

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN
FACULDADE DE CIÊNCIAS NATURAIS E EXATAS – FANAT
DEPARTAMENTO DO INFORMÁTICA – DI

BRUNA MARIA NUNES BARROS

**MODELAGEM DE DADOS NOSQL: UMA MODELAGEM DE BANCO DE DADOS
DE GRAFOS PARA PERSISTÊNCIA DE DADOS DE PESQUISA – UM ESTUDO
DE CASO DO PROJETO “OFICINA MÍMESIS”**

Mossoró/RN

2017

BRUNA MARIA NUNES BARROS

**MODELAGEM DE DADOS NOSQL: UMA MODELAGEM DE BANCO DE DADOS
DE GRAFOS PARA PERSISTÊNCIA DE DADOS DE PESQUISA – UM ESTUDO
DE CASO DO PROJETO “OFICINA MÍMESIS”**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de Bacharel em Ciência da Computação, sob orientação da Prof^a. M.Sc. Ceres Germanna Braga Moraes e coorientação do Prof^o. Alysson Mendes de Oliveira.

Mossoró/RN

2017

Ficha catalográfica gerada pelo Sistema Integrado de Bibliotecas
e Diretoria de Informatização (DINF) - UERN,
com os dados fornecidos pelo(a) autor(a)

B277m Barros, Bruna Maria Nunes.
Modelagem de Dados NoSQL: Uma Modelagem de Banco de Dados
de Grafos para Persistência de Dados de Pesquisa – Um Estudo de
Caso do Projeto “Oficina Mimesis” / Bruna Maria Nunes Barros - 2017.
70 p.

Orientadora: Prof^a. M.Sc. Ceres Germanna Braga Morais.

Coorientador: Prof^o. Alysson Mendes de Oliveira.

Monografia (Graduação) - Universidade do Estado do Rio Grande do
Norte, Ciência da Computação, 2017.

1. Banco de Dados. 2. NoSQL. 3. Grafos. 4. Oficina Mimesis. 5.
Neo4j. I. Morais, Prof^a. M.Sc. Ceres Germanna Braga, orient. II.
Oliveira, Prof^o. Alysson Mendes de, co-orient. III. Título.

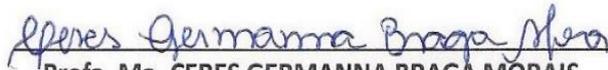
Bruna Maria Nunes Barros

MODELAGEM DE DADOS NOSQL: UMA MODELAGEM DE BANCO DE DADOS DE GRAFOS PARA PERSISTÊNCIA DE DADOS DE PESQUISA –UM ESTUDO DE CASO DO PROJETO “OFICINA MÍMESIS”

Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros.

Aprovada em: 26/04/2017

Banca Examinadora



Prof. Ma. CERES GERMANNA BRAGA MORAIS

Universidade do Estado do Rio Grande do Norte - UERN



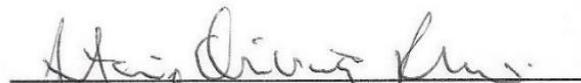
Prof. ALYSSON MENDES DE OLIVEIRA

Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. HENRIQUE JORGE AMORIM HOLANDA

Universidade do Estado do Rio Grande do Norte - UERN



Prof. Me. ANTÔNIO OLIVEIRA FILHO

Universidade do Estado do Rio Grande do Norte - UERN

AGRADECIMENTOS

Agradeço em primeiro lugar à Deus e sua infinita bondade para com esta pecadora que nada merece, e pelas forças que me renovou, permitindo que eu concluísse este ciclo iniciado há 6 anos atrás, à Ele toda honra e glória.

Aos meus pais pelo esforço e cuidado em ensinar o caminho da educação, nunca privando-me disto, agradeço por nunca terem desistido de mim.

Ao departamento de informática, técnicos, alunos e professores que de algum modo contribuíram para que eu estivesse aqui. Em especial ao professor Max Lopes por ter me incentivado a não desistir mesmo com as reprovações que por vezes me desmotivaram; ao professor Rommel Wladimir e ao Laboratório de Redes e Sistemas Distribuídos (LORDI), ao qual participei de um projeto de extensão nos primeiros anos da graduação; ao professor Dario José pela participação no projeto de iniciação Científica no Laboratório de Otimização e Inteligência Artificial (LOIA), e aos professores Ceres Germanna e Alysson Oliveira, por terem me orientado durante este trabalho de conclusão, e ao professor Willian Coelho do Departamento de Filosofia da UERN pelos dados disponibilizados para o desenvolvimento deste trabalho, à eles por todo conhecimento repassado.

Ao meu namorado, e também colega de graduação Michael Charles, por toda paciência, incentivo e disponibilidade em alguma atividade ou algo que eu precisasse aprender, por todo amor e força dada, principalmente nessa etapa final, e aos amigos que oraram ou estiveram presentes com palavras de ânimo, em especial: Mariza Moura, Valesca Juliane e Cecília Souza.

RESUMO

No cenário em que a sociedade encontra-se inserida, os volumes de dados apresentam-se cada vez mais extensos e complexos. Esses conjuntos de dados são associados ao termo *Big Data*, em que o desafio vai além do armazenamento, tratando-se da disponibilização de ferramentas e ambientes preparados para o suporte dos dados apresentados nesse contexto, o que caracteriza também os dados que já não conseguem ser processados por sistemas de banco de dados convencionais, devido à quantidade e velocidade em que os dados avançam. Diferente de quando surgiu o modelo relacional, os dados que agora são gerados na *web* são superiores às aplicações com dados estruturados. Tal cenário apresenta dados vindos de diferentes fontes e formatos, que são em grande maioria não estruturados e crescem em uma velocidade superior aos da década de 70. Por esses motivos, para lidar com as específicas aplicações que possuem tais dados surge a abordagem NoSQL, sendo uma alternativa, e não substituição, ao modelo relacional. Com o objetivo de apresentar essa nova alternativa o presente trabalho propõe modelar os dados do projeto Oficina Mímeses, desenvolvida no âmbito do curso de Filosofia, da Universidade do Estado do Rio Grande do Norte, utilizando a abordagem NoSQL, com a ferramenta Neo4j, que encaixa-se na categoria de bancos orientados à grafos.

Palavras-Chave: Banco de Dados; NoSQL; Grafos; Oficina Mímeses; Neo4j; Modelagem de dados.

ABSTRACT

In the scenario in which society is inserted, the data volumes are increasingly extensive and complex. These data sets are associated with the term Big Data, in which the challenge goes beyond storage, being the availability of tools and environments prepared to support the data presented in this context, which also characterizes the data that can no longer be processed by conventional database systems, because of the amount and speed at which data advances. Differently from when it came the relational model, the data that is now generated on the web is superior than applications with structured data. This scenario presents data coming from different sources and formats, which are in vast majority are not structured and increase at faster rate than the 1970s. For these reasons, to deal with the specific applications that have such data comes the NoSQL approach, being an alternative, not substitution, to the relational model. With the objective to present this new alternative, the present work proposes to model the data of Oficina Mímeses project, developed within the scope of philosophy course, of Universidade do Estado do Rio Grande do Norte, using the NoSQL approach, with the Neo4J tool, which suit in the category of Graph Oriented Databases.

Keywords: Database, NoSQL; Graph; Oficina Mímeses; Neo4J; data modeling.

LISTA DE SIGLAS

ACID	Atomicidade, Consistência, Isolamento, Durabilidade
ASCII	<i>American Standard Code for Information Interchange</i>
API	<i>Application Programming Interface</i>
BASE	<i>Basically Available, Soft State, Eventual consistency</i>
CAP	<i>Consistency, Availability, Partition Tolerance</i>
CRUD	<i>Create, Read, Update, Delete</i>
MDG-NoSQL	Modelo de Dados para Bancos NoSQL Baseados em Grafo
NoSQL	<i>Not Only SQL</i>
RFID	<i>Rádio Frequency Identification</i>
SGBDs	Sistema de Gerenciamento de Banco de Dados
SGBDG	Sistema Gerenciador de Bancos de Dados em Grafos (SGBDG)
SPARQL	<i>SPARQL Protocol and RDF Query Language</i>
SQL	<i>Structured Query Language</i>
UERN	Universidade do Estado do Rio Grande do Norte
XML	<i>eXtensible Markup Language</i>

LISTA DE FIGURAS

Figura 1: Teorema Cap	18
Figura 2: Modelo em grafos.....	24
Figura 3: Análise Comparativa Modelo Relacional x NoSQL	27
Figura 4: Modelo de dados usado para representar as entidades no modelo de recomendação.....	29
Figura 5: Quadro de correspondência de conceitos.....	31
Figura 6: Consulta de empresas que possuem o mesmo telefone.....	32
Figura 7: Resultado da consulta de telefones	33
Figura 8: Busca de empresas relacionadas por telefone utilizando o Cypher	33
Figura 9: Resultado da consulta para empresas por telefones no Neo4j	34
Figura 10: Pontos e linhas representando um grafo.....	35
Figura 11: Grafo representando rede social entre amigos	38
Figura 12: Exemplo de nó	39
Figura 13: Exemplo de relações.....	40
Figura 14: Exemplo criação de nós e relação em Cypher	41
Figura 15: Exemplo consulta match	41
Figura 16: Fluxo de trabalho (workflow)	45
Figura 17: Sessão de informações básicas e endereço do questionário	46
Figura 18: Elementos identificados na sessão informações básicas e endereço.....	47
Figura 19: Sessão de informações sobre a formação acadêmica graduação	47
Figura 20: Elementos identificados na sessão formação acadêmica – graduação ..	48
Figura 21: Sessão de informações sobre a formação acadêmica, especialização, mestrado e doutorado	48
Figura 22: Elementos identificados na sessão formação acadêmica, especialização, mestrado e doutorado	49
Figura 23: Sessão de informações profissionais	49
Figura 24: Elementos identificados na sessão dados profissionais.....	50
Figura 25: Sessão outras disciplinas.....	50
Figura 26: Elementos identificados na sessão outras disciplinas.....	51
Figura 27: Sobre as questões	51
Figura 28: Perguntas sobre a disciplina lecionada	52
Figura 29: Relação identificada na pergunta um	52
Figura 30: Relação identificada na pergunta dois	53
Figura 31: Relação identificada na pergunta três	53
Figura 32: Relação identificada na pergunta quatro.....	54
Figura 33: Relação identificada na pergunta cinco.....	54
Figura 34: Relação identificada na pergunta seis.....	54
Figura 35: Relação identificada na pergunta sete	55
Figura 36: Nó e relação identificada na pergunta oito	55
Figura 37: Relação identificada na pergunta nove	55
Figura 38: Modelagem completa.....	56
Figura 39: Consulta que retorna os nós que tenha a relação mora.....	57
Figura 40: Resposta da consulta em linhas	58
Figura 41: Consulta e resposta da relação mora e natural em linhas	58

Figura 42: Consulta e resposta da relação mora e natural em grafo.....	59
Figura 43: Consulta e resposta da relação formou-se em linhas	59
Figura 44: Consulta e resposta da relação formou-se em linhas	60
Figura 45: Consulta e resposta das pessoas que orientaram o nó Pedro.....	61
Figura 46: Consulta e resposta das pessoas que orientaram o nó Pedro em grafos	61
Figura 47: Consulta e resposta das relações entre o nó Pedro e o nó Filosofia	62
Figura 48: Consulta e resposta das relações entre o nó Pedro e o nó Escola.....	63

SUMÁRIO

INTRODUÇÃO	12
2. BANCOS DE DADOS NOSQL	15
2.1 ORIGEM.....	15
2.2 CONTEXTUALIZAÇÃO.....	15
2.3 CARACTERÍSTICAS DOS BANCOS NOSQL.....	16
2.4 RELACIONAL versus NOSQL.....	20
2.5 CLASSIFICAÇÃO DOS BANCOS NOSQL.....	21
2.5.1 Chave-valor	21
2.5.2 Colunas	22
2.5.3 Documentos	23
2.5.4 Grafos	23
3. TRABALHOS RELACIONADOS	26
3.1 SELEÇÃO DE UM SGBD NOSQL.....	26
3.2 MODELAGEM EM GRAFOS	28
3.3 MAPEAMENTOS ENTRE O MODELO RELACIONAL E O NOSQL	30
3.4 BANCOS ORIENTADOS A GRAFOS (NEO4J)	31
4. BANCOS DE DADOS ORIENTADOS A GRAFOS	35
4.1 DEFINIÇÃO DE GRAFOS.....	35
4.2 HISTÓRICO.....	36
4.3 MODELAGEM EM GRAFOS	36
4.4 MODELO GRAFO DE PROPRIEDADES	37
4.5 NEO4J	38
4.6 CYPHER	40
5. ESTUDO DE CASO – MODELAGEM DE DADOS DO PROJETO OFICINA MÍMESES	43
5.1 CONTEXTUALIZAÇÃO	43
5.2 DADOS.....	44
5.3 AVALIAÇÃO DOS DADOS	44
5.4 MODELAGEM	45
5.5 CONSTRUÇÃO DE SENTENÇAS.....	57
5.6 CONSTRUÇÃO DE CONSULTAS E VISUALIZAÇÃO DOS DADOS.....	57
CONSIDERAÇÕES FINAIS	64
REFERÊNCIAS	66

INTRODUÇÃO

A sociedade é inserida em um cenário de grandes volumes de dados, e estes apresentam-se cada vez mais extensos e complexos, por possuírem diferentes fontes, formatos e terem rápido crescimento.

Estes grandes conjuntos são definidos pelo termo *Big Data*, que além de se referir a grandes volumes de dados, tem por objetivo aproveitar o melhor processamento desses dados na busca por informações que auxiliem a tomada de decisões. O desafio torna-se então além de armazenar, disponibilizar ferramentas e ambientes que estejam preparados para suportar as necessidades dessa nova era (BERNARDO; BORLONE, 2015).

O conceito do *Big Data* também é utilizado para caracterizar dados que já não podem ser processados pelos sistemas de banco de dados convencionais, devido à quantidade e velocidade com que crescem, não se encaixando assim nas restrições de arquiteturas de banco de dados (SCHNEIDER, 2012 apud HENRIQUES et al., 2013).

Informações das mais variadas estão dispostas por todos os lados, desde um cadastro hospitalar que contém informações sobre um paciente, um registro de alunos ingressos ou egressos em determinado curso de uma universidade, até os milhões de compartilhamentos de uma rede social durante um dia.

De acordo com Souza et al. (2014), nesse contexto, que é diferente da década de 70, quando surgiram os bancos de dados relacionais, as aplicações lidavam com dados estruturados, ou seja, possuíam estruturas bem definidas, estes bancos caracterizam-se por utilizarem uma linguagem de consulta chamada SQL (*Structured Query Language*). Segundo Lóscio et al., (2011) a alternativa para lidar com essa parcela de aplicações que possuem dados não estruturados e com rápido crescimento, é a abordagem NoSQL (*Not Only SQL*), que passa então a ser uma opção a um mercado que já não é atendido pelo modelo relacional.

Diante do contexto, o presente trabalho apresenta um referencial teórico a respeito dessa alternativa não relacional, e traz sua aplicação, bem como validação, através da modelagem e armazenamento de um conjunto de dados.

O conjunto de dados utilizado é proveniente do projeto Oficina Mímeses. Este projeto é de origem do Departamento de Filosofia da UERN (Universidade do Estado do Rio Grande do Norte), e visa estabelecer um canal de comunicação direta entre o

curso de licenciatura em Filosofia e os profissionais que lá obtiveram sua formação, assim como compreender o contexto no qual o ensino da Filosofia vem sendo abordado. A proposta apresentada no presente trabalho é desenvolver a modelagem em forma de grafos para os dados provenientes de planilhas eletrônicas utilizando o banco de dados Neo4j, sendo este pertencente à categoria NoSQL.

De tal forma, o objetivo geral do trabalho é apresentar a abordagem NoSQL para o armazenamento e modelagem dos dados adotando uma alternativa de bancos de dados não relacionais com um banco de dados de grafos.

De modo a alcançar o objetivo geral, existem alguns objetivos específicos, como:

- a) Contextualizar o termo NoSQL no que diz respeito a um breve histórico de seu surgimento, bem como os cenários que impulsionaram tal abordagem;
- b) Realizar uma comparação entre os bancos tradicionais (relacionais) com os bancos NoSQL, identificando assim diferenças entre eles;
- c) Descrever as características que identificam um banco de dados NoSQL;
- d) Apresentar as categorias existentes: chave-valor, orientado a colunas, orientado a documentos, e orientado a grafos;
- e) Abordar com mais detalhes a categoria orientada a grafos, a qual esse trabalho tem por escolha para modelagem da base de dados;
- f) Apresentar a ferramenta que será utilizada como aporte tecnológico, o banco de dados Neo4j;
- g) Validar a pesquisa tendo como fonte de dados o projeto de pesquisa intitulado: “Oficina Mímese de Pesquisa em Ensino de Filosofia”.

Para melhor apresentar este Trabalho de Conclusão de Curso, a estrutura do mesmo encontra-se organizada da seguinte maneira:

O Capítulo 2 aborda o termo NoSQL, sua origem, características e classificações, bem como uma comparação entre os SGBDs (Sistema de Gerenciamento de Banco de Dados) Relacionais e NoSQL. O Capítulo 3 relaciona os trabalhos mais relevantes em consideração ao que a pesquisa do trabalho se propõe. O Capítulo 4 traz um contexto a respeito do uso de grafos, um breve histórico dos bancos na categoria orientada a grafos, o modelo de grafo de propriedades, a base de dados escolhida para modelagem dos dados, e sua linguagem de consulta, Cypher.

O Capítulo 5 apresenta o estudo de caso, que caracteriza o problema e a importância em modelar os dados da Oficina Mímeses em grafos, a modelagem e armazenamento no Neo4j. E, por fim, as Considerações Finais concluem o trabalho, apresentado as vantagens, desvantagens e dificuldades encontradas na utilização do banco de dados NoSQL, e trabalhos futuros.

2. BANCOS DE DADOS NOSQL

Este capítulo traz uma contextualização a respeito do cenário dos grandes volumes de dados, a origem do termo NoSQL, e apresenta as características e classificações desses bancos não relacionais.

2.1 ORIGEM

O termo NoSQL conforme Sadalage e Fowler (2012), apareceu pela primeira vez no final dos anos 90, como um banco de dados relacional de código aberto, desenvolvido por Carlo Strozzi, chamado Strozzi NoSQL. O nome dado ao banco se deu pelo fato de não usar a SQL (*Structured Query Language*) como uma linguagem de consulta. Após isso o termo só apareceu e foi discutido em 11 de Junho de 2009, em São Francisco, nos Estados Unidos, durante um encontro organizado pelo desenvolvedor de software Johan Oskarsson.

Neste evento foram reunidos diversos trabalhos e projetos que utilizavam essa nova forma de armazenamento de dados, em sua maioria, inspirados pelo *BigTable* da Google e *Dynamo* da Amazon desenvolvidos antes de 2009, que impulsionaram a criação de diversas outras bases de dados NoSQL. Segundo Brito (2010), esse termo passou então a representar soluções que seriam uma opção ao modelo relacional, sendo chamado de NoSQL e com o propósito de oferecer uma alternativa para contextos em que o modelo SQL não apresentasse uma performance satisfatória.

2.2 CONTEXTUALIZAÇÃO

Desde o surgimento dos SGBDs na década de 70, o modelo relacional de dados têm sido a opção mais utilizada quando se fala de armazenamento, processamento e consulta dos dados estruturados, que são aqueles bem definidos, que possuem as mesmas descrições (atributos), e cujas descrições apresentam o mesmo formato (esquema).

Diferentemente de quando o modelo relacional surgiu, o tamanho de dados gerados nas aplicações *web* atuais é maior do que as aplicações com dados estruturados (SOUZA et al., 2014).

Diante da necessidade de gerenciar os volumes de dados que além de não serem estruturados, precisam de alta escalabilidade e alta disponibilidade, o modelo NoSQL surge como uma nova opção e alternativa para determinadas aplicações que estão diante de tal necessidade (LÓSCIO et al., 2011).

Moura e Amorim (2014) apresentam alguns exemplos que permitem a visualização do contexto onde o número de dados cresce e a demanda por escalabilidade só aumenta:

Em 2012, o universo digital de dados cresceu 2,72 *zettabytes* (ZB) e irá dobrar a cada dois anos, atingindo 8 ZB em 2015. Isso equivale ao acervo de 18 milhões de Bibliotecas do Congresso dos Estados Unidos. Dois bilhões de dispositivos conectados, que vão desde Computadores e *smartphones* a dispositivos com sensores tais como, leitores de *Rádio Frequency Identification* (RFID) e câmeras de tráfego, geram essa enxurrada de dados complexos, estruturados e não estruturados (MOURA; AMORIM, 2014).

Nesse contexto, onde o número de dados produzidos é cada vez maior, surgem mais desafios, e segundo Vieira et. al (2012) os bancos tradicionais não são os mais adequados, ou completos às necessidades do domínio do problema do *Big Data*, como por exemplo: execução de consultas com baixa latência, tratamento de grandes volumes de dados, escalabilidade horizontal, suporte a modelos flexíveis de armazenamento de dados, e suporte simples a replicação e distribuição dos dados, problemas estes que podem ser solucionados através da escolha de um banco NoSQL.

2.3 CARACTERÍSTICAS DOS BANCOS NOSQL

O NoSQL, é definido como uma nova geração de bancos de dados que possuem como características: ser não-relacional, distribuído, *open-source*, e escalável horizontalmente (NoSQL, 2016).

Além das características citadas acima, estes bancos também possuem algumas peculiaridades como:

Escalabilidade horizontal: diferente da abordagem vertical (*scale up*) que adiciona mais recursos (memória, disco rígido, processadores) em cima de um mesmo nó no sistema, a estratégia horizontal (*scale out*) é a melhor opção a ser utilizada quando o número de dados só tende a aumentar. Esta abordagem segundo Lóscio et

al. (2011), pode ser a solução mais viável, porém requer que diversas *threads* (processos) de uma tarefa sejam criadas e distribuídas, onde neste caso o modelo relacional teria o problema de alta concorrência, pois vários processos se conectariam simultaneamente em um mesmo conjunto de dados, ocasionando um tempo maior para o acesso das tabelas. O modo relacional trata o controle de concorrência de escrita com o uso de bloqueios (*locks*), garantindo que dois usuários não atualizem o mesmo item dos dados na mesma hora, já o NoSQL atua diferente no controle de concorrência e por isso ganha em escalabilidade.

Ausência de esquema ou esquema flexível: Um esquema de banco de dados é considerado por Codd (1970) o processo no qual os campos e tabelas presentes na base são simplificados e organizados com o intuito de reduzir a redundância dos dados. Esse processo é resultado da normalização das tabelas no modelo relacional, porém quando o número de dados aumenta nas tabelas, o processo de normalizar tende a ficar mais lento (JACOBS, 2009 apud CARDOSO, 2012).

Tentando minimizar o tempo gasto com normalização de tabelas e uma das características que podem ser identificadas nos bancos NoSQL é a ausência quase que completa do uso de esquema pré-definidos para modelar os dados. Cardoso (2012) ainda afirma que:

Passa a existir então uma base de dados não relacional mais relaxada que armazena dados não estruturados em qualquer formato sem uma normalização ou definição rígida prévia com um desempenho aceitável para as grandes exigências do *Big Data*.

O fato de não ter um esquema para definir e modelar os dados ajuda na escalabilidade, e também na alta disponibilidade do sistema, porém não garante a integridade dos dados devido à flexibilidade da modelagem, diferente do banco relacional que tem um esquema rígido e garante a total integridade dos seus dados (SOUZA et al., 2014; LÓSCIO et al., 2011).

Suporte nativo à replicação: a replicação dos dados também é uma forma de escalar o sistema, e sendo uma replicação nativa o tempo gasto para fazer a recuperação das informações é menor. Lóscio et al. (2011), dizem que existem duas formas principais para essa replicação: *master-slave* (mestre-escravo) que se dá pela criação de um servidor nó chamado de *master*, o principal, e um outro nó chamado

slave, secundário. *Multi-master* (muitos mestres) onde vários nós são mestres, o que significa que qualquer nó pode responder as solicitações e replicar os dados aos demais.

Consistência Eventual: os SBGBs tradicionais não conseguem distribuir os dados de forma elástica pois seu modelo de garantir a consistência é baseado nas propriedades ACID (Atomicidade, Consistência, Isolamento, Durabilidade), e tal controle é inviável quando se trata dos nós e processamentos em vários nós (VIEIRA et. al 2012).

Sabendo disso, em geral, o modelo NoSQL utiliza o modelo BASE (Basicamente Disponível, Estado Leve, Consistência Eventual), do inglês: (*Basically Available, Soft State, Eventual consistency*) que implementa um modelo de consistência de dados eventual, proposta essa que é baseada no teorema CAP do inglês: (*Consistency, Availability, Partition Tolerance*), inicialmente elaborado por Eric Brewer (2000) e provado posteriormente por Gilbert e Lynch (2002) dando origem ao teorema de Brewer (BREWER, 2000).

Figura 1:Teorema Cap



Fonte: Próprio autor (2017)

Tal teorema afirma ser possível a ocorrência simultânea de duas das três propriedades, em um cenário distribuído e de grande porte. As propriedades são:

- a) Consistência: garantir que todos que acessem visualizem o mesmo valor dos dados;
- b) Disponibilidade: garantir que todas as solicitações recebam respostas;
- c) Tolerância à partição: garantir que o sistema continue disponível e operando mesmo em meio a falhas.

Segundo Brewer (2000), só são possíveis as seguintes combinações entre as propriedades:

Consistência e tolerância ao particionamento: aqui o sistema precisa dispensar pelo menos por um momento a disponibilidade. Um exemplo citado em (PANIZ, 2016) descreve que usando uma aplicação do tipo *master-slave* os dados só serão ditos salvos quando todas as réplicas confirmarem a escrita, evitando que os dados estejam desatualizados, e qualquer leitura retornaria os dados corretos, com alta consistência. Mas para confirmar essa escrita aumenta-se a latência, e caso ocorra uma falha de comunicação, os dados estarão consistentes, mas as operações de escritas estarão perdidas, então o sistema não estará disponível e nesse tipo de situação, perde-se a disponibilidade, porém são garantidas a consistência e tolerância ao particionamento.

Consistência e disponibilidade: nesse caso o sistema possui consistência e está disponível mas descarta-se a tolerância ao particionar. Em caso de falha na rede nem todos os pedidos serão atendidos, uma vez que só existe um nó, como por exemplo um servidor único que receberá tais solicitações. Exemplos: bancos de dados relacionais (STEPPAT, 2011).

Alta disponibilidade e tolerância ao particionamento: sistemas que não podem ficar *off-line* e não se importam em perder a consistência dos dados para ganhar resolução dos conflitos. Exemplos: Dynamo, Cassandra (CARDOSO, 2012).

Brito (2010) diz que “ao substituir um banco relacional por uma solução NoSQL, a arquitetura perde em consistência, mas pode ganhar em flexibilidade, disponibilidade e performance”. Bancos de dados NoSQL disponibilizam uma consistência eventual devido ao fato de não ser possível manter os dados consistentes entre todos os nós distribuídos.

Surge então um outro conceito onde o sistema opta por perder a consistência para obter respostas mais rápidas de uma forma que seja mais escalável, que é o paradigma BASE, ou seja, os dados são distribuídos em diferentes locais sendo assim sempre disponíveis, o que faz com que o sistema pareça estar funcionando todo o tempo. Por não precisar ser consistente todas as horas, ele é consistente no momento devido, por isso a consistência eventual (PEREIRA *et. al* 2013).

2.4 RELACIONAL VERSUS NOSQL

Queiroz et al (2013) diz que os SGBDs se apoiam na fundamentação teórica de Codd (1970), chamada de modelo relacional de dados. Nesse modelo os dados estão armazenados em tabelas que são compostas por linhas e colunas, onde cada coluna possui um determinado atributo de dado, e cada linha um dado referente a este mesmo atributo.

Os SGBDs relacionais oferecem aos usuários a validação, verificação e garantias de integridade dos dados, controle de concorrência, recuperação de falhas, segurança, controle de transações, entre outros.

Costa (2011) e Cuer (2014) citam que a arquitetura de um SGBD divide-se em três níveis:

- a) Físico: onde descreve os dados como estão realmente armazenados;
- b) Conceitual: modelo entidade e relacionamento, mostrando o relacionamento entre os tipos de dados, e o que está compreendido no banco;
- c) Externo: onde os dados são mostrados ao usuário.

Com base nesses modelos, os dados são previamente modelados em esquemas. E feitas as normalizações das tabelas, onde os dados são separados de acordo com elementos distintos em tabelas diferentes. Estes bancos mantêm o controle das operações através das propriedades ACID, que garantem a integridade, e consistência, evita a redundância entre os dados e utiliza a linguagem SQL como consulta padrão.

Estes recursos permitiram que os bancos relacionais fossem os únicos a serem utilizados em larga escala desde seu surgimento, apresentando limitações devido o novo cenário dos dados. Alguns dos pontos fracos do modelo relacional são:

- a) O custo e a dificuldade em escalar o sistema;
- b) A difícil adaptação às dinâmicas do desenvolvimento de software que mudam constantemente trazendo problemas para esse modelo que se torna inflexível a estas mudanças por apresentar uma estrutura pré-definida;
- c) A organização dos dados em um sistema distribuído;
- d) A complexidade das consultas SQL (BRITO, 2010; CUER, 2014; TOTH, 2011).

Em contrapartida, a abordagem não relacional não tem uma forma específica para tratar os dados. Na abordagem NoSQL os dados podem ser armazenados e

manipulados de diversas formas: através de colunas, chaves e valores, através de grafos, ou em forma de documentos.

Assim, esse modelo não tem uma linguagem de consulta padrão como a SQL, ou seja, cada linguagem terá uma forma distinta de definir os seus comandos e assim fazer as consultas (CARDOSO, 2012).

O controle de transações desses bancos, no geral, não se baseiam nas propriedades ACID, mas no paradigma chamado BASE.

De acordo com Pereira (2013), apesar de não seguir as mesmas propriedades, o banco NoSQL tem como pontos fortes ser livre de esquema, ou seja, podem tratar dados não estruturados, e garantir alta disponibilidade, escalabilidade e uma consistência eventual, onde tais características representam o que um determinado grupo de aplicações de fato deseja.

No que diz respeito a escalabilidade, ou seja, a forma com que os bancos se adaptam à rápida expansão dos dados, os bancos relacionais tendem a não ser a opção mais viável, pois a demanda por escalabilidade só aumenta, e para que um banco relacional conseguir alcançar acaba se tornando uma tarefa de custo elevado e de grande complexidade (TOTH, 2011).

A consistência é o ponto forte do modelo relacional, e o fraco ao utilizar uma solução NoSQL que requer perda de consistência, pelo ganho da disponibilidade, flexibilidade e performance. Essa decisão de perder um ponto e ganhar em outros é baseada no Teorema CAP. Atualmente, alguns bancos de dados NoSQL já implementam o modelo de consistência ACID.

2.5 CLASSIFICAÇÃO DOS BANCOS NOSQL

Os bancos de dados NoSQL são classificados em 4 grandes categorias: chave-valor (*key-value*); orientado a colunas (*column*); orientado a documentos (*documents*); orientado a grafos (*graph-based*).

2.5.1 Chave-valor

Dentre todas as categorias, esta é a mais simples. O banco chave-valor tem sua estrutura comparada com uma tabela *hash*, que associa chaves a seus respectivos valores (CARDOSO, 2012; FREITAS, 2015; LÓSCIO et al., 2011).

Como dito por Friess (2013), esses sistemas podem conter dados estruturados ou não estruturados, os dados são endereçados por uma única chave e são úteis para aplicações que necessitem de operações simples, que sejam baseadas somente nas chaves. O modelo chave-valor não agrupa dados de acordo com as entidades, ou tipos de dados como no modelo relacional, nessas bases toda a informação está contida em uma única cadeia de dados (FREITAS, 2015).

Exemplos de bancos chave-valor: Cassandra (APACHE CASSANDRA, 2017), Redis (REDIS, 2017), Ehcache (EHCACHE, 2017), Aerospike (AEROSPIKE, 2017).

2.5.2 Colunas

Similar ao modelo relacional por apresentar linhas e colunas, e uma evolução do modelo chave-valor os bancos orientados a colunas armazenam os dados de forma colunar. Sua orientação não é através das linhas (tuplas), e sim através dos atributos (colunas) (CARDOSO, 2012).

Freitas (2015) ressalta que na prática esse tipo de abordagem permite que análises mais eficientes sejam feitas em subconjuntos de dados.

As principais vantagens do modelo colunar em relação ao relacional, citadas por Soares e Boscaroli (2013), estão ligadas à:

a) Compressão: possibilidade de se manter dados do mesmo tipo comprimidos, onde apenas a base colunar oferece melhor capacidade de organizar dados do mesmo tipo;

b) Materialização: maneira de como recuperar as informações armazenadas no banco, que feitas no modo colunar evitam a vinda de tuplas desnecessárias;

c) Iteração: trata-se da forma como as consultas são executadas.

Um dos primeiros bancos desenvolvidos da categoria foi o *Bigtable*, que teve seu início por meados de 2004, projetado pela Google (SILVA, 2011). O *Bigtable* é um sistema de armazenamento distribuído que foi projetado para gerenciar dados estruturados, e adaptar-se à *petabytes* através de milhares de servidores.

Alguns exemplos de bancos de dados orientados a colunas: Hbase (APACHE HBASE, 2017), MonetDB (MONETBD, 2017), Vertica (VERTICA, 2017).

2.5.3 Documentos

Este tipo de modelo surgiu para armazenar e gerenciar documentos, ou dados semi-estruturados (FRIESS, 2013). Nesse caso também é utilizado um par chave-valor pois cada documento inserido na base é associado a uma chave única que irá representá-lo.

Apesar dessa semelhança, Lóscio (2011) destaca a diferença entre os modelos dizendo que no modelo chave-valor “apenas uma única tabela *hash* é criada para todo o banco. No modelo orientado a documentos temos um conjunto de documentos e em cada documento temos um conjunto de campos (chaves) e o valor deste campo.” O conjunto de campos que podem existir dentro dos documentos podem ser desde *strings*, até listas ou outros objetos aninhados (CUER, 2014).

O modelo orientado a documentos não possui um esquema rígido como nos modelos relacionais, onde cada campo possui uma estrutura definida para cada tipo de dado, dessa forma esta base permite atualização dos documentos que já foram inseridos, e até a adição de novos campos sem causar futuros problemas para a mesma (LÓSCIO et al., 2011).

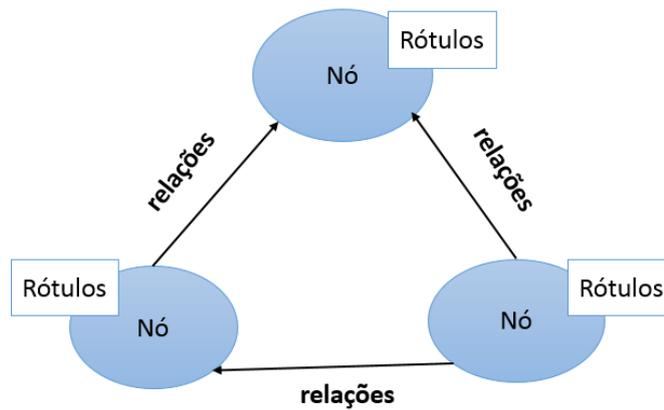
Portanto, para aplicações onde se deseja manter armazenados dados em formato de documento, tem-se essa possibilidade. Por último este padrão tende a possuir consultas mais complexas, visto que não temos apenas chaves e valores, mas inúmeros documentos de diferentes tipos (CARDOSO, 2012).

Exemplos de bancos orientados a documentos: MongoDB (MONGODB, 2017), CouchDB (APACHE COUCHDB, 2017).

2.5.4 Grafos

Este modelo é otimizado para o processamento eficiente de conjuntos de dados interligados. Os bancos orientados a grafos são os únicos dentre as categorias de bancos NoSQL que se preocupam com as relações e representação visual dos dados (MONIRUZZAMAN; HOSSAIN, 2013).

Possui como elementos principais o conjunto de nós (vértices), arestas (relações) e as propriedades (atributos dos nós ou relações). A Figura 2 apresenta um exemplo do modelo em grafos.

Figura 2: Modelo em grafos

Fonte: Próprio autor (2017)

Com tal representação é possível modelar inúmeros problemas do mundo real como um grafo, ou seja, essa abordagem permite representar diversos domínios de aplicação, como redes de computadores, web semântica, sistemas de recomendação de conteúdo e muitos outros. Além de conseguir modelar dinâmica e naturalmente os dados, o modelo de grafos permite uma visualização mais familiar para os programadores e um fácil entendimento mesmo por pessoas leigas ao assunto.

Uma das empresas que adotou o modelo orientado a grafos foi a Walmart, a mesma lida com milhões de clientes por semana através das suas 11.000 lojas espalhadas em 27 países, e através do seu site de varejo em 10 países. O desenvolvedor do comércio eletrônico do Brasil explica: “Um banco de dados relacional não estava satisfazendo nossas necessidades sobre o desempenho e simplicidade, devido à complexidade das nossas consultas” (Neo4j, 2015, tradução nossa).

A solução encontrada pelo grupo foi utilizar o banco de dados Neo4j, que consegue compreender o comportamento daqueles consumidores online e as relações entre eles e seus produtos, sendo uma ferramenta de recomendações em tempo real que atende muito bem o que se espera, a solução fornecido pelo base de dados orientada a grafos foi de rapidamente consultar as compras anteriores realizadas pelos clientes e rapidamente mostrar novos interesses na atual lista dos clientes online para fazer recomendações em tempo real.

Com Neo4j poderíamos substituir um complexo processo que usamos para preparar o nosso banco de dados relacional com um tempo real simples e de dados de gráficos. Poderíamos construir um sistema de

recomendação simples e em tempo real com consultas de baixa latência (Neo4ja, 2015, tradução nossa).

Outros exemplos de bancos de grafos: OrientDB (ORIENTDB, 2017), Titan (TITAN, 2017), ArangoDB (ARANGODB, 2017). Desses bancos citados todos implementam as propriedades ACID.

3. TRABALHOS RELACIONADOS

Na literatura existem diversas pesquisas relacionadas ao uso da abordagem NoSQL, inclusive na categoria orientada a grafos. Alguns trabalhos, os quais são considerados mais relevantes para o escopo deste, são apresentados nas Seções a seguir deste Capítulo.

3.1 SELEÇÃO DE UM SGBD NOSQL

Antes de tomar a decisão de utilizar uma abordagem NoSQL é fundamental entender se a aplicação que vai utilizá-la de fato terá um melhor desempenho, além de atender suas expectativas, se usar um modelo NoSQL.

Nesse âmbito, podemos destacar alguns trabalhos que visam estabelecer um análise comparativa entre as categorias de bancos NoSQL, comparação entre os bancos tradicionais e NoSQL, além de uma análise de desempenho entre ambos.

Souza et al., 2014 em sua pesquisa dos Critérios para Seleção de SGBD NoSQL, tem como objetivo identificar os principais critérios para escolher um SGBD em organizações privadas. A pesquisa identificou na literatura, e também através de grupo de 32 especialistas no tema os principais critérios de seleção. Os critérios para tal seleção e avaliação dos SGBDs foram extraídos de uma revisão sistemática apresentada em uma seção do mesmo trabalho, além da fundamentação teórica que foi complementada pelo trabalho de Elmasri e Navathe (2005).

Os resultados obtidos na análise sistemática foram analisados com um grupo de profissionais que atuam ou tem experiência em SGBD NoSQL. Para essa avaliação foi aplicado um questionário onde os profissionais confirmavam ou não o grau de importância dos critérios para seleção do SGBD NoSQL identificados na literatura. Depois de avaliados, surgiram mais três critérios.

Brito (2010) apresenta uma análise comparativa entre os bancos tradicionais e NoSQL, apresentando as limitações do modelo relacional, as características do modelo NoSQL e o comparativo entre os dois, que tem como destaque os termos de escalabilidade, consistência e disponibilidade uma vez que se opta por escolher um dos dois SGBDs, como mostrado na Figura 3:

Figura 3: Análise Comparativa Modelo Relacional x NoSQL

	Relacional	NoSQL
Escalonamento	Possível, mas complexo. Devido à natureza estruturada do modelo, a adição de forma dinâmica e transparente de novos nós no <i>grid</i> não é realizada de modo natural.	Uma das principais vantagens desse modelo. Por não possuir nenhum tipo de esquema pré-definido, o modelo possui maior flexibilidade o que favorece a inclusão transparente de outros elementos.
Consistência	Ponto mais forte do modelo relacional. As regras de consistência presentes propiciam uma maior grau de rigor quanto à consistência das informações.	Realizada de modo eventual no modelo: só garante que, se nenhuma atualização for realizada sobre o item de dados, todos os acessos a esse item devolverão o último valor atualizado.
Disponibilidade	Dada a dificuldade de se conseguir trabalhar de forma eficiente com a distribuição dos dados, esse modelo pode não suportar a demanda muito grande de informações do banco.	Outro fator fundamental do sucesso desse modelo. O alto grau de distribuição dos dados propicia que um maior número de solicitações aos dados seja atendida por parte do sistema e que o sistema fique menos tempo não-disponível.

Fonte: Brito (2010)

Por último o trabalho de Cuer (2014), que faz a comparação de desempenho entre os dois SGBDs. Para sua pesquisa o autor fez uso dos bancos relacionais: MySQL, PostgreSQL e o banco NoSQL, MongoDB, para apresentar o comparativo ele teve como foco as operações de inserção, busca e exclusão para ambos os bancos. Para que os ambientes de teste fossem os mesmos, o autor ainda utilizou o software VMware, que virtualizou um ambiente para cada banco de dados.

Como conclusão, Cuer (2014) destacou que nos testes de inserção e escrita o banco NoSQL teve desempenho ruim comparada aos bancos relacionais. Nos testes de busca o modelo NoSQL superou o modelo relacional, nos testes de alteração o banco relacional MySQL teve melhor desempenho, seguido do MongoDB. E nos testes de remoção o banco que demonstrou maior estabilidade foi o MySQL. Em resumo o MongoDB teve melhor desempenho na operação de busca, o MySQL teve melhor estabilidade nos testes, e o PostgreSQL teve melhor desempenho nos testes de inserção de dados.

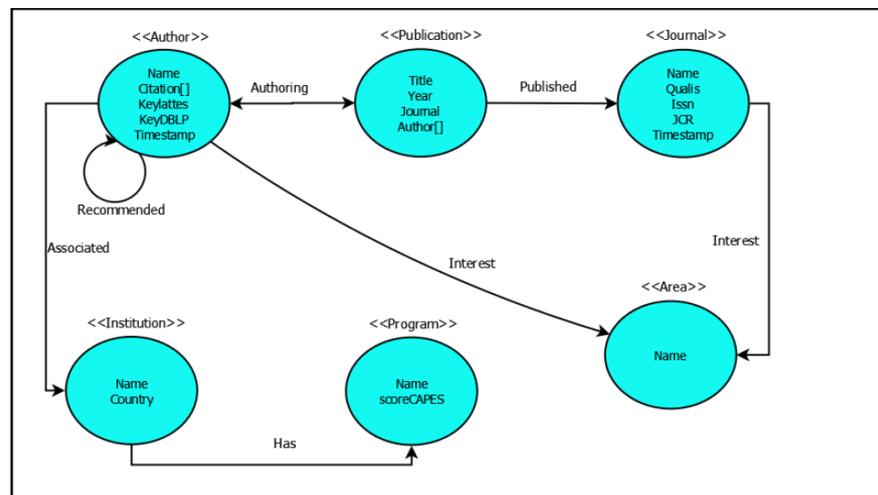
3.2 MODELAGEM EM GRAFOS

Ao utilizar os bancos orientados a grafos, os pesquisadores notam dificuldades em modelar os dados que serão armazenados, isto porque até então, na literatura não existe um modelo padrão que abranja todos os bancos orientados a grafos.

A pesquisa de Costa (2016) foi fundamentada na análise das Redes de Colaboração Científica, ou seja, nas interações sociais entre os agentes nas publicações de suas pesquisas, além de envolver os sistemas de recomendação, onde desenvolveu um modelo de recomendação de parceria para pesquisas científicas construindo um banco de dados NoSQL baseado em grafos que armazenava sua rede de colaboração científica de pesquisadores.

A solução proposta consistia em definir um modelo de recomendações e a arquitetura de um sistema que implementasse o modelo. A modelagem dos dados utilizou o modelo de entidades inspirado na teoria dos grafos dirigidos, por não encontrar um modelo de dados NoSQL orientado a grafos que fosse amplamente aceito na comunidade de banco de dados. O modelo orientado a grafos foi adotado pelo autor por aproveitar melhor as características e conceitos das redes de colaboração. O modelo de dados usado para representar as entidades do modelo de recomendação encontra-se na Figura 4.

Figura 4: Modelo de dados usado para representar as entidades no modelo de recomendação



Fonte: Costa (2016)

A construção do artefato que implementasse o modelo de recomendações usou determinadas tecnologias, como o banco de dados Neo4j, por possuir o melhor desempenho de consultas, para o armazenamento da rede de colaboração.

O modelo de solução foi implementado em um artefato de software, e a fim de verificar a viabilidade, um estudo de caso foi conduzido, sendo possível responder o questionamento da possibilidade de se definir e implementar um artefato para analisar as redes de colaboração científica. A viabilidade foi comprovada por avaliações quantitativas e qualitativas. O autor ainda sugere para trabalhos futuros melhorar o desempenho das recomendações geradas, expandir a base de dados para abranger mais pesquisadores, e otimizar o desempenho da geração de recomendações, para que seja possível realizar um conjunto de recomendações de imediato após um pesquisador ser inserido na rede.

A pesquisa de Erven (2015), destaca a importância da modelagem de dados, sendo esta uma etapa fundamental na organização dos dados que devem ser armazenados. Partindo dessa necessidade Erven (2015) propõem em seu trabalho o Modelo de Dados para Bancos NoSQL Baseados em Grafo (MDG-NoSQL), com o objetivo de suprir a lacuna presente com uma notação para modelar diagramas mais simples às estruturas de grafos mais complexas. Para validação do modelo o autor construiu um banco de vínculos entre entidades do poder executivo envolvidas no

processo licitatório, e também apresentar uma comparação entre o modelo de dados em grafos e o relacional.

3.3 MAPEAMENTOS ENTRE O MODELO RELACIONAL E O NOSQL

Freitas (2015) identifica em sua pesquisa algumas questões que envolvem os bancos NoSQL, como a possibilidade em converter uma base de dados do modelo relacional à abordagem NoSQL, as possíveis correspondências entre as duas estruturas, e se existe alguma compatibilidade entre os modelos, a fim de permitir uma forma automática a conversão dessas estruturas. O objetivo do trabalho foi definir uma abordagem de conversão das estruturas de dados entre o modelo relacional e os diferentes modelos NoSQL. A solução proposta foi semiautomática, diminuindo assim a participação do usuário.

As correspondências definiram o nível de similaridade entre as estruturas, sendo feita uma descrição para cada modelo NoSQL para os conceitos de: tabela, atributos simples, atributos multivalorados e compostos, restrições, relacionamentos, resumo das correspondências.

Um mapeamento das principais categorias de bancos NoSQL foi realizado, apresentando um comparativo entre eles e o modelo relacional. Freitas (2015) observou que a capacidade de representar os conceitos comuns ao modelo relacional variavam de acordo com a abordagem escolhida: chave-valor, colunas, documentos, e grafos.

O protótipo da ferramenta de conversão foi implementado, e com ele foram realizados experimentos para validar a hipótese da pesquisa. Para o estudo de caso, o sistema de banco destino NoSQL escolhido foi o MongoDB, que é baseado em documentos.

Os experimentos realizados por Freitas (2015) consistiam na execução de consultas nos dois bancos (origem e destino) e indicaram a viabilidade da abordagem. Nestes, foram comparados os resultados das consultas submetidas nas bases de origem e destino que apresentaram resultados similares.

Freitas (2015) ainda faz uma demonstração que apresenta os conceitos do modelo relacional e sua correspondência no modelo orientado a grafos utilizando o Neo4j mostradas na Figura 5:

Figura 5: Quadro de correspondência de conceitos

Conceito	Relacional	Grafos (Neo4j)
Conjunto de entidade	tabela ou conjunto de dados	rótulo
Instância	tupla	nó
Atributo		
Simples	atributo	propriedade
Composto	dividido em atributos simples ou tabelas com dependência entre elas	conjunto de propriedades ou novo nó
Multivalorados	dividido em atributos ou tabelas com dependência entre elas	conjunto de propriedades
Nulo	possui (obrigatório)	não possui
Restrições		
Identificador de entidade	chave primária	id
Referência	foreign Key	possui (aresta)
Integridade Referencial	possui	possui
Relacionamento (1:1, 1:N, M:N)	referência por chave estrangeira	elemento especial (relacionamento)

Fonte: Adaptado de Freitas (2015)

Em resumo, o modelo orientado a grafos no Neo4j, apresenta alguns conceitos semelhantes ao modo relacional como por exemplo os atributos, que são chamados de propriedades. A forma como os relacionamentos entre as entidade são modeladas, é por meio das relações estabelecidas através das arestas, que armazenam um nó de entrada e um nó de saída, fazendo associações entre os nós. De forma simples o número de arestas sobre um nó determina a intensidade das relações sobre ele.

Portanto, o autor contribui com uma alternativa em que o processo de conversão de dados por grupos de desenvolvedores que necessitem migrar do Modelo Relacional para o NoSQL possa ocorrer de modo mais simples e ágil.

3.4 BANCOS ORIENTADOS A GRAFOS (NEO4J)

Novamente pode-se relacionar o trabalho de Erven (2015), que além da proposta de modelagem, traz uma análise comparativa entre consultas de bancos de dados em grafos e bancos relacionais. O autor preparou um ambiente para os testes computacionais. Para o modelo relacional foi utilizado o banco MySQL e a linguagem

padrão SQL. Para o banco de grafos foi escolhido o Neo4j, com sua linguagem também declarativa Cypher.

Os objetivos do teste eram identificar relacionamentos fraudulentos no processo de licitação, através de três consultas em cada um dos bancos. As seguintes questões foram levantadas para identificar tal caso:

- a) Questão 1: Quais empresas possuem o mesmo telefone?
- b) Questão 2: Quais itens de licitação duas empresas concorreram juntas?
- c) Questão 3: Como duas empresas se relacionam a partir de suas sociedades até o terceiro nível de empresa?

Para o primeiro questionamento, só foi possível obter tal relação fazendo a função JOIN entre as seguintes tabelas do banco relacional: EmpresasTelefones, Empresas e Telefones. A consulta pode ser vista na Figura 6.

Figura 6: Consulta de empresas que possuem o mesmo telefone

```

1  SELECT
2    CL.nome_empresa as empresaA ,
3    T.numero_telefone ,
4    CR.nome_empresa as empresaB
5  FROM EmpresasTelefones AS CT01
6    INNER JOIN Empresas AS CL
7      ON CL.pk_empresa = CT01.fk_empresa
8    INNER JOIN Telefones AS T
9      ON T.pk_telefone = CT01.fk_telefone
10   INNER JOIN EmpresasTelefones AS CT02
11     ON CT02.fk_telefone = CT01.fk_telefone
12   INNER JOIN Empresas AS CR
13     ON CR.pk_empresa = CT02.fk_company
14  WHERE CT01.fk_empresa <> CT02.fk_empresa ;

```

Fonte: Erven (2015)

O resultado da consulta é mostrado na Figura 7:

Figura 7: Resultado da consulta de telefones

company_name	telephone_number	company_name
COMPANY A	55550001	COMPANY G
COMPANY G	55550001	COMPANY A
COMPANY B	55550002	COMPANY D
COMPANY D	55550002	COMPANY B
COMPANY E	55550004	COMPANY H
COMPANY H	55550004	COMPANY E

Fonte: Erven (2015)

No resultado pode-se observar que as relações estão duplicadas devido o produto cartesiano da tabela Empresas x Empresas.

A mesma consulta foi realizada no modelo de grafos, apenas especificando a origem, ou seja, uma empresa qualquer, e o destino também do tipo empresa, através do relacionamento de tem_numero entre si. Para se chegar aos telefones das empresas foi levado em consideração a direção da aresta, que nesse caso era bidirecional.

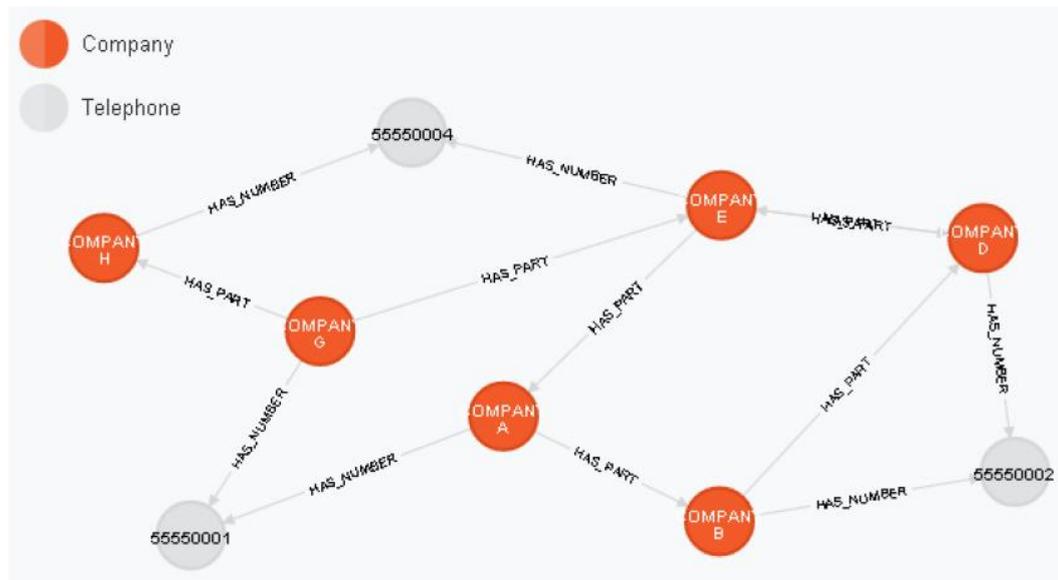
Figura 8: Busca de empresas relacionadas por telefone utilizando o Cypher

```
MATCH (empresa0:Empresas) -[r:TEM_NUMERO*1..2]- (empresaD:Empresas)
RETURN empresa0, r, empresaD
```

Fonte: Erven (2015)

Observando a Figura 8, um outro aspecto foi levado em consideração por Erven (2015). No relacionamento também foi especificado o número de saltos que seriam avaliados até se chegar a uma empresa. Como existiam apenas dois telefones entre elas, o número de saltos seria dois, especificado na consulta por *1..2. O resultado foi obtido com o uso da cláusula *return*, que especifica o retorno das duas empresas e os relacionamentos entre elas na Figura 9.

Figura 9: Resultado da consulta para empresas por telefones no Neo4j



Fonte: Erven (2015)

Foi retornado o grafo das empresas que se relacionam por telefones, através do padrão que foi especificado (Empresa->Telefone<-Empresa).

Em suas conclusões o autor afirma que a comparação entre as duas abordagens permitiu mostrar como cada um dos modelos se comporta, gerar considerações sobre a modelagem em grafos, além das diferenças entre ela e modelagem relacional.

Através dos grupos de consultas realizadas o autor percebeu também que o banco de dados em grafos mostrou vantagem quando as consultas envolviam as navegações entre os relacionamentos, assim como as consultas mais complexas que ficaram mais simples quando utilizadas no ambiente de grafos.

4. BANCOS DE DADOS ORIENTADOS A GRAFOS

Este Capítulo tem como objetivo apresentar o contexto no qual o banco orientado a grafos está inserido e porque é utilizado. Neste Capítulo é apresentada a definição de grafos, a modelagem dos mesmos, um breve histórico sobre os bancos em grafos, a justificativa de uso de grafos como modelo de dados, o modelo de grafo de propriedades, o banco de dados Neo4j e a linguagem Cypher.

4.1 DEFINIÇÃO DE GRAFOS

Os grafos são bastante utilizados em diversas áreas e são muito úteis na compreensão de uma grande diversidade de conjuntos de dados. O mundo físico possui muitos dados interligados, tanto uniformes como não uniformizados, e a forma como o modelo relacional modela tais dados é bastante restritiva (ROBINSONS et al., 2015).

Um grafo é um objeto matemático, uma estrutura formada por dois conjuntos, o primeiro compreende ao conjunto de vértices e o outro ao conjunto de relações entre os vértices, as arestas (NETTO; JURKIEWICZ 2009).

Segundo Neto e Jurkiewicz (2009), se pode criar um esquema gráfico que irá associar cada vértice a uma determinada aresta, tendo assim uma maneira abstrata de representar problemas do mundo real em grafos. Tal representação ocorre através de pontos e linhas, sendo os vértices representados por pontos, e as linhas por sua vez as arestas que se conectam aos pontos, como exemplificado na Figura 10.

Figura 10: Pontos e linhas representando um grafo



Fonte: Próprio autor (2017)

4.2 HISTÓRICO

Conforme Satone (2014), as pesquisas para banco de dados em grafos começaram por volta de 1990 com modelos de bancos como: LDM, GOOD, O2 e GraphDb. Esses bancos surgiram como uma alternativa aos bancos de dados relacionais, e com isto foi dada uma maior importância aos sistemas que tem a interconectividade dos dados como aspecto essencial. Apesar do sucesso inicial, os banco de dados de grafos perderam espaço de maneira rápida para os bancos que tratavam os dados semiestruturados, que na época os sistemas de base de dados baseados em XML, do inglês (*eXtensible Markup Language*), estavam sendo adotados (PENTEADO et al., 2014).

Na primeira década dos anos 2000 o interesse pelos bancos de dados de grafos foi impulsionado pela web semântica e pela necessidade de manipular as relações entre grandes volumes de dados, isso devido à grande quantidade de dados que vinha surgindo e ao movimento NoSQL que visava novas abordagens para armazenar e manipular diferentes tipos de dados (ERVEN, 2015).

4.3 MODELAGEM EM GRAFOS

As modelagens de dados são formas abstratas onde pode-se trazer determinadas características de um domínio de problema para um cenário em que é possível estruturar e manipular tais dados. A forma de modelagem relacional perde um pouco a representação natural do domínio do problema, forçando-a ao modelo físico, não havendo uma harmonia entre o modelo lógico e o físico, onde por sua vez o modelo em grafos tem bastante afinidade entre os dois modelos (ROBINSON et al., 2015).

Um modelo pode ser pensado como coleções de objetos (livros, pessoas), e as relações que existem entre objetos os elementos destas coleções (tipos, gostos). Graficamente isso pode ser representado como pontos e linhas, formando assim uma estrutura de grafos (RODRIGUEZ; NEUBAUER, 2010).

Os bancos em grafos mantém uma preocupação maior em tratar os dados implicitamente conectados e, diferente do modelo relacional, é estabelecido uma forma coesa do problema como um todo. Onde existir ligações no domínio do problema existirão conexões entre os dados, dessa forma o banco orientado a grafos

mantém os dados que são conectados, armazenados também como dados interligados, ou seja, preservando o cenário do problema a ser representado (ROBINSON et al., 2015).

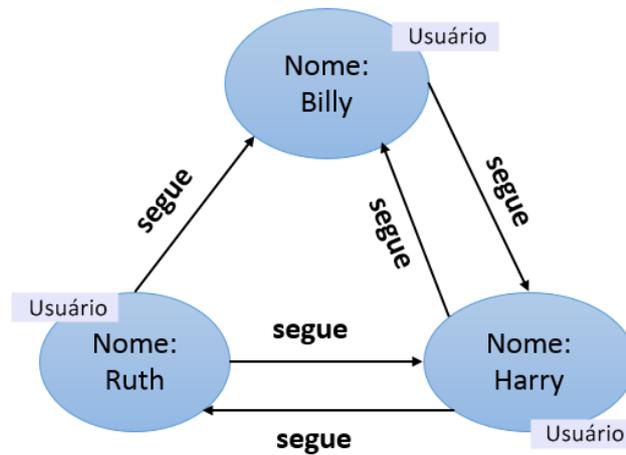
A modelagem lógica de um sistema gerenciador de bancos de dados em grafos (SGBDG), pode ser expressa pelo modelo básico de grafos, como: $G = (V, E)$, conjunto de vértices e arestas contido em um grafo G . A forma de mapear o modelo lógico para o físico classifica os SGBDG em nativos ou não-nativos. Os nativos preservam tanto a estrutura do grafo no armazenamento dos dados quanto no processamento das consultas. Já os não-nativos modelam grafos no nível lógico mas armazenam os dados em outros modelos, por exemplo em tabelas relacionais (PENTEADO et al., 2014).

Em relação à forma de armazenar, os sistemas podem ser centralizados ou distribuídos. Centralizados são aqueles cujos dados se localizam em apenas um servidor. Já o distribuído particiona o banco em partes e estas são fisicamente distribuídas entre vários servidores, solução esta que visa melhorar a disponibilidade nas aplicações que lidam com grandes quantidades de dados (DATE (2004) apud PENTEADO et al., 2014).

4.4 MODELO GRAFO DE PROPRIEDADES

Rodriguez e Neubauer (2010) definem o grafo de propriedades (*Graph Property Model*) como um grafo multi-relacional que contém atributos e arestas direcionadas. Como exemplo desse tipo de grafo temos a Figura 11 que representa uma rede social entre amigos.

Figura 11: Grafo representando rede social entre amigos



Fonte: Adaptado de Robinson et al., 2015

Neste modelo os nós possuem propriedades, e podem ser marcados com um ou mais rótulos, isto para agrupar os nós de acordo com a sua representação dentro de um conjunto de dados. Os relacionamentos conectam os nós e fazem a estruturação do grafo, eles possuem um nome e uma direção, fazendo com que se tenha uma clareza semântica sobre os dados. As propriedades que podem ser adicionadas aos nós e relacionamentos servem para adicionar qualidade ou determinados pesos aos dados, fazendo com que consultas sejam melhor otimizadas em tempo de execução (ROBINSON et al., 2015).

4.5 NEO4J

O Neo4j é um banco orientado a grafos, *open-source*, desenvolvido em Java, e que é compatível com as propriedades ACID. É transacional e oferece dois tipos de arquiteturas, a centralizada e a distribuída. Desenvolvido pela empresa *Neo Technology, Inc.*, lançado em 2010, o Neo4j possui duas versões: a *Community Edition*, e a versão empresarial *Enterprise Edition*, esta última tem como diferença ser código fechado e fornece um suporte empresarial aos seus usuários, além de dispor de um número maior de recursos para garantir o desempenho de aplicações maiores (PENTEADO et al., 2014).

Esse banco contém implementações dos algoritmos mais comuns para grafos, como os de caminho mais curto, de todos os caminhos, de todos os caminhos simples. Entre tantas características o banco em grafos consegue lidar com milhões

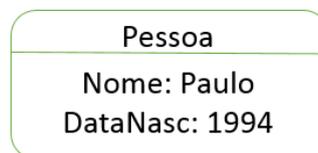
de nós inter-relacionados de forma otimizada, podendo destacar-se ainda a sua capacidade de integração às diversas plataformas de desenvolvimento existentes através de sua API (*Application Programming Interface*) (SANTOS; SILVA, 2013; LOPES, 2014).

Suportando o modelo de grafo de propriedades, o Neo4j possui uma linguagem de consulta própria, chamada de Cypher, podendo também suportar outras linguagens como SPARQL (*SPARQL Protocol and RDF Query Language*), e a *Gremlin* (ROBINSON et al., 2015).

Para inserir os dados no Neo4j e assim compor o grafo, é preciso fazer a adição de nós, e relacionamentos, com seus respectivos rótulos e propriedades (LOPES, 2014).

Nós: Nós são usados para representar entidades, ou seja, interesses do domínio do problema que podem ser rotulados e agrupados. Um nó pode ter associado a ele, relações e propriedades. Além de poder ser marcado com *labels* (Neo4j, 2016).

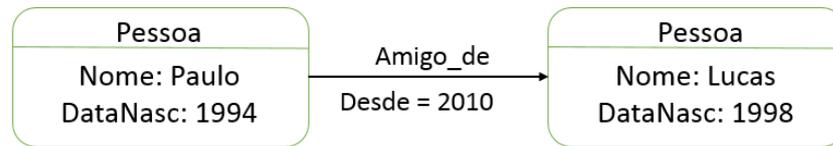
Figura 12: Exemplo de nó



Fonte: Adaptado de Andrade (2016)

Como mostrado na Figura 12, um nó pode ter um *label* (rótulo: pessoa), e possuir propriedades: nome e data de nascimento.

Relações: Parte fundamental de um banco de dados orientado a grafos, as relações permitem encontrar os dados relacionados entre si. Da mesma forma que os nós, as relações também podem ter propriedades. As relações organizam os nós ligando-os, conectando um nó de início e um nó final. Ou seja, expressam as conexões entre as entidades, e estabelecendo um contexto semântico que estrutura o domínio (ROBINSON et al., 2015).

Figura 13: Exemplo de relações

Fonte: Adaptado de Andrade (2016)

A Figura 13 exemplifica a relação de amizade entre os nós do tipo pessoa, com o relacionamento “Amigo_de”, que por sua vez possui a propriedade que caracteriza desde quando os dois objetos são amigos.

Propriedades: Semelhante aos atributos no modelo relacional o modelo em grafos utiliza o conceito de propriedade. As propriedades são valores nomeados através de uma *string*, que são associados tanto a nós quanto as relações, associando metadados necessários ao problema (FREITAS, 2015; ROBINSON et al., 2015).

Podem suportar os seguintes tipos de dados: *strings*, inteiros, *float*, *double*, *char*, *booleans*. Uma observação importante é que as propriedades não podem ser armazenadas com *null*, caso uma determinada propriedade não possua um valor em um nó ou relacionamento, simplesmente aquele nó ou relacionamento não deve possuir aquela propriedade (Neo4j, 2016).

Rótulos: No Neo4j quando é preciso categorizar os nós usamos o conceito de *label*, que significa rótulo. Um nó pode ter zero ou mais rótulos associados a ele, além disso, qualquer sequência de caracteres Unicode não vazio pode ser usado como um rótulo.

4.6 CYPHER

A linguagem *Cypher* foi projetada para ser facilmente lida e compreendida pelos desenvolvedores, profissionais de banco de dados, e qualquer pessoa envolvida no projeto. Isso porque a sua forma de representação é bastante similar à forma como nós representamos os grafos (ROBINSON et al., 2015).

É inspirada por diferentes abordagens e baseia-se em práticas estabelecidas para consultas expressivas, a maior parte das suas palavras-chave como *where* e

order by são inspirada em SQL. A capacidade de corresponder padrões vem das abordagens da SPARQL, entre outras (Neo4jb, 2016).

Inspirado em SQL, Cypher descreve padrões visuais em grafos utilizando uma sintaxe de arte ASCII (*American Standard Code for Information Interchange*). Esse padrão permite que o desenvolvedor crie diversas cláusulas como filtros, agregações, além das operações básicas de uma base de dados: CRUD (*Create, Read, Update, Delete*) (SANTOS; SILVA, 2013; Neo4jc, 2016).

A seguir, na Figura 14 é demonstrado a criação da relação “amigo_de”, estabelecendo a amizade entre duas pessoas. No trecho de código, foi usada a cláusula *create* para criar ambos os nós e a relação entre eles.

Figura 14: Exemplo criação de nós e relação em Cypher

```
create (paulo:Pessoa{nome:'Paulo',datanascimento:'1994'}),
      (lucas:Pessoa{nome:'Lucas',datanascimento:'1998'})
create (paulo)-[:amigo_de]->(lucas)
```

Fonte: Próprio autor (2017)

Para encontrar as informações inseridas no banco usa-se a cláusula *match* que determina um padrão a ser procurado, como mostrado na Figura 15.

Figura 15: Exemplo consulta match

```
match (p1:peessoa) -[:amigo_de] -> (p2:peessoa)
return p1,p2
```

Fonte: Próprio autor (2017)

Outras cláusulas podem ser adicionadas de forma a filtrar a consulta, na Figura 15, o uso do *return* define quais declarações seriam retornadas e exibidas ao usuário.

Outras Cláusulas: Robinson et al., (2015), ainda descreve outras cláusulas oferecidas pela linguagem, como por exemplo:

- a) *Match*: está no centro da maior parte das consultas, é responsável por montar um padrão de busca que retorne os nós e relações de interesse;
- b) *Return*: responsável por retornar as consultas da forma com o que o usuário a especifique, você pode por exemplo retornar linhas ou agregações de dados, somas e muitas outras coisas características;

- c) *Delete*: apaga nós ou a relação inteira;
- d) *Where*: fornece critérios para filtragem de correspondência de padrões;
- e) *Merge*: assegura que o padrão estabelecido exista no grafo antes de criá-lo.

5. ESTUDO DE CASO – MODELAGEM DE DADOS

Para validação da pesquisa proposta com a alternativa NoSQL, foi utilizado como fonte de dados, os dados do Projeto Oficina Mímese, do departamento de Filosofia, da UERN.

5.1 CONTEXTUALIZAÇÃO

O projeto de pesquisa intitulado por: Oficina Mímese de Pesquisa em Ensino de Filosofia, com data de submissão de 05 de janeiro 2013, atendendo o Edital 004/2012-DP/PROPEG/UERN, teve doze meses de duração, e como coordenador o professor Ms. William Coelho de Oliveira, do Departamento de Filosofia da UERN campus Central. O resumo do projeto traz o seguinte texto:

O projeto de interação Universidade-Escolas, através de pesquisa e extensão sobre estratégias do ensino da Filosofia, em Mossoró e região atendida pela UERN, constitui-se em investigar as dificuldades pedagógicas locais da Filosofia, objetivando a produção conjunta de textos didático-filosóficos e o debate sobre o papel da Filosofia nas Escolas. Visa promover e incentivar a interação entre alunos e professor de Filosofia da UERN e das escolas parceiras. Além disso, deve proporcionar a auto avaliação e direcionamento da prática pedagógica do Curso de Filosofia, propiciando campo de pesquisa e demanda para um Programa de Pós-graduação em Filosofia na UERN (OLIVEIRA, 2013).

Como justificativa o projeto apresenta os seguintes dados à respeito do curso de Filosofia da UERN:

O curso de Filosofia da UERN foi criado no ano de 2003. Em 2012 ingressou a 10ª turma, em Mossoró. Em 2008 o ensino da Filosofia tornou-se obrigatório no ensino médio, até essa data o curso havia formado apenas duas turmas. Em 2011 deu-se o único concurso público para professores de Filosofia no Rio Grande do Norte (OLIVEIRA, 2013).

Dessa forma, a proposta central do projeto era levantar possíveis questionamentos, como saber quantos diplomados existem na cidade de Mossoró, quantos ensinam Filosofia, como os mesmos trabalham os conteúdos filosóficos com seus alunos, entre tantas questões que podem avaliar a prática pedagógica do curso.

E o objetivo geral da Oficina Mímese consistia em fomentar a interação entre Universidade e Escola através de pesquisa e extensão sobre estratégias do ensino

da Filosofia, em Mossoró e região, junto a professores e alunos do curso de Licenciatura em Filosofia da UERN e das escolas locais. Para alcançar tal objetivo o projeto destaca dois objetivos específicas:

1-Diagnosticar as condições do ensino de Filosofia nas Escolas de Mossoró-RN, proporcionando entre elas e a UERN um ambiente de construção dialógica para estudantes e professores de Filosofia exercitarem sua profissionalização através do discurso crítico e investigativo acerca da realidade educacional local. 2- Promover a produção de material didático eletrônico (e-book flip) e o uso da tecnologia educacional para aplicação na disciplina de Filosofia, no ensino médio (OLIVEIRA, 2013).

5.2 DADOS

Dentro do projeto Oficina Mímese, existem alguns dados que foram utilizados para a modelagem proposta nesse trabalho, os quais são:

- a) Planilha com dados dos licenciados em Filosofia da UERN em Mossoró, dos anos de 2006 à 2012;
- b) Questionários que foram aplicados com alunos, professores, gestores das escolas; e
- c) Relatório final do projeto.

Para efeito deste Trabalho de Conclusão de Curso, a modelagem concentrou-se apenas no questionário com os professores de Filosofia do ensino básico.

5.3 AVALIAÇÃO DOS DADOS

Para realizar a modelagem dos dados da Oficina Mímese, primeiro foi preciso avaliar os dados disponíveis provenientes do questionário de Cadastro de Professor de Filosofia do Ensino Básico, armazenados em vários arquivos.

Parte dos dados presentes no questionário escolhido também eram os mesmos da planilha dos licenciados, divergindo apenas por conter campos para adição de informações sobre os dados profissionais do licenciado, e questões voltadas para a avaliação do curso, e disciplina de Filosofia.

A princípio foi possível notar a dificuldade na gestão dos dados dos questionários, pois havia muitos arquivos, o que dificultava o controle sobre versões que fossem alteradas.

Ao verificar a necessidade de um banco de dados, avaliou-se que para utilizar o modelo relacional seria necessário definir exatamente as estruturas das tabelas, pois após a criação do banco determinadas alterações que viessem a ser feitas, como atualização de campos em tabelas ou remoção de algum atributo, provocaria uma mudança no esquema do banco de dados.

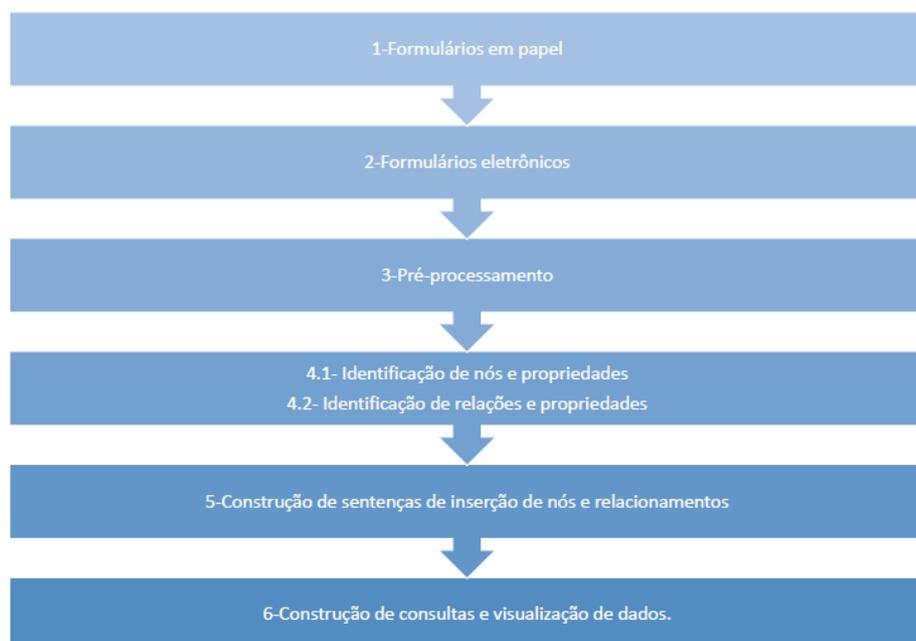
Dadas as dificuldades pode-se perceber que para sair de planilhas para um banco de dados relacional, surgiriam muitos problemas ligados à normalização dos dados, devido aos campos vazios nos questionários, e os possíveis dados que teriam o mesmo valor.

Partindo desses pontos o presente trabalho se propôs a modelar esses dados em uma estrutura orientada à grafos, a qual tem por objetivo priorizar a interconectividade entre as informações por meio de relacionamentos e propriedades deste relacionamento. Então o modelo de dados para a modelagem foi o de grafo de propriedades.

5.4 MODELAGEM

Para desenvolver a modelagem organizou-se um fluxo de trabalho (*workflow*), como mostra a Figura 16:

Figura 16: Fluxo de trabalho (workflow)



Fonte: Próprio autor (2017)

Nas etapas 1 e 2, migrou-se do modelo em papel para o modelo eletrônico, através da ferramenta gratuita Google Forms, que permite criar formulários online. Isto foi feito para que depois de preenchidos com dados¹ a visualização das respostas do formulário estivessem em planilhas.

A etapa 3 foi realizada no intuito de tratar os dados com alguns ruídos, como por exemplo, o nome da mesma escola escrito de formas diferentes, reduzindo a redundância entre os dados.

As etapas 4.1 e 4.2, ocorrem em paralelo, pois quando se identifica os nós, determinamos suas propriedades, e as relações entre si, o objetivo de tais etapas foi analisar quais colunas das planilhas de respostas virariam nós e quais seriam as suas propriedades e quais relações associadas a cada nó.

As etapas 5 e 6 estão descritas na sessão 5.5 e 5.6.

Na Figura 17 temos a sessão do formulário correspondente as informações básicas, e endereços das pessoas que responderam o questionário online:

Figura 17: Sessão de informações básicas e endereço do questionário

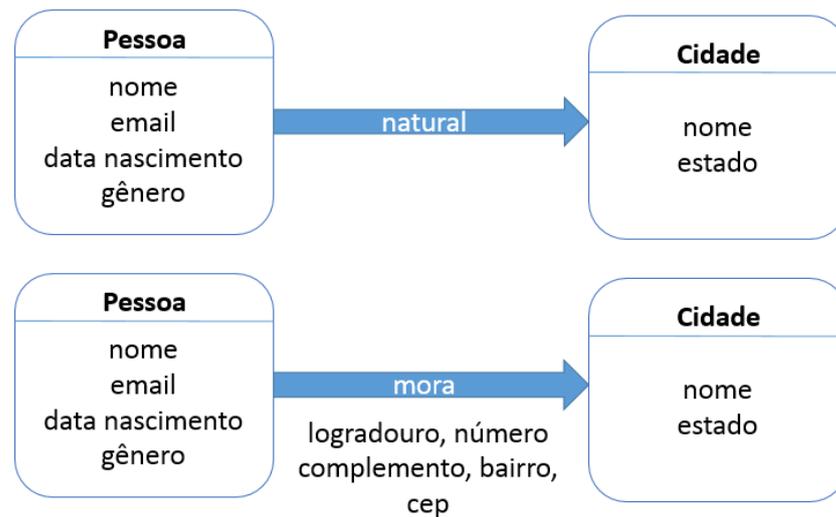
Informações Básicas	Endereço
Nome: *	Logradouro
Email: *	Número
Data de nascimento *	Complemento
Sexo/Genêro *	Bairro
Naturalidade *	Cep
	Estado *
	Cidade *

Fonte: Próprio autor (2017)

¹ Os exemplos utilizados neste trabalho tratam-se de dados fictícios para não expor o conteúdo do projeto Oficina Mímeses.

A partir dessas sessões foi possível identificar que pessoas e cidades são nós, e pessoas possuem a relação de “morar” e ser “natural” de uma cidade, como mostra a Figura 18, com os respectivos nós, relações e propriedades:

Figura 18: Elementos identificados na sessão informações básicas e endereço



Fonte: Próprio autor (2017)

A Figura 19 apresenta as sessões sobre a formação acadêmica das pessoas, que podem ser: graduação, especialização, mestrado e doutorado.

Figura 19: Sessão de informações sobre a formação acadêmica graduação

Formação Acadêmica - Graduação

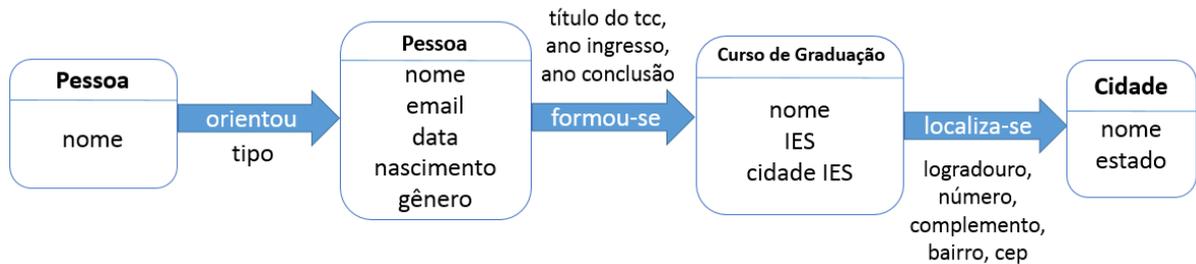
Curso de Graduação *	Título do Trabalho de Conclusão de Curso
Modalidade *	Professor Orientador *
Instituição de Ensino Superior *	Ano Ingresso *
Cidade da Instituição de Ensino Superior *	Ano Conclusão *

Fonte: Próprio autor (2017)

Foram extraídas as relações de que uma pessoa “formou-se” em um curso de graduação, e que ela foi orientada por um professor orientador, sendo assim, o

professor é um nó também do tipo pessoa, com a relação “orientou” determinada pessoa. Na Figura 20, temos os nós, relações e propriedades identificados:

Figura 20: Elementos identificados na sessão formação acadêmica – graduação



Fonte: Próprio autor (2017)

O nó pessoa também pode possuir outros tipos de formação, que estão apresentadas na Figura 21:

Figura 21: Sessão de informações sobre a formação acadêmica, especialização, mestrado e doutorado

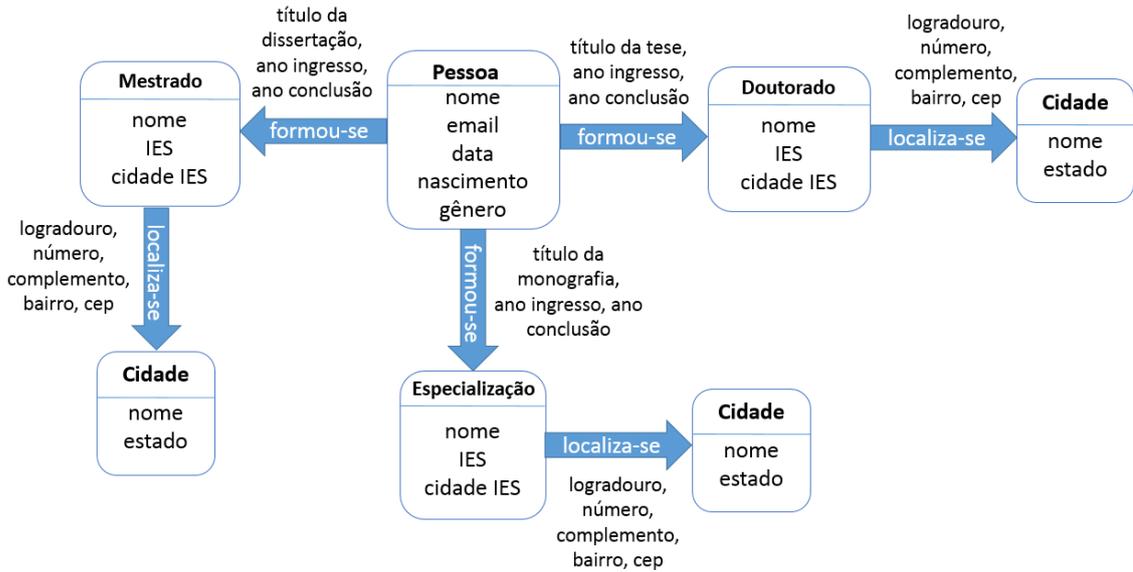
Formação Acadêmica - Especialização	Mestrado	Doutorado
Possui especialização? *	Possui mestrado? *	Possui Doutorado? *
Nome: *	Nome: *	Nome: *
Instituição de Ensino Superior *	Instituição de Ensino Superior *	Instituição de Ensino Superior *
Cidade de Instituição de Ensino *	Cidade de Instituição de Ensino *	Cidade da Instituição de Ensino *
Ano Ingresso *	Ano Ingresso *	Ano Ingresso *
Ano Conclusão *	Ano Conclusão *	Ano Conclusão *
Título da Monografia	Título da Dissertação *	Título da Tese *
Professor Orientador *	Professor Orientador *	Professor Orientador *

Fonte: Próprio autor (2017)

Aqui foi identificado que a pessoa possui novamente a relação “formou-se”, que pode ser em: especialização, mestrado ou doutorado, cada uma das opções passa então a ser um nó do tipo específico, e essa formação tem a relação de

“localizar-se” em uma determinada cidade, então o nó pessoa passa a ter mais três relações de formou-se, a Figura 22 mostra os respectivos nós, relações e propriedades:

Figura 22: Elementos identificados na sessão formação acadêmica, especialização, mestrado e doutorado



Fonte: Próprio autor (2017)

Não foi mostrado a criação do nó pessoa e relações “orientou”, pois essas características referentes ao professor orientador, já foram mostradas anteriormente.

A sessão que traz as informações profissionais, ou seja, referente à disciplina Filosofia que a pessoa leciona nas escolas, segue na Figura 23:

Figura 23: Sessão de informações profissionais

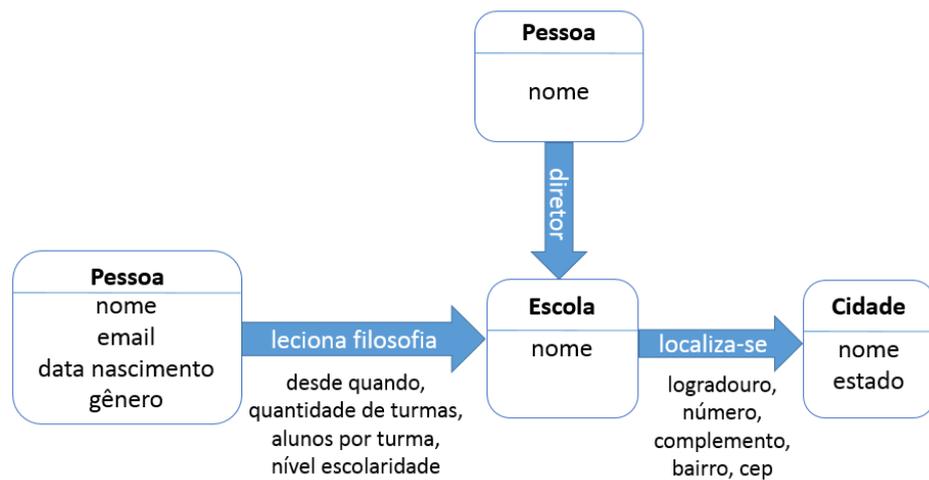
Dados Profissionais

Nome da Instituição *	Quantas turmas? *
Diretor/Coordenador	Quantidade de alunos por turma? *
Ensina na Instituição desde quando? *	Qual nível? *

Fonte: Próprio autor (2017)

Quanto aos dados profissionais, foi identificado os nós de pessoa que descreve o diretor da escola, onde o mesmo está relacionado à escola pela relação “diretor”, e a relação de “lecionar Filosofia” na escola que está associada ao nó do tipo pessoa. O nó do tipo escola tem a relação de “localizar-se” em uma cidade. Na Figura 24 temos estes elementos e suas propriedades:

Figura 24: Elementos identificados na sessão dados profissionais



Fonte: Próprio autor (2017)

No questionário ainda há uma sessão de outras disciplinas, as quais podem ser lecionadas além de Filosofia, que é mostrada na Figura 25:

Figura 25: Sessão outras disciplinas

Outras Disciplinas

Ensina alguma disciplina além de Filosofia? * Quantidade de alunos por turma? *

Quais? *

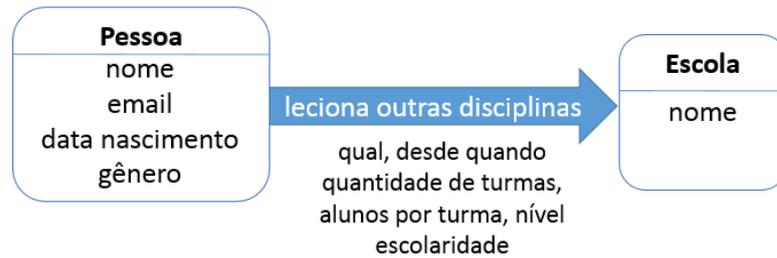
Qual nível? *

Quantas turmas? *

Fonte: Próprio autor (2017)

A identificação foi apenas da relação “lecionar outras disciplinas” entre o nó do tipo pessoa, e o nó do tipo escola, como segue demonstrado na Figura 26:

Figura 26: Elementos identificados na sessão outras disciplinas



Fonte: Próprio autor (2017)

Caso o nó do tipo pessoa ainda lecionasse outra disciplina, se aplicaria novamente a relação de “leciona outra disciplina”, onde as diferenças seriam apenas nos novos dados das propriedades referentes à relação.

A partir da sessão sobre a disciplina lecionada, as próximas perguntas do formulário tinham várias opções de resposta, que deveriam ser preenchidas conforme a Figura 27:

Figura 27: Sobre as questões

Sobre a disciplina lecionada

Nas questões a seguir, inserir uma resposta com os valores entre 1 e 4, conforme o grau de importância de cada tópico: (1=pouco/fraco/ruim),(2=médio/regular),(3=muito/forte/bom), (4=excelente/extrema).

Fonte: Próprio autor (2017)

Cada pergunta do formulário poderia identificar novos nós, relações e atributos, ou apenas alguma adição aos elementos já modelados. Na Figura 28, encontra-se as perguntas dessa etapa:

Figura 28: Perguntas sobre a disciplina lecionada

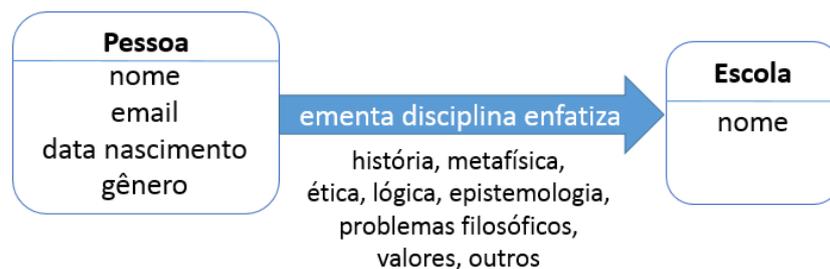
- 1-O que o programa/ementa da disciplina de Filosofia enfatiza?
- 2-Como o professor tem trabalhado os conteúdos filosóficos com seus alunos?
- 3-Quais as principais dificuldades?
- 4-O que tem ensinado, realmente da Filosofia?
- 5-E o que tem aprendido com tal prática?
- 6-Com quais problemas filosóficos tem se deparado na relação com o alunado adolescente?
- 7-Como a Universidade pode interagir com a Escola no tocante à compreensão dos problemas filosóficos e seu ensino na nossa realidade educacional de Nordeste do Brasil no interior do Rio Grande do Norte?
- 8-Em que a Filosofia lhe tem servido e à sociedade, para além da docência, uma vez que seu conteúdo foi considerado "necessário ao exercício da cidadania" (Lei 9.934/96)?
- 9-Como a Filosofia pode contribuir para melhorar a educação dos futuros universitários, para que eles cheguem à Universidade propedeuticamente melhor do que chegaram aqueles que hoje lhes ensinam?

Fonte: Próprio autor (2017)

Cada uma das perguntas tinha uma série de tópicos a serem avaliados com um número entre 1 e 4 como mostrou a Figura 27. Essas opções de avaliação foram identificadas como propriedades de determinadas relações referentes à cada uma das perguntas. Cada uma dessas propriedades representa um valor inteiro, que corresponde à resposta dada pela pessoa.

A pergunta um identificou uma nova relação entre o nó pessoa e a escola, como mostra a Figura 29:

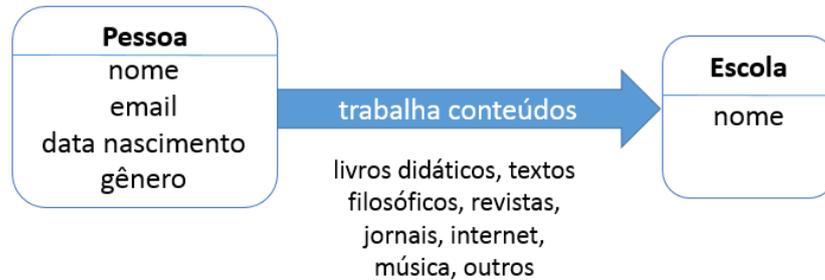
Figura 29: Relação identificada na pergunta um



Fonte: Próprio autor (2017)

A pergunta dois identificou a relação “trabalha conteúdos”, entre o nó pessoa e a escola, como mostra a Figura 30:

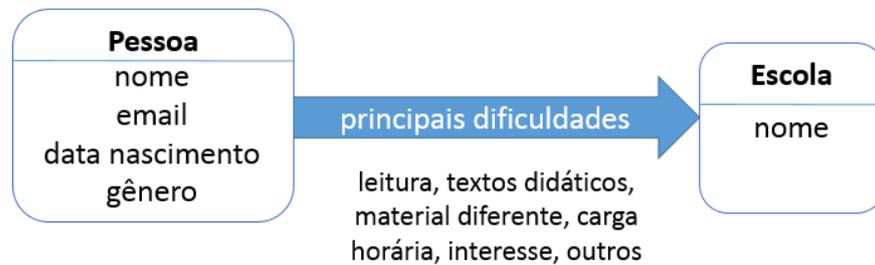
Figura 30: Relação identificada na pergunta dois



Fonte: Próprio autor (2017)

A pergunta três identificou a relação “principais dificuldades”, entre o nó pessoa e a escola, como mostra a Figura 31:

Figura 31: Relação identificada na pergunta três



Fonte: Próprio autor (2017)

A pergunta quatro identificou a relação “ensina sobre”, entre o nó pessoa e a escola, como mostra a Figura 32:

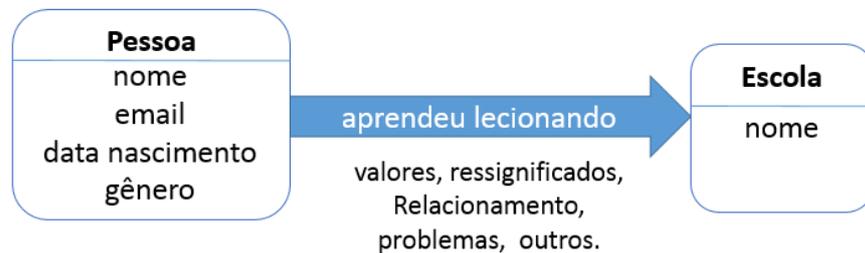
Figura 32: Relação identificada na pergunta quatro



Fonte: Próprio autor (2017)

A pergunta cinco identificou a relação “aprendeu lecionando”, entre o nó pessoa e a escola, como mostra a Figura 33:

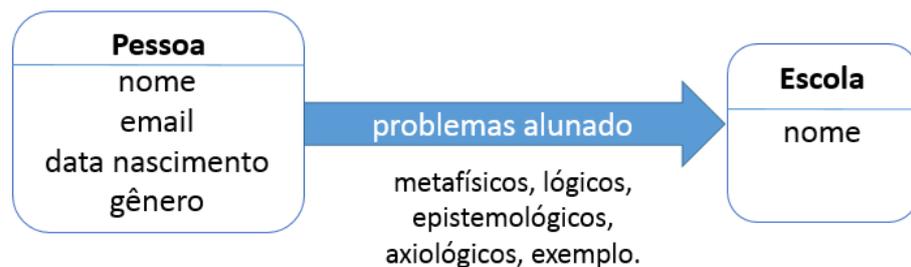
Figura 33: Relação identificada na pergunta cinco



Fonte: Próprio autor (2017)

A pergunta seis identificou a relação “problemas alunado”, entre o nó pessoa e a escola, como mostra a Figura 34:

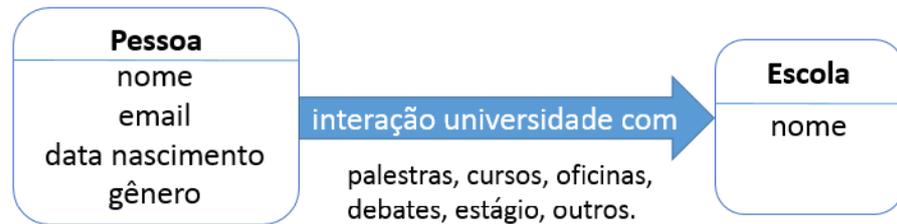
Figura 34: Relação identificada na pergunta seis



Fonte: Próprio autor (2017)

A pergunta sete identificou a relação “interação universidade com”, entre o nó pessoa e a escola, como mostra a Figura 35:

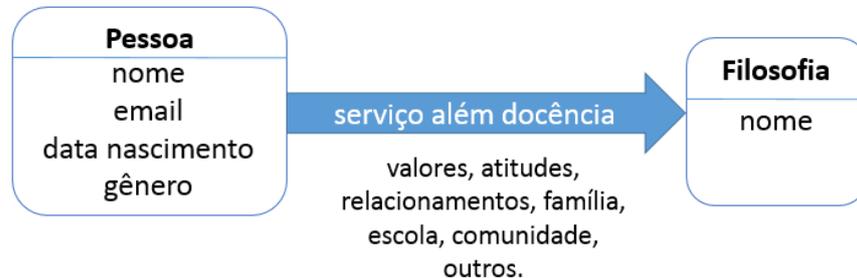
Figura 35: Relação identificada na pergunta sete



Fonte: Próprio autor (2017)

A pergunta oito identificou a necessidade da criação de um novo nó, determinado como Filosofia, no que se refere à área, onde terá a relação de “serviço além docência”, entre o nó pessoa e o nó Filosofia, como mostra a Figura 36:

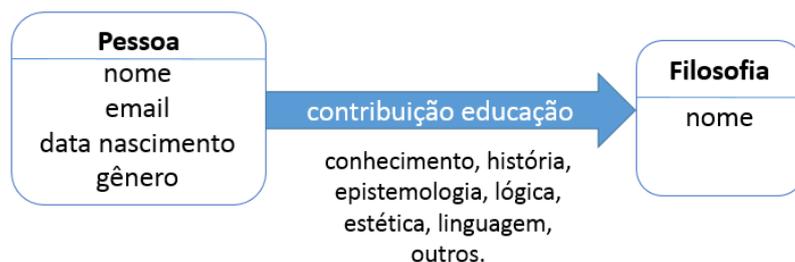
Figura 36: Nó e relação identificada na pergunta oito



Fonte: Próprio autor (2017)

A pergunta nove identificou a relação “contribuição educação”, entre o nó pessoa e o Filosofia, como mostra a Figura 37:

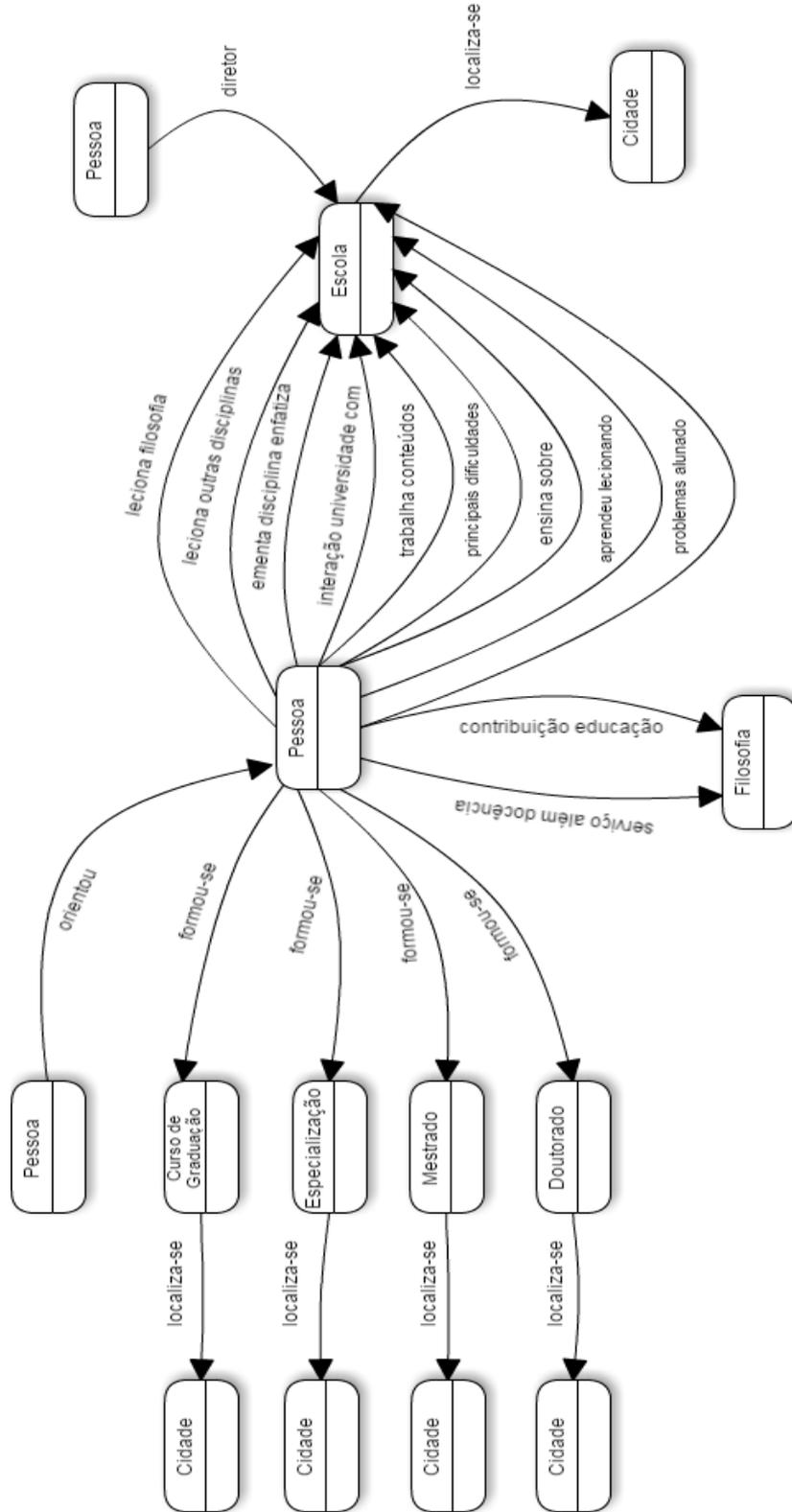
Figura 37: Relação identificada na pergunta nove



Fonte: Próprio autor (2017)

E por fim a Figura 38 apresenta a modelagem completa do questionário, com seus nós, relações.

Figura 38: Modelagem completa



Fonte: Próprio autor (2017)

5.5 CONSTRUÇÃO DE SENTENÇAS

Depois de identificados todos os nós, seus rótulos, propriedades e relações, foram construídas suas sentenças de criação através da linguagem Cypher, no Neo4j. A construção das consultas foi feita com os comandos básicos demonstrados no sessão 4.5 deste trabalho.

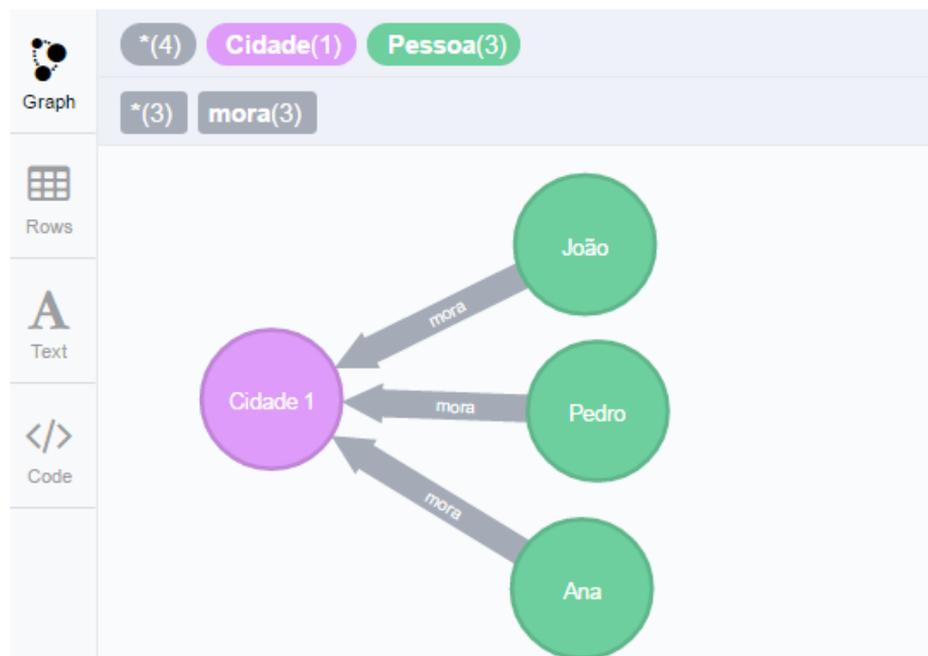
5.6 CONSTRUÇÃO DE CONSULTAS E VISUALIZAÇÃO DOS DADOS

Foram feitas algumas consultas sobre os dados, para exemplificar como se dá a visualização dos dados modelados, inseridos no Neo4j. Para isso as consultas que serão mostradas a seguir priorizaram as relações entre os dados nesse trabalho modelado.

De tal forma a Figura 39 mostra a consulta que retorna os nós que tenham a relação “mora”, e todas as informações que estejam entre eles.

Figura 39: Consulta que retorna os nós que tenha a relação mora

```
1 MATCH (p) -[r:mora]->(c)
2 RETURN *
```



Fonte: Próprio autor (2017)

A consulta mostrada traz primeiro a visualização da resposta em forma de grafo. Porém, com a mesma consulta, é possível ter outro tipo de visualização, sendo o resultado em linhas. Para isso, clica-se no ícone que está circulado na Figura 40. Vale salientar que para retornar os dados apenas em linhas pode ser feito uma pequena modificação na mesma consulta. O objetivo deste exemplo é mostrar os dois tipos de visualização em uma mesma consulta.

Figura 40: Resposta da consulta em linhas

```

1 MATCH (p)-[r:mora]->(c)
2 RETURN *
    
```

	c	p	r
estado	Estado	data_nascimento	13/03/1980
nome	Cidade 1	nome	Ana
		email	ana@email.com
		gênero	Feminino
		complemento	(empty)
		bairro	Primeiro bairro
		logradouro	Rua Primeira
		número	1400
		cep	5000000

Fonte: Próprio autor (2017)

Os exemplos a seguir apresentaram todas as relações modeladas, em específico as relações do nó com nome “Pedro”. A Figura 41, mostra a consulta que retorna as relações de “mora” e “naturalidade” em linhas, através da especificação da cláusula *return*, onde “c.nome” traz a cidade ao qual ele mora, “a.nome”, a cidade ao qual ele é natural, e “m” corresponde ao endereço que está nas propriedades associados à relação “mora”:

Figura 41: Consulta e resposta da relação mora e natural em linhas

```

1 MATCH (c)<-[m:mora]-(p:Pessoa{nome:"Pedro"})-[n:natural]->(a)
2 RETURN c.nome,a.nome,m
    
```

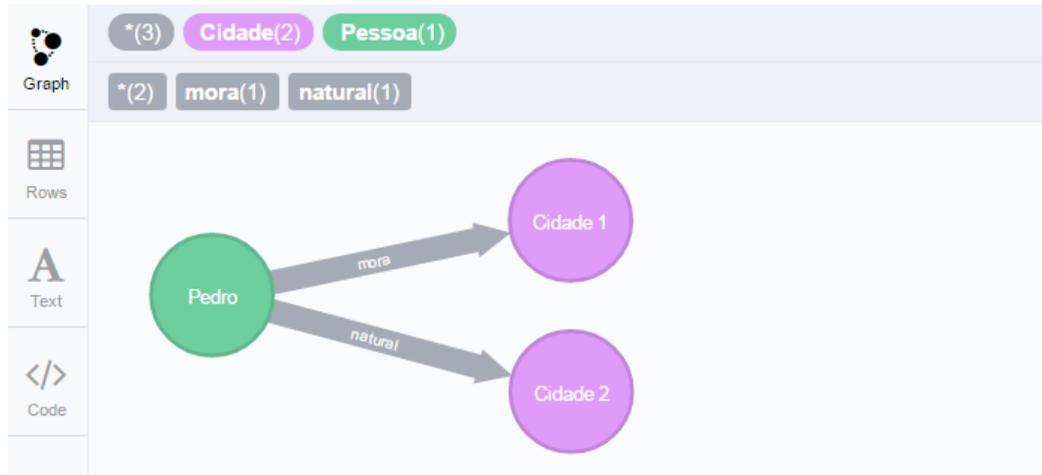
	c.nome	a.nome	m
	Cidade 1	Cidade 2	complemento (empty)
			bairro Bairro do Bairro
			logradouro Rua Rua Rua
			número 540
			cep 5700000

Fonte: Próprio autor (2017)

Em seguida, a mesma consulta é exibida em grafos, dessa vez modificando a cláusula *return*, como mostrado na Figura 42:

Figura 42: Consulta e resposta da relação mora e natural em grafo

```
1 MATCH (c) <- [m:mora] - (p:Pessoa{nome:"Pedro"}) - [n:natural] -> (a)
2 RETURN m,p,n
```



Fonte: Próprio autor (2017)

A Figura 43 expõe o nó “Pedro”, e sua conexão com a relação “formou-se”, no formato de linhas:

Figura 43: Consulta e resposta da relação formou-se em linhas

```
1 MATCH (p:Pessoa{nome:"Pedro"}) - [f:formou_se] -> (a)
2 RETURN f,a
```

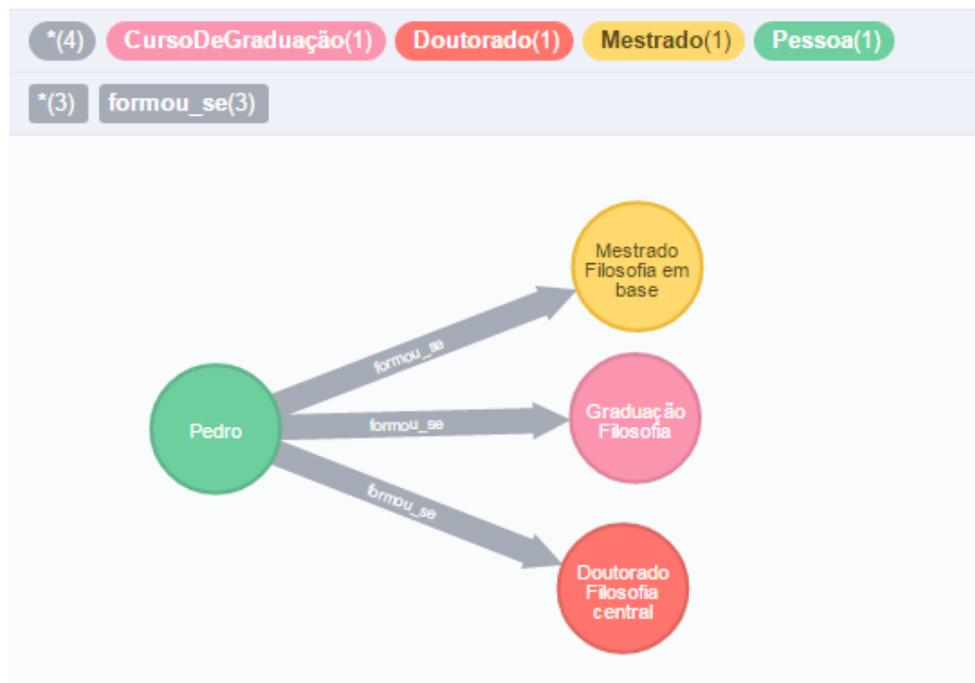
f	a														
<table border="1"> <tr><td>titulo_tcc</td><td>A descoberta do ser</td></tr> <tr><td>ano_ingresso</td><td>2007</td></tr> <tr><td>ano_conclusao</td><td>2012</td></tr> </table>	titulo_tcc	A descoberta do ser	ano_ingresso	2007	ano_conclusao	2012	<table border="1"> <tr><td>IES</td><td>Universidade 1</td></tr> <tr><td>cidade_IES</td><td>Cidade 1</td></tr> <tr><td>nome</td><td>Graduação Filosofia</td></tr> <tr><td>modalidade</td><td>Licenciatura</td></tr> </table>	IES	Universidade 1	cidade_IES	Cidade 1	nome	Graduação Filosofia	modalidade	Licenciatura
titulo_tcc	A descoberta do ser														
ano_ingresso	2007														
ano_conclusao	2012														
IES	Universidade 1														
cidade_IES	Cidade 1														
nome	Graduação Filosofia														
modalidade	Licenciatura														
<table border="1"> <tr><td>titulo_dissertação</td><td>A filosofia como resposta</td></tr> <tr><td>ano_ingresso</td><td>2012</td></tr> <tr><td>ano_conclusao</td><td>2014</td></tr> </table>	titulo_dissertação	A filosofia como resposta	ano_ingresso	2012	ano_conclusao	2014	<table border="1"> <tr><td>cidade_IES</td><td>Cidade 1</td></tr> <tr><td>IES</td><td>Universidade 1</td></tr> <tr><td>nome</td><td>Mestrado Filosofia em base</td></tr> </table>	cidade_IES	Cidade 1	IES	Universidade 1	nome	Mestrado Filosofia em base		
titulo_dissertação	A filosofia como resposta														
ano_ingresso	2012														
ano_conclusao	2014														
cidade_IES	Cidade 1														
IES	Universidade 1														
nome	Mestrado Filosofia em base														
<table border="1"> <tr><td>titulo_tese</td><td>Filosofia nas escolas</td></tr> <tr><td>ano_ingresso</td><td>2014</td></tr> <tr><td>ano_conclusao</td><td>2016</td></tr> </table>	titulo_tese	Filosofia nas escolas	ano_ingresso	2014	ano_conclusao	2016	<table border="1"> <tr><td>cidade_IES</td><td>Cidade 2</td></tr> <tr><td>IES</td><td>Universidade 2</td></tr> <tr><td>nome</td><td>Doutorado Filosofia central</td></tr> </table>	cidade_IES	Cidade 2	IES	Universidade 2	nome	Doutorado Filosofia central		
titulo_tese	Filosofia nas escolas														
ano_ingresso	2014														
ano_conclusao	2016														
cidade_IES	Cidade 2														
IES	Universidade 2														
nome	Doutorado Filosofia central														

Fonte: Próprio autor (2017)

A mesma consulta no formato de grafos mostrado na Figura 44, retornou todas as relações de “formou-se” do nó “Pedro”, no caso do exemplo “Pedro” formou-se três vezes, na qual cada uma dos nós correspondentes à sua formação são diferenciados pelo *labels* (CursoDeGraduação, Mestrado, Doutorado), apresentados por cores diferentes para facilitar a visualização. Pelos atributos é possível identificar as características de cada uma dessas formações.

Figura 44: Consulta e resposta da relação formou-se em linhas

```
1 MATCH (p:Pessoa{nome:"Pedro"})-[f:formou_se]->(a)
2 RETURN *
```



Fonte: Próprio autor (2017)

Para exibir a relação das pessoas que “orientaram” o nó “Pedro”, a Figura 45 mostra a consulta que foi realizada, onde pelo *return* “p.nome” retorna os nomes dos orientadores, e “o.tipo” em quais tipos de formações eles orientaram “Pedro”:

Figura 45: Consulta e resposta das pessoas que orientaram o nó Pedro

```
1 MATCH (p:Pessoa) - [o:orientou] -> (p1:Pessoa{nome:"Pedro"})
2 RETURN p.nome, o.tipo
```

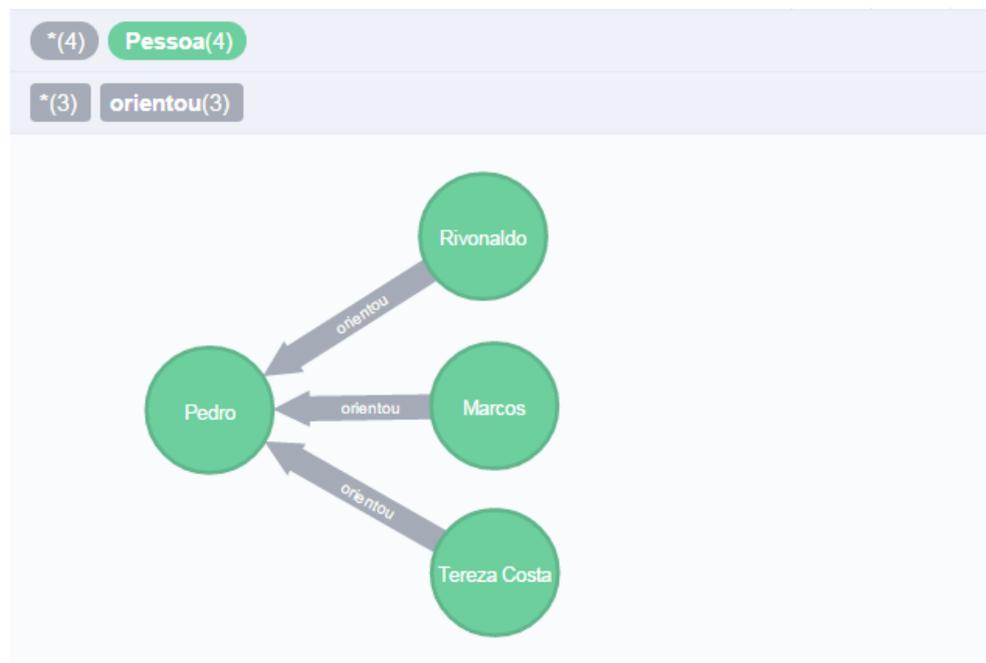
p.nome	o.tipo
Marcos	Doutorado
Tereza Costa	Mestrado
Rivonaldo	Curso de Graduação

Fonte: Próprio autor (2017)

Para visualizar o mesmo resultado em grafos, a consulta apresenta-se descrita na Figura 46:

Figura 46: Consulta e resposta das pessoas que orientaram o nó Pedro em grafos

```
1 MATCH (p:Pessoa) - [o:orientou] -> (p1:Pessoa{nome:"Pedro"})
2 RETURN o
```

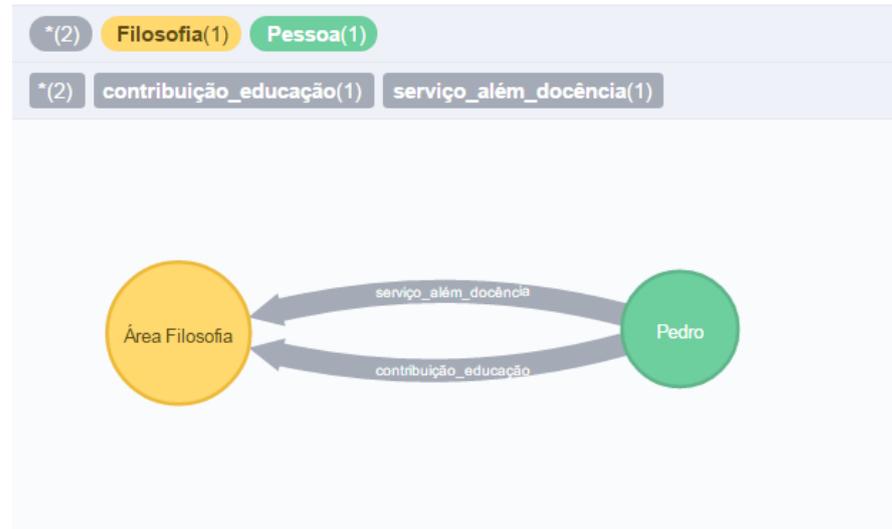


Fonte: Próprio autor (2017)

As relações entre o nó Pedro e a área Filosofia, são identificadas na Figura 47:

Figura 47: Consulta e resposta das relações entre o nó Pedro e o nó Filosofia

```
1 MATCH (p:Pessoa{nome:"Pedro"})-[r]->(f:Filosofia)
2 RETURN r
```

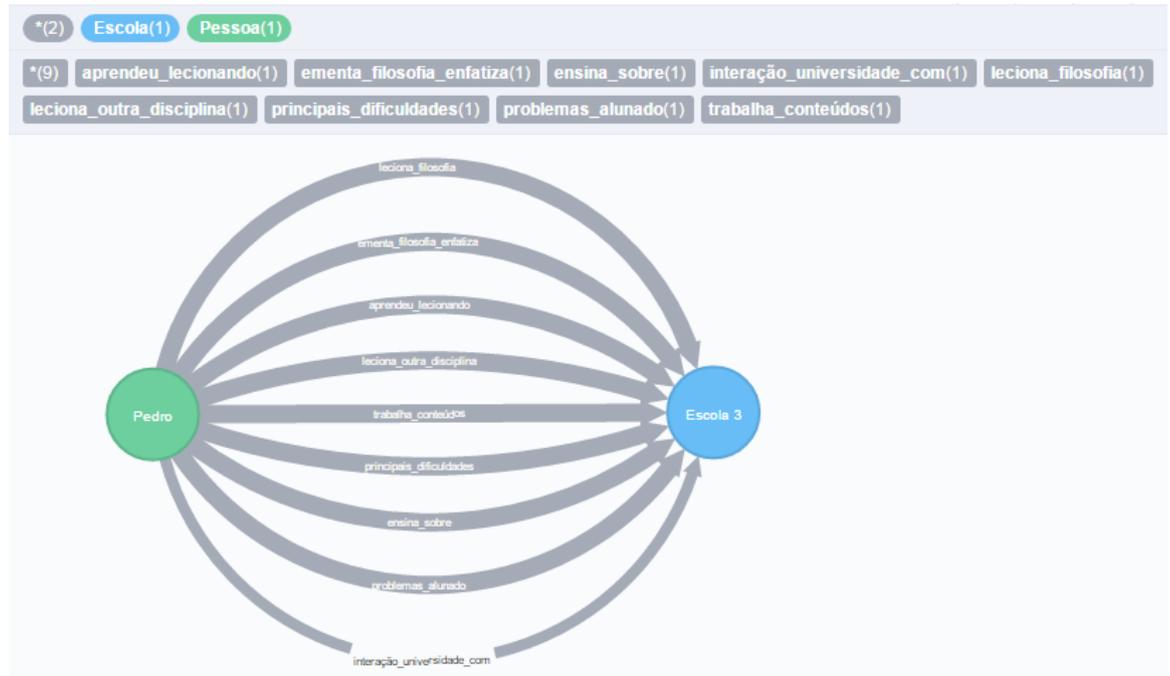


Fonte: Próprio autor (2017)

E por fim, para mostrar as relações modeladas a partir das perguntas da Figura 28, a consulta realizada exibe todas as relações existentes, entre o nó Pedro e a escola, como mostra a Figura 48:

Figura 48: Consulta e resposta das relações entre o nó Pedro e o nó Escola

```
1 MATCH (p:Pessoa{nome:"Pedro"})-[r]->(e:Escola)
2 RETURN r
```



Fonte: Próprio autor (2017)

CONSIDERAÇÕES FINAIS

Este trabalho apresentou os bancos de dados NoSQL e enfatizou os bancos de dados de grafos. A adoção destes bancos de dados estão passando por uma fase de crescimento e tem possibilitado o desenvolvimento de novas aplicações e uma maior exposição das relações existentes entre os dados.

A modelagem dos dados da Oficina Mímese com uma abordagem NoSQL orientada a grafos, permitiu concentrar o processo de análise de dados na figura do professor e sua relação com a escola. De maneira geral verificamos que o processo de modelagem de dados com grafos permite que os atributos das entidades, definidos como suas propriedades, tenham um papel secundário em relação à entidade em si e suas relações com outras entidades.

A escolha do banco de dados Neo4j possibilitou além de modelar os dados do estudo de caso, desenvolver um maior nível de compreensão da adoção do banco de dados em grafos, por ser um banco nativamente em grafo. Sua abordagem de definir apenas nós, relacionamentos e propriedades torna o modelo implementado mais próximo da definição matemática de grafo.

Com os dados modelados de forma a se adaptar em um contexto de grafos, onde cada pergunta do formulário transformou-se em um conjunto de nós, vértices e propriedades possibilitou enfrentar as dificuldades encontradas para modelar tais dados a partir de uma forma de pensar em dados tabulares, para pensar em dados estruturados em grafos.

Uma das vantagens notadas foi uma menor complexidade na estruturação dos dados, em já que há uma liberdade maior em adaptar o contexto já modelado, com adição ou remoção de determinados elementos. E os dados que conseguiram se manter interconectados desde à modelagem até o contexto final, quando os dados são inseridos no Neo4j.

Com o estudo de caso mostrou-se uma nova forma de modelar dados, que não fosse a forma tabular. Como validação da pesquisa, foi possível modelar o perfil dos professores do ensino básico que lecionam filosofia, descobrindo assim se de fato os alunos que se formam no curso de Filosofia em Mossoró estão no mercado de trabalho e como é essa atuação.

Dessa forma notou-se que à partir da inserção dos dados no Neo4j, as relações entre os dados foram priorizadas, e com isso permitiu que fosse possível extrair conhecimento à respeito das informações coletadas.

Como trabalhos futuros, pretende-se relacionar o KDD (Knowledge Discovery in Database) que é o processo de descoberta de conhecimento em bases de dados (FAYYAD et al., 1996 apud CARVALHO, 2016), com a modelagem em banco de dados de grafos e verificar como os bancos de dados de grafos podem contribuir na construção de sistemas de recomendação de conteúdo.

Identificamos que no âmbito do projeto Oficina Mímeses, temos outros formulários a serem modelados, pois só foi modelado os dados de um tipo de questionário, e desenvolver um sistema de interface com o banco de dados com uma aplicação para disponibilizar esses dados aos usuários.

Por fim, esperamos que este trabalho possa ter sua continuidade por meio de outras ações de pesquisa e aplicação na área de banco de dados de grafos.

REFERÊNCIAS

AEROSPIKE. Disponível em: <<http://www.aerospike.com/>>. Acesso em 16/03/17.

APACHE CASSANDRA. Disponível em: <<http://cassandra.apache.org/>>. Acesso em: 16/03/17.

APACHE COUCHDB. Disponível em: <<http://couchdb.apache.org/>>. Acesso em: 16/03/17.

APACHE HBASE. Disponível em: <<https://hbase.apache.org/>>. Acesso em 16/03/17

ARANGODB. Disponível em: <<https://www.arangodb.com/>>. Acesso em: 14/03/17.

BERNARDO D. S.; BORLONE D. R. Análise comportamental por intermédio das tecnologias de Big Data. **REFAS-Revista Fatec Zona Sul**, v. 2, n. 2, p. 20-42, 2016.

BOAVENTURA NETTO, P. O.; JURKIEWICZ, S. **Grafos: introdução e prática**. São Paulo: Blucher, 2009.

BOSCARIOLI C.; SOARES B. E. **Modelo de Banco de Dados Colunar: Características, Aplicações e Exemplos de Sistemas**, Universidade Estadual do Oeste do Paraná, 2013.

BREWER, E. A. Towards robust distributed systems. (Invited Talk). **Principles of Distributed Computing (PODC)**, Portland, Oregon, Julho 2000.

BRITO, Ricardo W. **Bancos de Dados NoSQL x SGBDs Relacionais: Análise Comparativa**. Faculdade Farias Brito e Universidade de Fortaleza, 2010.

CARDOSO R. M. F. **Bases de Dados NoSQL**. 2012. 77 p. Dissertação (Mestrado em Engenharia Informática) - Área de Especialização em Arquiteturas Sistemas e Redes, Instituto Superior de Engenharia do Porto, Porto, 2012.

CARVALHO D. C. **Obtenção de Padrões Sequenciais em Data Streams Atendendo Requisitos do Big Data**. 2016. 90 p. Dissertação (Mestrado em Ciência da Computação) – Área de concentração em Engenharia de Software e Banco de

Dados, Programa de PósGraduação em Ciência da Computação, Universidade Federal De São Carlos, São Carlos, 2016.

COSTA E. R. **Bancos de Dados Relacionais**. 2011. 63 p. Trabalho de Conclusão de Curso (Tecnólogo em Processamento de Dados) - Faculdade de Tecnologia de São Paulo, São Paulo.

COSTA, A. R. **Análise de redes sociais científicas para ampliação de redes de colaboração**. 2016. 77 p. Dissertação (Mestrado em Informática) - Universidade de Brasília, Brasília, 2016.

CUER E. S. **Comparação de Desempenho de Bancos de Dados Sql e Nosql**. 2014. 92 p. Monografia (Bacharel em Ciência da Computação) - Centro Universitário Eurípides de Marília, Marília; 2014.

EHCACHE. Disponível em: <<http://www.ehcache.org/>>. Acesso em 15/03/17.

ERVEN G. C. V. **MDG-NoSQL: Modelo de Dados para Bancos NoSQL Baseados em Grafos**. 2015. 91 p. Dissertação (Mestrado Profissional em Computação Aplicada), Universidade de Brasília, Brasília, 2015.

FREITAS M. C. **Mapeamentos Conceituais entre Modelo Relacional e Estruturas Nosql: um Estudo de Caso com Documentos**. 2015. 114 p. Dissertação (Mestrado em Ciência da Computação) – Área de Concentração em Bancos de Dados, Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, Recife, 2015.

FRIESS I. I. **Análise de Bancos de Dados Nosql e Desenvolvimento de Uma Aplicação**. 2013. 74 p. Trabalho de Graduação (Bacharel em Ciência da Computação) - Universidade Federal de Santa Maria, Santa Maria; 2013.

HENRIQUES M. S. B.; FURTADO M. R. S.; SILVA P. E. S.; MORAVIA R. V. Big Data. **Revista Pensar**, 2013.

LOPES J. M. **Um Estudo Comparativo entre Bancos de Dados Considerando as Abordagens Relacional e Orientada a Grafo**. 2014. 85 p. Trabalho de Conclusão de Curso (Bacharel em Tecnologias da Informação e Comunicação) - Universidade Federal de Santa Catarina, Araranguá, 2014.

LÓSCIO, Bernadette F.; OLIVEIRA, Hélio R.; PONTES, Jonas C. S. NoSQL no desenvolvimento de aplicações web colaborativas. **VIII Simpósio Brasileiro de Sistemas Colaborativos**, v. 10, p. 11, 2011.

MONETDB. Disponível em: <<https://www.monetdb.org/>>. Acesso em: 16/03/17.

MONGODB. Disponível em: <<https://www.mongodb.com/>>. Acesso em: 15/03/17.

MONIRUZZAMAN A. B. D.; HOSSAIN S. A. NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison, **International Journal of Database Theory and Application**, Vol. 6, No. 4. 2013.

MOURA, Aristoteles L. T.; AMORIM, Dinani G. Big Data: O Impacto e Sua Funcionalidade na Sociedade Tecnológica. **Revista Opara**, v. 4, n. 1, p. 71-83, 2014.

Neo4ja. **Walmart Optimizes Customer Experience with Real-time Recommendations**. Disponível em: <<https://neo4j.com/case-studies/walmart/>>. Acesso em: 17/10/2016.

Neo4jb. Disponível em: <<https://neo4j.com/docs/developer-manual/current/cypher/>>. Acesso em: 28/11/2016.

Neo4jc, 2016 Disponível em: <<https://neo4j.com/developer/cypher-query-language/>>. Acesso em: 20/02/17.

NOSQL, 2016. Disponível em: <<http://nosql-database.org/>>. Acesso em: 31/09/16.

OLIVEIRA W C. **Projeto de Pesquisa: Oficina Mímeses de Pesquisa em Ensino de Filosofia**, Edital 004/2012-DP/PROPEG/UERN, Mossoró, 2013.

ORIENTDB. Disponível em: <<http://orientdb.com/>>. Acesso em: 14/03/17.

PANIZ D. **Nosql: como armazenar os dados de uma aplicação moderna**. São Paulo, Casa do Código, 2016.

PEREIRA F. S.; BORGES H. P.; RUBENS H.; SANTANA S. A. **Utilização de Banco de Dados NoSql em Ambientes Corporativos**. Uberlândia; 2013.

PENTEADO R. R. M.; SCHROEDER R.; HOSS D.; NANDE J.; MAEDA R. M.; COUTO W. O.; HARA C. S. **Um Estudo sobre Bancos de Dados em Grafos Nativos**. 2014.

QUEIROZ, Gilberto R. D.; MONTEIRO, Antônio M. V.; CÂMARA, Gilberto. Bancos de dados geográficos e sistemas NoSQL: onde estamos e para onde vamos. **Revista Brasileira de Cartografia**, v. 65, n. 3, 2013.

REDIS. Disponível em: <<https://redis.io/>>. Acesso em 15/03/17.

ROBINSON, I.; WEBBER, J.; EIFREM, E. **Graph databases: new opportunities for connected data**. " O'Reilly Media, Inc.", 2015.

RODRIGUEZ, M. A.; NEUBAUER, Peter. Constructions from dots and lines. **Bulletin of the American Society for Information Science and Technology**, v. 36, n. 6, p. 35-41, 2010.

SADALAGE, P. J.; FOWLER, M. **NoSQL distilled: a brief guide to the emerging world of polyglot persistence**. Pearson Education, 2012.

SANTOS, E. U.; SILVA, M. A. L. **Abordagem ao Banco de Dados Orientado a Grafos Neo4j em um Nível Empresarial**, 2013.

SATONE K. N. Modern Graph Databases Models. **International Journal of Engineering Research and Applications**, 2014.

SILVA V. H. S. F. **BigTable: Um sistema de armazenamento distribuído para dados estruturados**. UFSCar – campus Sorocaba, 2011.

SOUZA A. M. ; PRADO E. P. V.; SUN V.; FANTINATO M. Critérios para Seleção de SGBD NoSQL: o Ponto de Vista de Especialistas com base na Literatura, **X Simpósio Brasileiro de Sistemas de Informação**, 2014, Londrina, PR.

STEPPAT N. **NoSQL - Do teorema CAP para P? (A|C):(C|L)**, 2011. Disponível em: <<http://blog.caelum.com.br/nosql-do-teorema-cap-para-paccl/>>. Acesso em 10/10/2016

TITAN. Disponível em: <<http://titan.thinkaurelius.com/>>. Acesso em: 14/03/17.

TOTH, R. M. **Abordagem NoSQL – uma real alternativa**. Sao Paulo; 2011.

VERTICA. Disponível em: <<https://www.vertica.com/>>. Acesso em: 15/03/17.

VIEIRA M. R.; FIGUEIREDO J. M.; LIBERATTI G.; VIEBRANTZ A. F. M. Bancos de Dados NoSQL: Conceitos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data, **Simpósio Brasileiro de Banco de Dados**, 2012, São Paulo, SP.