------------------------+
==> 9 rows
==>
==> 108 ms
neo4j-sh (?)$
```

Fonte: Elaboração do autor, 2013.

A Figura 32 ilustra a consulta aplicada no console do Neo4J e os resultados da mesma. Nela podem-se verificar todos os menores caminhos percorridos até o nó estabelecido como destino.

O resultado da consulta apresentado na Figura 30, demonstra que uma consulta que necessitaria de um grande poder computacional e a utilização de diversas operações de

junção ou Common Table Expression em uma base de dados relacional, pode ser resolvida em uma consulta de poucas linhas em uma base orientada a grafos.

5.5.2.2 Sugestão de Novos Grupos de Indivíduos

Desde a antiguidade os indivíduos se organizam em grupos, comunidades e associações, baseando-se interesses em comum. Na área da sociologia, a teoria dos grafos serve como base para o estudo das redes formadas, e a análise dessas características similares entre indivíduos e suas relações sociais, no mundo virtual e concreto, é um dos pontos estudados utilizando os grafos.

As teorias de redes são usadas para estabelecer uma ponte nas explicações de diferentes fenômenos na sociedade e na natureza. O estudo dessas redes pode ser visto como parte de uma abordagem da teoria de complexidade. As redes complexas não são apenas um conceito usado com frequência, elas estão distribuídas fisicamente na nossa sociedade. (SCHARNOHORST, 2003, tradução nossa).

Dentro deste contexto, as propriedades estruturais e dinâmicas de redes, são diretamente ligadas ao estudo da matemática. Porém a análise das redes pode envolver várias áreas de conhecimento.

Nos últimos anos, a automatização da aquisição de dados e a disponibilidade de um elevado poder computacional têm levado ao surgimento de grandes bases de dados de topologia complexa de várias redes reais. A disponibilidade desta enorme quantidade de dados reais impulsionou o estudo e um grande interesse em descobrir as propriedades e particularidades dessas redes. (WANG, 2002, tradução nossa).

A estrutura livre de esquema das bases de dados orientadas a grafo fornece uma estrutura favorável para a utilização das mesmas na análise de redes. Devido a essa maleabilidade, os atores sociais podem ser facilmente interconectados através de diversas relações formando clusters interligados.

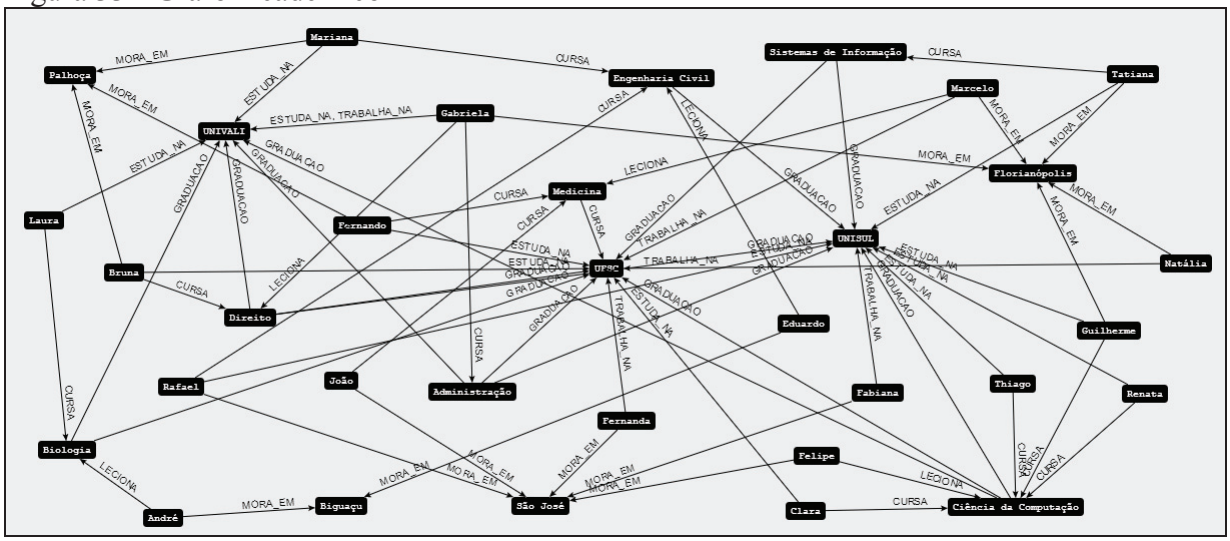
As ações dos indivíduos estão diretamente vinculadas às relações e ligações dos mesmos nas redes. Essas ações acabam definindo o comportamento do individuo diante da comunidade a que pertence e a quais elementos da rede esse indivíduo está conectado.

Análises de grupos devem ir além dos atributos individuais e considerar as relações entre os indivíduos, focando nos laços sociais, densidade da rede, multiplexidade, clusterização e composição do laço social.

A análise estatística dessas redes formadas concentra-se no resultado de centenas de milhares de interações no nível macro do sistema. Embora o comportamento individual no nível micro seja responsável por aquilo que pode ser mensurado no nível macro. (SCHARNOHORST, 2003, tradução nossa).

Abaixo segue a estrutura de um grafo social, no qual podemos identificar os diversos tipos de relações atores envolvidos.

Figura 33 - Grafo Acadêmico



Fonte: O Autor (2013)

A Figura 33 apresenta um grafo desenvolvido no Neo4J que registra a interação entre pessoas, universidades, cidades e cursos de graduação. Cada nó do grafo possui uma propriedade “nome” onde o nome do mesmo está registrado e eles se conectam através dos relacionamentos ESTUDA_NA, TRABALHA_NA, CURSA, GRADUACAO, LECIONA e MORA_EM.

Devido ao fato da relação entre entidades ser definida através de um relacionamento no grafo e não através de tabelas e o uso de chaves estrangeiras, a interação entre essas entidades é organizada de forma intuitiva e próxima ao que ocorre na vida real.

Da mesma forma que ocorre na nossa sociedade, os atores sociais ilustrados no grafo possuem características em comum, como o local de estudo, local de trabalho, local de moradia e graduação cursada.

A Figura 34 ilustra a consulta realizada na base de dados orientada a grafos para localizar os indivíduos que cursam Ciência da Computação.

Figura 34 - Consulta por Graduação

```
neo4j-sh (?)$ START pessoa = node(*) MATCH pessoa-[:CURSA]-graduacao WHERE (graduacao.Nome = "Ciência da Computação") return pessoa;
==> +--------------------------+
==> | pessoa                   |
==> +--------------------------+
==> | Node[7]{Nome:"Guilherme"} |
==> | Node[35]{Nome:"Thiago"}   |
==> | Node[36]{Nome:"Renata"}   |
==> | Node[37]{Nome:"Clara"}    |
==> +--------------------------+
==> 4 rows
```

Fonte: O Autor (2013)

Através do resultado obtido na consulta verifica-se que Thiago, Renata, Clara e Guilherme cursam o mesmo curso de graduação, Ciência da Computação.

Outras cláusulas podem ser adicionadas a consulta realizada, tornando a pesquisa ainda mais refinada. A Figura 35 ilustra a consulta realizada para localizar os indivíduos que cursam Ciência da Computação na UNISUL.

Figura 35 - Consulta por Graduação e Universidade

```
neo4j-sh (?)$ START pessoa = node(*)  MATCH pessoa-[:CURSA]-graduacao-[:GRADUACAO]-universidade<-[:ESTUDA_NA]-
pessoa WHERE (graduacao.Nome = "Ciência da Computação") and (universidade.Nome = "UNISUL")  return pessoa;
==> +--------------------------+
==> | pessoa                   |
==> +--------------------------+
==> | Node[7]{Nome:"Guilherme"} |
==> | Node[35]{Nome:"Thiago"}   |
==> | Node[36]{Nome:"Renata"}   |
==> +--------------------------+
==> 3 rows
```

Fonte: O Autor (2013)

Através do resultado da consulta verifica-se que somente Thiago, Renata e Guilherme cursam Ciência da Computação na UNISUL.

Dessa forma, características e interesses em comum podem ser utilizados para a formação de novos grupos, associações e comunidades de indivíduos. Dentro do contexto acadêmico apresentado no grafo, sugestões de amizade, formação de grupos de estudo, comunidades de pesquisa, comunidades de alunos e professores de uma universidade entre outros, podem ser apresentadas levando em consideração os interesses em comum e as relações apresentadas no grafo.

5.5.2.3 Sugestão de Novos Amigos

De acordo com Aggarwal (2011, tradução nossa), identificar os nós mais próximos em um grafo é uma peça chave em um conjunto diversificado de aplicações, desde busca e sugestão de amigos em uma rede social, marketing viral na web, busca e análise inteligente de palavras chave em uma base de dados.

Redes sociais, como o Facebook, apresentam sugestões de amizade aos usuários. Essas sugestões são feitas levando em consideração as relações em comum entre esses usuários, seus amigos e outros usuários que são amigos de seus amigos até um determinado grau de profundidade.

A Figura 36 ilustra uma consulta realizada na base de dados do Neo4J para identificar possíveis amigos em uma rede social.

Figura 36 - Consulta para Sugestão de Amigos

```
neo4j-sh (?)$ START inicio=node(1102) MATCH inicio-[:AMIGO]->amigo-[:AMIGO]->amigo_do_amigo WHERE NOT (inicio-[:AMIGO]-
amigo_do_amigo) AND amigo_do_amigo<>inicio RETURN amigo_do_amigo.id, amigo_do_amigo.first_name, COUNT(*) AS amigos_em_comum ORDER BY amigos_em_comum DESC;
==> +-------------------------------------------------------------------------------+
==> | amigo_do_amigo.id | amigo_do_amigo.first_name                | amigos_em_comum |
==> +-------------------------------------------------------------------------------+
==> | 1523568773        | "Saulo"                                  | 7               |
==> | 1478192142        | "Giovane"                                | 7               |
==> | 100002364948880   | "Thárcio"                                | 6               |
==> | 100002175442825   | "Marcelo"                                | 6               |
==> | 100001224812087   | "Lucas"                                  | 6               |
==> | 906400009         | "Bernardo"                               | 6               |
==> | 100000377564373   | "Guilherme Cesar"                        | 6               |
==> | 100001685210927   | "Eduardo"                                | 5               |
==> | 1825852029        | "Ricardo"                                | 5               |
==> | 100003538901572   | "Fernanda"                               | 5               |
==> | 1241320239        | "Maria Inés"                             | 5               |
==> | 100002310723977   | "Renata"                                 | 5               |
==> | 100001239510454   | "Ricardo"                                | 5               |
==> | 1584712310        | "Angela"                                 | 5               |
==> | 100002289718551   | "Antônio Luiz"                           | 5               |
==> | 100000114623640   | "Débora"                                 | 5               |
==> | 100001702956684   | "Roberto"                                | 5               |
==> | 100002184521605   | "Heliton"                                | 5               |
==> | 734333847         | "Ian"                                    | 4               |
==> | 100001436421095   | "Gustavo"                                | 4               |
==> | 1101694972        | "Gabriel"                                | 4               |
==> | 1104793434        | "Alessandro"                             | 4               |
==> | 100002407975256   | "Laura"                                  | 4               |
==> | 100000256278277   | "Matheus"                                | 4               |
==> | 100000069956883   | "Mazinha"                                | 4               |
==> | 1797125197        | "Maristéla"                              | 4               |
==> | 100002149847690   | "Alexandre"                              | 4               |
==> | 100002620502489   | "Andreia"                                | 4               |
==> | 516708910         | "Kaio"                                   | 4               |
==> | 100001374411098   | "Fernanda"                               | 4               |
==> | 100003124291241   | "Marcello"                               | 4               |
==> | 100001034642733   | "Giana"                                  | 4               |
```

Fonte: O Autor (2013)

A consulta ilustrada na Figura 34 segue o mesmo padrão da utilizada nos testes de desempenho apresentados neste trabalho. Porém ao invés de somente contabilizar o número total de amigos de amigos em uma determinada profundidade, ela apresenta os dados do possível amigo e a quantidade de amigos em comum que o usuário e esse indivíduo analisado possuem.

Conforme pode ser visto nos testes da seção 5.4.1 deste trabalho, bases de dados orientadas a grafo apresentam uma vantagem de desempenho para esse tipo de pesquisa, pois as travessias realizadas no grafo são mais eficientes que o motor de consulta das bases relacionais para pesquisas recursivas.

5.5.2.4 Estudo da Estrutura de um Grafo

Em uma rede social de amizades é possível que os amigos de seus amigos sejam seus amigos diretos, ou então, dois de seus amigos sejam amigos um do outro. Esta propriedade é chamada de clusterização de redes. (WANG, 2002, tradução nossa).

O coeficiente de clusterização de um vértice selecionado em uma rede é definido como a probabilidade de que dois vizinhos desse vértice selecionados aleatoriamente sejam ligados uns aos outros.

Supondo que um vértice *n* de uma rede possua *y* arestas, que ligam a *y* outros vértices, só podem existir no máximo *y(y - 1)/2* arestas entre esses vértices. Portando para obter o coeficiente de clusterização do vértice *n*, deve-se é calcular a razão entre o número de arestas que existem entre os vértices adjacentes ao vértice *n* e o total de arestas possíveis entre eles. Para obter o coeficiente de clusterização de toda a rede, deve-se calcular a média do coeficiente de clusterização de todos os vértices da rede.

Um grafo gerado aleatoriamente geralmente não apresenta clusterização. Pois em um grafo gerado aleatoriamente a probabilidade de dois vértices adjacentes a um vértice x estarem conectados um ao outro através de uma aresta *y*, não é maior do que a probabilidade de dois vértices escolhidos aleatoriamente adjacentes. (NEWMAN, 2000, tradução nossa).

A Figura 37 apresenta a consulta realizada para identificar o número de vértices adjacentes a um vértice especifico e o número de arestas existentes entre esses vértices adjacentes.

Figura 37 - Coeficiente de Clusterização

```
neo4j-sh (?)$ START inicio = node(1102)  MATCH (inicio)--(b) WITH inicio, count(DISTINCT b) AS y MATCH (inicio)--()-[r]-()--
(inicio) RETURN y, count(DISTINCT r) AS r;
==> +-----------+
==> | y   | r   |
==> +-----------+
==> | 339 | 691 |
==> +-----------+
==> 1 row
==>
==> 8200 ms
```

Fonte: O Autor (2013)

Com o número de vértices adjacentes e o número de ligações recíprocas entre esses vértices pode-se calcular o coeficiente da seguinte maneira:

O número de possíveis ligações entre os dois vizinhos é $y(y - 1)/2 = 339(339 -1)/2 = 57291$, onde y é o número de vértices adjacentes $y = 339$, e o número de aresta existentes é de conexões r é 691. Portanto, o coeficiente de clusterização do nó n é de 691/57291 = 0,012.

5.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo teve como objetivo apresentar o histórico de desenvolvimento deste trabalho, as ferramentas tecnológicas utilizadas e infraestrutura criada para implementar o estudo de caso proposto. Além disso, foram expostos os resultados obtidos nos testes comparativos realizados nas bases de dados relacional e orientada a grafos.

Com o propósito de demonstrar a versatilidade do modelo de dados orientado a grafo na análise de redes complexas, foram apresentados casos de uso utilizando o contexto de análise social.

6 CONCLUSÕES E TRABALHOS FUTUROS

Nesse capítulo são abordadas as conclusões relacionadas aos resultados obtidos com o desenvolvimento deste trabalho e os possíveis trabalhos futuros. Os problemas encontrados durante o desenvolvimento do estudo de caso, objetivos alcançados e não alcançados também serão citados.

6.1 CONCLUSÕES

Este trabalho apresentou conceitos sobre bancos de dados relacionais e orientados a grafo, tendo como objetivo principal a modelagem e implementação uma base de dados orientada a grafos, a fim de identificar suas diferenças em relação à abordagem relacional. Outro objetivo deste trabalho era realizar um estudo de caso sobre análise social a fim de testar o desempenho dos modelos relacional e orientado a grafos ao realizarem consultas que envolvem cálculos relacionais e relacionamentos recursivos.

Durante os últimos anos, a quantidade de informações acumuladas em grandes bancos de dados, como o do Facebook, se tornou um impeditivo para o alto desempenho dos sistemas. Essas massas de dados, Big Data, se tornaram muito grande para serem manipuladas e analisadas através modelos tradicionais de banco de dados, como o relacional.

As bases de dados NOSQL (Not Only SQL), tem se tornado cada vez mais populares, pois são altamente escaláveis e conseguem operar com gigantescas cargas de dados. Grandes corporações como o Facebook, Google, Twitter e Amazon, utilizam bases de dados NOSQL para armazenar e gerenciar seus dados. A base de dados orientada a grafos utilizada neste trabalho é um exemplo desse tipo de modelagem.

A fim de avaliar as diferenças e benefícios de se utilizar uma base de dados NOSQL, este trabalho apresentou um estudo de caso baseado em análise social, no qual foram implementadas uma base de dados orientada a grafos e uma seguindo o modelo relacional.

A rede de dados foi construída utilizando de informações fornecidas por usuários da rede social Facebook através de uma aplicação construída para facilitar a conexão com o Facebook e captura dos dados dos usuários dispostos a colaborar com este projeto.

Os testes iniciais de desempenho foram realizados baseando-se nas relações entre os indivíduos da rede, consultando os dados de amigos de amigos até a profundidade 4, tendo o autor deste trabalho como vértice inicial da pesquisa.

Ao analisar os resultados obtidos nos testes é possível identificar a diferença de desempenho entre o modelo relacional e o orientado a grafos em consultas recursivas, visto que as consultas realizadas na base de dados orientada a grafos apresentaram um desempenho superior ao apresentado pelo modelo relacional de dados.

Ao simular o contexto de *Big Data* para verificar a performance das bases de dados frente a um volume maior de dados, a diferença de desempenho se mostra ainda maior. Portanto, pode-se concluir que utilizar bases de dados orientadas a grafo para realizar pesquisas em redes complexas de dados ou em *Big Data*, é mais vantajoso quando se busca soluções de alto desempenho e escalabilidade.

A estrutura dos grafos facilita a modelagem dos dados, pois não há necessidade de reestruturar o esquema de dados cada vez que um novo tipo de entidade ou relacionamento é adicionado. E ainda, a natureza livre de índice melhora a persistência de dados e as consultas de alto desempenho na base de dados.

O modelo de dados orientado a grafos se mostrou muito útil para modelagem, armazenamento e análise de redes complexas de dados como as utilizadas na análise social. Pois a forma que os dados são modelados nas bases orientadas a grafo é muito próxima à maneira que os mesmos estão estruturados e conectados na vida real, tornando a modelagem em grafos versátil, simples e eficiente. Como exemplo pode-se citar as consultas de busca de menor caminho, as quais normalmente são escritas utilizando várias linhas de código no paradigma relacional, mas podem ser realizadas utilizando poucas linhas de código no modelo orientado a grafo.

Após analisar a problemática apresentada neste trabalho, a proposta de solução, e os resultados obtidos através dos testes de desempenho, conclui-se que a proposta de utilização de uma base orientada a grafos para análise social é válida, podendo também ser utilizada em outros cenários e não apenas no apresentado no estudo de caso.

Por se tratar de um paradigma relativamente novo e menos utilizado, encontrar materiais científicos e ferramentas de apoio relacionadas à base de dados orientada a grafos, se tornou um problema no desenvolvimento deste trabalho. Além disso, o modelo e aplicação de bancos de dados orientados a grafos eram novos para o autor deste trabalho, pois o mesmo não havia tido a oportunidade de trabalhar com esse tipo de estrutura de dados. Também foram encontradas dificuldades ao tentar representar graficamente os grafos criados, pois

inúmeras ferramentas gratuitas disponíveis para esse fim apresentaram erros devido ao tamanho e complexidade da rede de dados.

A elaboração deste trabalho proporcionou uma oportunidade única de desenvolvimento técnico e pessoal ao autor do mesmo. Devido aos desafios e dificuldades impostas durante este projeto, a revisão de velhos conceitos assimilados e a busca por novos conhecimentos necessários foi constante durante todo o projeto. Também é valido citar a colaboração de familiares, colegas, professores, amigos e outros usuários do Facebook que mesmo não conhecendo o autor deste trabalho, colaboraram fornecendo seus dados de perfil e lista de amigos para serem utilizados na elaboração da rede utilizada no estudo de caso e divulgaram o web site criado para a captura de dados.

6.2 TRABALHOS FUTUROS

Tendo em vista possíveis trabalhos futuros e algumas dificuldades encontradas durante o desenvolvimento deste trabalho, pode-se citar o desenvolvimento de uma interface interativa para a visualização e manipulação de grafos complexos e de grande porte.

Esse tipo de interface facilitaria as análises estruturais das redes de dados e a permitiria rápida identificação de clusters e ilhas de conhecimento em uma rede.

Durante o desenvolvimento deste trabalho, o autor do mesmo percebeu que os algoritmos utilizados para organizar e estruturar as redes complexas de dados utilizam altos recursos de hardware para processar essas informações. Logo, pode ser desenvolvido um algoritmo de alto desempenho que não utilize tanto poder computacional para a estruturação e representação de redes complexas de dados.

REFERÊNCIAS

ABRAHAM, Ajith; HASSANIEN, Aboul-Ella; SNÁSEL, Vaclav. **Computational Social Network Analysis: Trends, Tools and Research Advances.** Londres: Springer, 2009.

AGGARWAL, Charu. C.; WANG, H. **Managing and Mining Graph Data**. Nova York: Springer, 2010.

AGGARWAL, Charu C. **Social Network Data Analytics**. Nova York: Springer, 2011.

ANDRADE, Maria Margarida de. **Introdução à metodologia do trabalho científico:** elaboração de trabalhos na graduação. 5 ed. São Paulo: Atlas, 2001.

ANGLES, Renzo. **A comparison of current graph database models**. In: 28th IEEE International Conference on Data Engineering Workshops, 2012, Washington. p. 171 - 177. Disponível em: < http://planetlab2.iiitb.ac.in/data/272_gdm2012.pdf>. Acesso em: 25 de maio 2013.

ANGLES, Renzo; GUTIERREZ, Claudio. **Survey of Graph Database Models**. ACM Computing Surveys, 2008.

BARBIERI, Carlos. **Modelagem de dados**. Rio de Janeiro. IBPI Press, 1994.

BATRA, Shalini; TYAGI, Charu. **Comparative analysis of relational and graph databases**. International Journal of Soft Computing, vol 2, 2012.

BEN-GAN, Itzik. **Microsoft SQL Server 2012 - T-SQL Fundamentals**. Sebastopol: O'Reilly Media Inc., 2012.

BEZERRA, Eduardo. **Princípios de analise e projeto de sistemas com UML**. Rio de Janeiro: Campus, 2002.

BOAVENTURA NETTO, P. O., 2003. **Grafos – Teoria, Modelos**, Algoritmos. 2ª ed. São Paulo: Ed. Edgard Blucher Ltda, 2001.

BOOCH, James; RUMBAUGH, Grady; JACOBSON, Ivar. UML: guia do usuário. Rio de Janeiro: Campus, 2000.

BOYD, Danah; CRAWFORD, Kate. 2011. **Six provocations for big data.** IN: A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society, Oxford Internet Institute, 2011, Oxford. Disponível em: <http://papers.ssrn.com/sol3/papers.cfm?abstract_id= 1926431>. Acesso em: 01 maio 2013

BRAGA, Mauro Joaquim da Costa; GOMES, Luiz Flavio Autran Monteiro; RUEDIGER, Marco Aurélio. **Mundos pequenos, produção acadêmica e grafos de colaboração: um estudo de caso dos Enanpads**. Rev. Adm. Pública [online]. 2008, vol.42, n.1, pp. 133-154.

CAMPOS, M. L. M., BORGES, V. J. A. S. **Diretrizes para a Modelagem Incremental de Data Marts**. Anais do XVII Simpósio Brasileiro de Bancos de Dados, Gramado, Brasil, 2002.

CECI, Flávio. **Business intelligence** : livro digital / Flávio Ceci ; design instrucional Silvana Souza da Cruz Clasen ; João Marcos de Souza Alves. – Palhoça : UnisulVirtual, 2012.

CHARTRAND, Gary; ZHANG, Ping. **Chromatic Graph Theory**. Boca Raton: Chapman & Hall Books, 2009.

CLEVE, Anthony; MENS, Tom; HAINAUT, Jean; **Data-Intensive System Evolution**. Agosto 2010, p. 110-112.

COOK, Diane. J.; HOLDER, Lawrence. B**. Mining Graph Data**. Nova Jersey: John Wiley & Sons, Inc., 2007

CORRIGAN, David; DEROOS, Dirk; DEUTSCH, Tom; ZIKOPOULOS, Paul; PARASURAMAN, Krishnan; GILES, James. **Harness the Power of Big Data.** McGraw-Hill Companies. 2013.

CROCKFORD, Douglas. **The application/json Media Type for JavaScript Object Notation (JSON)**. Internet Informational RFC 4627, Julho 2006. Disponível em: <http://www.ietf.org/rfc/rfc4627.txt>. Acesso em: 01 outubro 2013.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. Tradução da 7ª Edição Americana. Rio de Janeiro: Campus, 2000.

DAVIS, Kord; PATTERSON, Doug. **Ethics of Big Data: Balancing Risk and Innovation**. Sebastopol: O'Reilly Media Inc., 2012.

DEITEL, H. M.; DEITEL, P. J. **Java: Como Programar**. 4ª edição. Porto Alegre: Bookman, 2003.

EATON, Chris; DEROOS, Dirk; DEUTSCH, Tom; LAPIS, George; ZIKOPOULOS, Paul. **Understanding Big Data**.: McGraw-Hill Companies, 2012.

ENGHOLM, Hélio Jr**. Engenharia de software na prática**. 1ª edição. São Paulo: Novatec, 2010.

FEOFILOFF, P; KOHAYAKAWA, Y; WAKABAYSHI, Y. **Uma introdução sucinta a teoria dos grafos.** São Paulo, 2011. Disponível em:< http://www.ime.usp.br/~pf/ teoriadosgrafos/texto/TeoriaDosGrafos.pdf> Acesso em: 01 maio 2013

FOGGIA, P; SANSONE, C; VENTO, M**. A Database of Graphs for Isomorphism and Sub-Graph Isomorphism Benchmarking**, In: 3RD IAPR TC-15 INTERNATIONAL WORKSHOP ON GRAPH-BASED REPRESENTATIONS, 2001, Ischia. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.6803&rep=rep1&type=pdf >. Acesso em: 01 maio 2013

FORTULAN, Marcos Roberto; FILHO, Eduardo Vila Gonçalves. **Uma proposta de aplicação de Business Intelligence no chão-de-fábrica.** São Carlos, SP: 2005. Disponível em: < http://www.scielo.br/pdf/%0D/gp/v12n1/a06v12n1.pdf>. Acesso em: 08 maio 2013

FURLAN, José Davi. **Modelagem de objetos através da UML**. São Paulo: Makron Books, 1998.

GALLIANO, A. Guilherme. O Método Científico: Teoria e Prática. São Paulo: Harbra, 1979.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. São Paulo: Atlas, 1999.

GRAVES, Mark; BERGEMAN, Ellen R.; LAWRENCE, Charles B. Graph Database Systems for Genomics. **Engineering in Medicine and Biology**. vol. 14, nº 6, pp. 737–745, 1995. Disponível em: < http://www.mgmail1.com/people/mgraves/pubs/emb95.pdf>. Acesso em: 11 maio 2013

GOOGLE-GSON. **A Java library to convert JSON to Java objects and vice-versa**. Disponível em: < https://code.google.com/p/google-gson/> Acesso em: 06 outubro 2013.

GUIMARÃES, Célio Cardoso. **Fundamentos de bancos de dados. Modelagem, projeto e linguagem SQL**. Campinas (SP): Unicamp, 2003.

HERNANDES, Fábio. **Algoritmos para grafos com incertezas**. 2007. Tese (Doutorado) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 2007. Disponível em http://www.bibliotecadigital.unicamp.br/document/?down=vtls00 0412033> Acesso em: 11 maio 2013

HURWITZ, Judith; NUGENT, Alan; KAUFMAN, Marcia; HALPER, Fen. **Big Data for Dummies**. Wiley, 2013.

KAPLAN, Ian L; ABDULLA, Ghaleb M; BRUGGER, S Terry; KOHN, Scott R.; **Implementing Graph Pattern Queries on a Relational Database**. Lawrence Livermore National Laboratory, 2008. Disponível em: < http://www.bearcave.com/misl/misl _tech/Implementing_Graph_Pattern_Queries.pdf>. Acesso em: 01 maio 2013

KHOSHAFIAN, SETRAG. **Banco de Dados Orientado a Objetos**; traduzido por Tryte Informática. Rio de Janeiro: Infobook, 1994.

KIMBALL, Ralph; ROSS, Margy. **The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling**. Wiley, 2002.

LARMAN, Craig. **Utilizando UML e Padrões**: uma introdução à análise e ao projeto orientados a objetos. Porto Alegre: Bookman, 2000.

LOHR, Steve. The Age of Big Data. **The New York Times**, Nova York, 11 de fev. de 2012. Disponível em:<http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html?pagewanted=1&_r=2&sq=Big%20Data&st=cse&scp=1> Acesso em: 11 maio 2013

MICROSOFT. **Microsoft® SQL Server® 2008 Management Studio Express**. Publicado em: fev. 2009. Disponível em: < http://www.microsoft.com/pt-br/download/details.aspx ?id=7593 >. Acesso em: 01 de outubro de 2013

MILLER, Justin J; **Graph Database Applications and Concepts with Neo4j**. In: Southern Association for Information Systems Conference. 2013, Atlanta. p.141 – 147. Disponível em: < http://sais.aisnet.org/2013/MillerJ.pdf>. Acesso em: 05 maio 2013

NEO4J. **Graph Database**. Disponível em: <http://www.neo4j.org/>. Acesso em: 01 outubro 2013.

NETBEANS. **NetBeans IDE**. Disponível em: <https://netbeans.org/>. Acesso em: 01 outubro 2013.

NEWMAN, Mark E. J. **Models of Small World**. Journal of Statistical Physics, v. 101, p. 819-841, 2000. Disponível em: < http://arxiv.org/pdf/cond-mat/0001118.pdf>. Acesso em: 22 outubro 2013

OSTROTSKI, Alvaro; MENONCINI, Lucia. **Teoria dos Grafos e Aplicações**, In: 13º Encontro Regional de Matemática Aplicada e Computacional, 2009, Pato Branco. Disponível em: < http://revistas.utfpr.edu.br/pb/index.php/SysScy/article/view/709/465>. Acesso em: 01 maio 2013

PHP. **PHP: Hypertext Preprocessor**. Disponível em: < http://php.net>. Acesso em: 02 outubro 2013.

POWELL, Gavin. **Beginning Database Design**. San Francisco: Wiley Publishing, 2006.

RABUSKE, M.A. **Introdução à teoria dos grafos**. Florianópolis: UFSC, 1992.

RECUERO, Raquel. **Comunidades em Redes Sociais na Internet: Proposta de Tipologia baseada no Fotolog.com**. Tese de Doutorado. Porto Alegre. UFRGS. 2006. Disponível em : < http://www.bibliotecadigital.ufrgs.br/da.php?nrb=000582681&loc=2007&l=d5c1ded066871f30> Acesso em: 01 maio 2013

ROBINSON, Ian; WEBBER, Jim; EIFREM, Emil. **Graph Databases**. [S.l.]: O'Reilly Media, Inc. 2013. Disponível em : <http://info.neotechnology.com/rs/neotechnology/images/Graph Databases_EarlyRelease.pdf> Acesso em: 01 maio 2013

SCHARNOHORST, Andrea. **Complex Networks and the Web: Insights From Nonlinear Physics**. Journal of Computer Mediated Communication, V. 8, issue 4. Julho, 2003. Disponível em<http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2003.tb00222.x/full>. Acesso em 16/10/2013.

SELL, Denilson. **Uma arquitetura para business intelligence baseada em tecnologias semânticas para suporte a aplicações analíticas**. 2006. Tese (Doutorado) -Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia de Produção, Florianópolis, 2006.

SILBERSCHATZ, Abraham. ; KORTH, Henry.F. ; SUDARSHAN, S. **Sistema de Banco de Dados.** 3 ed. São Paulo: Makron Books, 1999.

SILVA, Dhiogo Cardoso da. **Uma arquitetura de business intelligence para processamento analítico baseado em tecnologias semânticas e em linguagem natural**. 2011. Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Florianópolis, 2011. Disponível em: < http://btd.egc.ufsc.br/wp-content/uploads/2011/04/DhiogoCardosoDaSilva.pdf> Acesso em: 09 de maio 2013

SILVA, E. L. da; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4. ed. Florianópolis: UFSC, 2005. 138 p. Disponível em: <http://www.convibra.com.br/upload/paper/adm/adm_3439.pdf>. Acesso em: 26 de maio 2013.

SMITH, M. Paula Marques; MARTINS, Paula Mendes. **Matemática discreta.** Braga: Universidade do Minho. Departamento de Matemática, 2009.

SOMMERVILLE, Ian. **Engenharia de Software**. 6ª ed. São Paulo: Pearson Addison-Wesley, 2003

SOUSA, Paulo Alexandre Morgado. **Efeito estruturante das redes de transporte no território modelo de análise**. 2010. 313 f. Tese (Doutorado) - Universidade de Lisboa, Lisboa 2010.

SZWARCFITER, Jayme Luiz. **Grafos e algoritmos computacionais**. Rio de Janeiro: Campus, 1984.

SZWARCFITER, Jayme; MARKENSON, **Lilian. Estrutura de Dados e seus algoritmos**. Rio de Janeiro : LTC – Livros Técnicos e Científicos, 1994.

VASUDEV, C. **Graph Theory with Applications**. Nova Deli: New Age International (P) Ltd., Publishers, 2006.

VICKNAIR, chad; MACIAS, Michael; ZHAO, Zhendong; NAN, Xiaofei; CHEN, Yixin; WILKINS, Dawn. **A comparison of a graph database and a relational database: A data provenance perspective**. In: 48th Annual Southeast Regional Conference, 2010, Nova York. p. 68–80. Disponível em: < http://www.cs.olemiss.edu/~ychen/publications/conference/vicknair_acmse10.pdf>. Acesso em: 25 de maio 2013.

VILA, Maria do Carmo. Ensino de matemática: uma proposta alternativa. **Educ. Rev**., Belo Horizonte, n. 02, dez. 1985 . Disponível em <http://educa.fcc.org.br/scielo.php?script=sci_arttext&pid=S0102-46981985000200010&lng=pt&nrm=iso>. acessos em 01 maio 2013

WANG, Xiao Fan. **Complex networks: Topology, dynamics and synchronization**. International Journal of Bifurcation and Chaos, V. 12, issue 5, p. 885–916. Maio, 2002. Disponível em< http://www.worldscientific.com/doi/pdf/10.1142/S0218127402004802>. Acesso em 22 outubro de 2013.

WASSERMAN, Stanley; FAUST, Katherine. **Social Network Analysis: Methods and Application**. Nova York: Cambridge University Press, 1994.

YAN, X; YU, P. S; HAN, J. **Graph indexing: a frequent structure-based approach**. In: 2004 ACM SIGMOD international conference on Management of data, 2004, Nova York, pp. 335–346, ACM Press. Disponível em:<http://www.cs.ucsb.edu/~xyan/papers/sigmod04_gindex.pdf >. Acesso em: 11 maio 2013

ZHAO, Peixiang; HAN, Jiawei. **On graph query optimization in large networks**. In: 36th INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 2010, Singapura. Disponível em:< http://www.vldb.org/pvldb/vldb2010/papers/R30.pdf>. Acesso em: 11 maio 2013